

Alibaba Cloud

Apsara Stack Enterprise

Cloud Defined Storage Developer Guide

Product Version: v3.16.2

Document Version: 20220916

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
<code>Courier font</code>	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.OSS	28
1.1. Usage notes	28
1.2. API Reference	28
1.2.1. Overview	28
1.2.2. Access control	28
1.2.2.1. Authenticate users by using signatures	28
1.2.2.2. Add signatures to the Authorization header	28
1.2.2.3. Generate a signed URL	32
1.2.2.4. Authorize temporary access	33
1.2.2.5. Bucket ACL	33
1.2.3. Common HTTP headers	34
1.2.4. Service-relevant operations	35
1.2.4.1. GetService (ListBuckets)	35
1.2.5. Bucket-relevant operations	37
1.2.5.1. Basic operations	37
1.2.5.1.1. PutBucket	37
1.2.5.1.2. GetBucket (ListObjects)	38
1.2.5.1.3. GetBucketInfo	44
1.2.5.1.4. GetBucketLocation	46
1.2.5.1.5. DeleteBucket	46
1.2.5.2. ACL	47
1.2.5.2.1. PutBucketACL	47
1.2.5.2.2. GetBucketAcl	48
1.2.5.3. Lifecycle	49
1.2.5.3.1. PutBucketLifecycle	49
1.2.5.3.2. GetBucketLifecycle	50

1.2.5.3.3. DeleteBucketLifecycle	52
1.2.5.4. Logging	52
1.2.5.4.1. PutBucketLogging	52
1.2.5.4.2. GetBucketLogging	54
1.2.5.4.3. DeleteBucketLogging	55
1.2.5.5. Static website hosting	56
1.2.5.5.1. PutBucketWebsite	56
1.2.5.5.2. GetBucketWebsite	62
1.2.5.5.3. DeleteBucketWebsite	63
1.2.5.6. Hotlink protection	64
1.2.5.6.1. PutBucketReferer	64
1.2.5.6.2. GetBucketReferer	65
1.2.5.7. Tags	66
1.2.5.7.1. PutBucketTags	66
1.2.5.7.2. GetBucketTags	67
1.2.5.7.3. DeleteBucketTags	68
1.2.5.8. CORS	69
1.2.5.8.1. Overview	69
1.2.5.8.2. PutBucketcors	69
1.2.5.8.3. GetBucketcors	71
1.2.5.8.4. DeleteBucketcors	72
1.2.5.8.5. OptionObject	72
1.2.5.9. Versioning	73
1.2.5.9.1. PutBucketVersioning	73
1.2.5.9.2. GetBucketVersioning	74
1.2.5.9.3. GetBucketVersions(ListObjectVersions)	75
1.2.6. Object-relevant operations	80
1.2.6.1. Basic operations	80

1.2.6.1.1. PutObject	80
1.2.6.1.2. GetObject	82
1.2.6.1.3. CopyObject	86
1.2.6.1.4. AppendObject	89
1.2.6.1.5. DeleteObject	91
1.2.6.1.6. DeleteMultipleObjects	93
1.2.6.1.7. HeadObject	97
1.2.6.1.8. GetObjectMeta	100
1.2.6.1.9. PostObject	101
1.2.6.2. ACL	106
1.2.6.2.1. PutObjectACL	106
1.2.6.2.2. GetObjectACL	107
1.2.6.3. Tagging	108
1.2.6.3.1. PutObjectTagging	108
1.2.6.3.2. GetObjectTagging	109
1.2.6.3.3. DeleteObjectTagging	110
1.2.6.4. Multipart upload	111
1.2.6.4.1. Overview	111
1.2.6.4.2. InitiateMultipartUpload	111
1.2.6.4.3. UploadPart	112
1.2.6.4.4. UploadPartCopy	113
1.2.6.4.5. CompleteMultipartUpload	115
1.2.6.4.6. AbortMultipartUpload	117
1.2.6.4.7. ListMultipartUploads	118
1.2.6.4.8. ListParts	120
1.2.7. Error responses	122
1.3. ASAPI Developer Guide	125
1.3.1. Preparations	125

1.3.1.1. Obtain API information	125
1.3.1.2. Obtain an AccessKey pair	125
1.3.1.3. Obtain the values of common header parameters	126
1.3.1.4. Obtain an STS AccessKey pair	127
1.3.2. Use ASAPI to make API calls	129
1.3.2.1. ASAPI overview	129
1.3.2.2. Obtain the endpoint of ASAPI	130
1.3.2.3. Obtain Apsara Stack SDKs	131
1.3.2.4. Common parameters	131
1.3.2.5. SDK description	132
1.3.2.5.1. ASAPI SDK for Java	132
1.3.2.5.2. ASAPI SDK for Python	135
1.3.2.5.3. ASAPI SDK for Node.js	137
1.3.3. HTTP requests	139
1.3.3.1. Overview	139
1.3.3.2. Request syntax	139
1.3.3.3. Request signatures	139
1.3.3.4. Responses	141
1.4. SDK Reference	142
1.4.1. Java SDK	142
1.4.1.1. Preface	142
1.4.1.2. Installation	143
1.4.1.3. Initialization	144
1.4.1.4. Quick start	146
1.4.1.5. Buckets	147
1.4.1.5.1. Create buckets	147
1.4.1.5.2. List buckets	147
1.4.1.5.3. Determine whether buckets exist	148

1.4.1.5.4. Manage bucket ACLs	148
1.4.1.5.5. Bucket tagging	149
1.4.1.5.6. Delete buckets	150
1.4.1.5.7. Manage lifecycle rules	150
1.4.1.5.8. Configure hotlink protection	151
1.4.1.5.9. Configure CORS	152
1.4.1.5.10. Configure logging	154
1.4.1.5.11. Configure static website hosting	154
1.4.1.6. Upload objects	155
1.4.1.6.1. Overview	155
1.4.1.6.2. Simple upload	155
1.4.1.6.3. Form upload	156
1.4.1.6.4. Append upload	158
1.4.1.6.5. Resumable upload	159
1.4.1.6.6. Multipart upload	160
1.4.1.6.7. Use a progress bar	165
1.4.1.7. Download objects	166
1.4.1.7.1. Overview	166
1.4.1.7.2. Streaming download	167
1.4.1.7.3. Download to local files	167
1.4.1.7.4. Range download	167
1.4.1.7.5. Resumable download	169
1.4.1.7.6. Conditional download	170
1.4.1.7.7. Use a progress bar	170
1.4.1.8. Manage objects	171
1.4.1.8.1. Overview	171
1.4.1.8.2. Determine whether an object exists	171
1.4.1.8.3. Manage object ACLs	172

1.4.1.8.4. Manage object metadata	172
1.4.1.8.5. List objects	174
1.4.1.8.6. Query objects	180
1.4.1.8.7. Delete objects	181
1.4.1.8.8. Copy objects	183
1.4.1.9. Versioning	185
1.4.1.9.1. Versioning	185
1.4.1.9.2. Upload objects	186
1.4.1.9.3. Download objects	188
1.4.1.9.4. Copy objects	189
1.4.1.9.5. List objects	190
1.4.1.9.6. Delete objects	195
1.4.1.9.7. Query object metadata	197
1.4.1.9.8. Manage the ACL of an object	198
1.4.1.10. Authorized access	198
1.4.1.11. IMG	201
1.4.1.12. Handle exceptions	204
1.4.1.13. FAQ	205
1.4.2. Python-SDK	209
1.4.2.1. Preface	209
1.4.2.2. Installation	210
1.4.2.3. Getting started	211
1.4.2.4. Initialization	212
1.4.2.5. Buckets	213
1.4.2.5.1. Create a bucket	213
1.4.2.5.2. List buckets	213
1.4.2.5.3. Determine whether a bucket exists	213
1.4.2.5.4. Manage bucket ACLs	214

1.4.2.5.5. Bucket tagging	214
1.4.2.5.6. Delete buckets	215
1.4.2.5.7. Manage lifecycle rules	216
1.4.2.5.8. Static website hosting	217
1.4.2.5.9. CORS	218
1.4.2.5.10. Hotlink protection	218
1.4.2.5.11. Configure logging	219
1.4.2.6. Upload objects	219
1.4.2.6.1. Overview	220
1.4.2.6.2. Simple upload	220
1.4.2.6.3. Append upload	221
1.4.2.6.4. Resumable upload	221
1.4.2.6.5. Multipart upload	222
1.4.2.6.6. Use progress bars	224
1.4.2.7. Download objects	224
1.4.2.7.1. Overview	224
1.4.2.7.2. Streaming download	224
1.4.2.7.3. Download to local files	225
1.4.2.7.4. Range download	225
1.4.2.7.5. Resumable download	226
1.4.2.7.6. Use progress bars	227
1.4.2.8. Manage objects	227
1.4.2.8.1. Overview	227
1.4.2.8.2. Determine whether an object exists	227
1.4.2.8.3. Manage object ACLs	227
1.4.2.8.4. Manage object metadata	228
1.4.2.8.5. List objects	229
1.4.2.8.6. Query objects	230

1.4.2.8.7. Delete objects	233
1.4.2.8.8. Copy objects	234
1.4.2.9. Versioning	234
1.4.2.9.1. Manage versioning	234
1.4.2.9.2. Upload objects	235
1.4.2.9.3. Download objects	236
1.4.2.9.4. Copy objects	237
1.4.2.9.5. List objects	238
1.4.2.9.6. Delete objects	242
1.4.2.9.7. Obtain object metadata	245
1.4.2.9.8. Manage the ACL of an object	245
1.4.2.10. Authorized access	246
1.4.2.11. Chinese characters and time	247
1.4.2.12. IMG	248
1.4.2.13. Handle exceptions	251
1.4.3. PHP-SDK	252
1.4.3.1. Preface	252
1.4.3.2. Installation	252
1.4.3.3. Initialization	253
1.4.3.4. Getting Started	255
1.4.3.5. Buckets	257
1.4.3.5.1. Create buckets	257
1.4.3.5.2. List buckets	258
1.4.3.5.3. Determine whether a bucket exists	258
1.4.3.5.4. Manage the ACL of a bucket	259
1.4.3.5.5. Manage bucket tagging	260
1.4.3.5.6. Delete buckets	261
1.4.3.5.7. Manage lifecycle rules	262

1.4.3.5.8. Hotlink protection	264
1.4.3.5.9. Configure logging	265
1.4.3.5.10. Configure static website hosting	267
1.4.3.5.11. Configure CORS	268
1.4.3.6. Upload objects	270
1.4.3.6.1. Overview	270
1.4.3.6.2. Simple upload	270
1.4.3.6.3. Append upload	271
1.4.3.6.4. Multipart upload	272
1.4.3.7. Download objects	276
1.4.3.7.1. Overview	276
1.4.3.7.2. Download to local files	276
1.4.3.7.3. Download objects to local memory	277
1.4.3.7.4. Range download	277
1.4.3.7.5. Conditional download	278
1.4.3.8. Manage objects	279
1.4.3.8.1. Overview	279
1.4.3.8.2. Determine whether an object exists	279
1.4.3.8.3. Manage the ACL of an object	279
1.4.3.8.4. Manage object metadata	280
1.4.3.8.5. List objects	282
1.4.3.8.6. Delete objects	284
1.4.3.8.7. Copy objects	285
1.4.3.9. Versioning	287
1.4.3.9.1. Manage versioning	287
1.4.3.9.2. Upload objects	289
1.4.3.9.3. Download objects	292
1.4.3.9.4. Copy objects	292

1.4.3.9.5. Delete objects	294
1.4.3.9.6. Query object metadata	297
1.4.3.9.7. Manage the ACL of an object	298
1.4.3.10. Authorized access	299
1.4.3.11. IMG	301
1.4.3.12. Error handling	305
1.4.4. C-SDK	306
1.4.4.1. Preface	306
1.4.4.2. Installation	307
1.4.4.3. Initialization	309
1.4.4.4. Getting Started	311
1.4.4.5. Buckets	316
1.4.4.5.1. Create buckets	316
1.4.4.5.2. List buckets	317
1.4.4.5.3. Manage bucket ACLs	318
1.4.4.5.4. Delete buckets	320
1.4.4.5.5. Manage lifecycle rules	321
1.4.4.5.6. CORS	324
1.4.4.5.7. Configure hotlink protection	327
1.4.4.5.8. Configure static website hosting	330
1.4.4.6. Upload objects	333
1.4.4.6.1. Overview	333
1.4.4.6.2. Simple upload	333
1.4.4.6.3. Append upload	336
1.4.4.6.4. Resumable upload	338
1.4.4.6.5. Multipart upload	339
1.4.4.6.6. Use progress bars	342
1.4.4.7. Download objects	343

1.4.4.7.1. Overview	343
1.4.4.7.2. Download to local files	344
1.4.4.7.3. Download to local memory	344
1.4.4.7.4. Range download	345
1.4.4.7.5. Resumable download	346
1.4.4.7.6. Use progress bars	348
1.4.4.8. Manage objects	349
1.4.4.8.1. Overview	349
1.4.4.8.2. Manage object ACLs	349
1.4.4.8.3. Manage object metadata	350
1.4.4.8.4. List objects	352
1.4.4.8.5. Delete objects	362
1.4.4.8.6. Copy objects	365
1.4.4.9. Authorized access	368
1.4.4.10. IMG	371
1.4.4.11. Error handling	377
1.4.5. Go-SDK	379
1.4.5.1. Preface	379
1.4.5.2. Installation	379
1.4.5.3. Initialization	380
1.4.5.4. Quick start	381
1.4.5.5. Buckets	383
1.4.5.5.1. Create buckets	384
1.4.5.5.2. List buckets	384
1.4.5.5.3. Manage bucket ACLs	385
1.4.5.5.4. Configure bucket tagging	386
1.4.5.5.5. Delete buckets	388
1.4.5.5.6. Manage lifecycle rules	389

1.4.5.5.7. Hotlink protection	390
1.4.5.5.8. Configure logging	392
1.4.5.5.9. Static website hosting	393
1.4.5.5.10. CORS	394
1.4.5.6. Upload objects	395
1.4.5.6.1. Overview	395
1.4.5.6.2. Simple upload	395
1.4.5.6.3. Append upload	397
1.4.5.6.4. Resumable upload	398
1.4.5.6.5. Multipart upload	399
1.4.5.6.6. Use a progress bar	404
1.4.5.7. Download objects	404
1.4.5.7.1. Overview	404
1.4.5.7.2. Streaming download	405
1.4.5.7.3. Range download	406
1.4.5.7.4. Download to local files	408
1.4.5.7.5. Resumable download	409
1.4.5.7.6. Conditional download	410
1.4.5.7.7. Use a progress bar	411
1.4.5.8. Manage objects	412
1.4.5.8.1. Overview	412
1.4.5.8.2. Determine whether objects exist	412
1.4.5.8.3. Manage object ACLs	413
1.4.5.8.4. Manage object metadata	414
1.4.5.8.5. List objects	416
1.4.5.8.6. Delete objects	421
1.4.5.8.7. Copy objects	423
1.4.5.9. Versioning	426

1.4.5.9.1. Manage versioning	426
1.4.5.9.2. Upload objects	427
1.4.5.9.3. Download objects	429
1.4.5.9.4. Copy objects	430
1.4.5.9.5. List objects	432
1.4.5.9.6. Delete objects	438
1.4.5.9.7. Query object metadata	442
1.4.5.9.8. Manage the ACL of an object	443
1.4.5.10. Authorized access	444
1.4.5.11. IMG	446
1.4.5.12. Error handling	450
1.5. Obtain an endpoint	453
1.6. Obtain an AccessKey pair	454
2.NAS	456
2.1. Preparations	456
2.1.1. Log on to the API Tool console	456
2.1.2. View the information about API operations	456
2.1.3. Obtain an AccessKey pair	457
2.1.4. Obtain an STS AccessKey 3-tuple	457
2.1.5. Obtain the values of common header parameters	459
2.1.6. Obtain the endpoint of NAS	460
2.1.7. Obtain NAS SDKs	460
2.2. Call examples for SDK for Java	461
2.2.1. Quick start	461
2.2.2. Use SDK	462
2.2.2.1. Install NAS SDK for Java	462
2.2.2.2. Configure authentication credentials	462
2.2.2.3. Configure connection requests	463

2.2.2.4. Initiate a request	463
2.2.2.5. Handle errors	464
2.2.2.6. Use CommonRequest to call API operations	464
2.3. Call examples for SDK for Python	466
2.3.1. Quick start	466
2.3.2. Use SDK	466
2.3.2.1. Install the SDK for Python	466
2.3.2.2. Configure authentication credentials	467
2.3.2.3. Set HTTPS requests	467
2.3.2.4. Initiate a request	467
2.3.2.5. Handle errors	467
2.3.2.6. Use CommonRequest to call API operations	468
2.4. API Reference	469
2.4.1. List of operations by function	469
2.4.2. Make API requests	470
2.4.2.1. Common parameters	470
2.4.2.2. Request structure	470
2.4.3. Region	471
2.4.3.1. DescribeRegions	471
2.4.4. NAS instance	472
2.4.4.1. CreateFileSystem	472
2.4.4.2. DeleteFileSystem	473
2.4.4.3. ModifyFileSystem	474
2.4.4.4. DescribeFileSystems	474
2.4.5. Mount point	479
2.4.5.1. CreateMountTarget	479
2.4.5.2. DeleteMountTarget	480
2.4.5.3. DescribeMountTargets	481

2.4.5.4. ModifyMountTarget	482
2.4.6. Permission group	483
2.4.6.1. CreateAccessGroup	483
2.4.6.2. DeleteAccessGroup	484
2.4.6.3. ModifyAccessGroup	485
2.4.6.4. DescribeAccessGroups	486
2.4.6.5. CreateAccessRule	487
2.4.6.6. DeleteAccessRule	488
2.4.6.7. ModifyAccessRule	489
2.4.6.8. DescribeAccessRules	490
2.4.7. Appendix: Error codes	492
3.EBS	495
3.1. Usage notes	495
3.2. Preparations	495
3.2.1. Log on to the API Tool console	495
3.2.2. Obtain API information	495
3.2.3. Obtain access credentials assigned by RAM	495
3.2.4. Obtain access credentials assigned by STS	496
3.2.5. Obtain the values of common request headers	498
3.2.6. Obtain an endpoint	499
3.2.7. Obtain an SDK	500
3.3. Java SDK call example	501
3.3.1. Quick start	501
3.3.2. User Manual	503
3.3.2.1. Install ECS or EBS SDK for Java	503
3.3.2.2. Configure authentication credentials	504
3.3.2.3. Configure connection requests	504
3.3.2.4. Initiate a request	505

3.3.2.5. Troubleshooting	506
3.3.2.6. Use CommonRequest to call API operations	506
3.4. Python SDK call example	509
3.4.1. Quick start	509
3.4.2. User Manual	511
3.4.2.1. Install ECS or EBS SDK for Python	511
3.4.2.2. Configure authentication credentials	512
3.4.2.3. Set HTTPS requests	512
3.4.2.4. Initiate a request	513
3.4.2.5. Handle errors	513
3.4.2.6. Use CommonRequest to call API operations	514
3.5. Go SDK call example	516
3.5.1. Quick start	516
3.5.2. User Manual	518
3.5.2.1. Install the core package of SDK for Go	518
3.5.2.2. Configure authentication credentials	518
3.5.2.3. Initiate a request	519
3.5.2.4. Handle errors	520
3.5.2.5. Use CommonRequest to call API operations	521
3.6. API references	523
3.6.1. Make API requests	523
3.6.1.1. Overview	523
3.6.1.2. Request syntax	523
3.6.1.3. API authorization	523
3.6.1.4. Request signatures	524
3.6.1.5. Responses	525
3.6.2. RAM authentication	526
3.6.3. Common parameters	527

3.6.4. Disk	528
3.6.4.1. CreateDisk	528
3.6.4.2. DescribeDisks	533
3.6.4.3. AttachDisk	541
3.6.4.4. DetachDisk	544
3.6.4.5. ModifyDiskAttribute	546
3.6.4.6. ReInitDisk	547
3.6.4.7. ResetDisk	549
3.6.4.8. ReplaceSystemDisk	551
3.6.4.9. ResizeDisk	555
3.6.4.10. DeleteDisk	558
3.6.5. Snapshots	559
3.6.5.1. CreateSnapshot	559
3.6.5.2. DeleteSnapshot	562
3.6.5.3. DescribeSnapshots	563
3.6.5.4. DescribeSnapshotLinks	568
3.6.5.5. DescribeSnapshotsUsage	570
3.6.5.6. CreateAutoSnapshotPolicy	571
3.6.5.7. DeleteAutoSnapshotPolicy	573
3.6.5.8. ModifyAutoSnapshotPolicyEx	574
3.6.5.9. ModifySnapshotAttribute	575
3.6.5.10. DescribeAutoSnapshotPolicyEX	576
3.6.5.11. ApplyAutoSnapshotPolicy	579
3.6.5.12. CancelAutoSnapshotPolicy	580
3.6.5.13. CreateSnapshotGroup	581
3.6.5.14. DescribeSnapshotGroups	584
3.6.5.15. DeleteSnapshotGroup	586
3.6.5.16. ModifySnapshotGroup	587

3.6.6. Storage sets	588
3.6.6.1. CreateStorageSet	588
3.6.6.2. DescribeStorageSets	590
3.6.6.3. DescribeStorageSetDetails	592
3.6.6.4. ModifyStorageSetAttribute	593
3.6.6.5. DeleteStorageSet	595
3.6.7. Asynchronous replication	595
3.6.7.1. CreateDiskReplicaPair	595
3.6.7.2. StartDiskReplicaPair	598
3.6.7.3. DescribeDiskReplicaPairs	600
3.6.7.4. DescribeDiskReplicaPairProgress	604
3.6.7.5. ModifyDiskReplicaPair	606
3.6.7.6. FailoverDiskReplicaPair	608
3.6.7.7. ReprotectDiskReplicaPair	609
3.6.7.8. StopDiskReplicaPair	611
3.6.7.9. DeleteDiskReplicaPair	613
3.6.8. Replication pair-consistent groups	615
3.6.8.1. CreateDiskReplicaGroup	615
3.6.8.2. AddDiskReplicaPair	617
3.6.8.3. StartDiskReplicaGroup	619
3.6.8.4. DescribeDiskReplicaGroups	621
3.6.8.5. ModifyDiskReplicaGroup	624
3.6.8.6. FailoverDiskReplicaGroup	626
3.6.8.7. ReprotectDiskReplicaGroup	628
3.6.8.8. RemoveDiskReplicaPair	629
3.6.8.9. StopDiskReplicaGroup	631
3.6.8.10. DeleteDiskReplicaGroup	633
3.6.9. Appendix	635

3.6.9.1. How to ensure idempotence	635
3.6.9.2. API operations and states related to a replication p... ..	636
3.6.9.3. API operations and states related to a replication p... ..	637
4. Log Service	639
4.1. Preparations	639
4.1.1. Log on to the API Tool console	639
4.1.2. Obtain API information	639
4.1.3. Obtain an AccessKey pair	639
4.1.4. Obtain the values of common headers	640
4.1.5. Obtain the endpoint of Log Service	641
4.1.6. Obtain a Log Service SDK	641
4.2. SDK Reference	642
4.2.1. Introduction	642
4.2.1.1. Overview	642
4.2.1.2. Basic configurations	642
4.2.1.3. Error handling	643
4.2.1.4. API specifications	646
4.2.2. Java SDK	647
4.2.3. .NET SDK	650
4.2.4. .NET Core SDK	651
4.2.5. PHP SDK	652
4.2.6. Python SDK	653
4.2.7. SDK for Android	656
4.2.8. SDK for C	656
4.2.9. SDK for Go	656
4.2.10. iOS SDK	656
4.2.11. SDK for C ++	657
4.2.12. Node.js SDK	657

4.3. API Reference	659
4.3.1. Overview	659
4.3.2. Common request headers	659
4.3.3. Common response headers	660
4.3.4. Request signatures	660
4.3.5. Common error codes	663
4.3.6. Project API	664
4.3.6.1. CreateProject	664
4.3.6.2. DeleteProject	665
4.3.6.3. UpdateProject	666
4.3.6.4. GetProject	667
4.3.6.5. ListProject	668
4.3.6.6. GetProjectLogs	670
4.3.7. Logstore operations	671
4.3.7.1. CreateLogstore	672
4.3.7.2. DeleteLogstore	673
4.3.7.3. UpdateLogstore	674
4.3.7.4. GetLogstore	675
4.3.7.5. ListLogstore	676
4.3.7.6. ListShards	677
4.3.7.7. SplitShard	679
4.3.7.8. MergeShards	681
4.3.7.9. GetCursor	683
4.3.7.10. GetCursorTime	684
4.3.7.11. PullLogs	685
4.3.7.12. PostLog	687
4.3.7.13. GetShipperStatus	689
4.3.7.14. RetryShipperTask	691

4.3.7.15. GetLogs	692
4.3.7.16. GetHistograms	695
4.3.7.17. UpdateIndex	697
4.3.7.18. CreateIndex	701
4.3.7.19. DeleteIndex	703
4.3.7.20. GetIndex	704
4.3.7.21. PutWebtracking	708
4.3.8. Logtail machine group operations	709
4.3.8.1. CreateMachineGroup	709
4.3.8.2. DeleteMachineGroup	711
4.3.8.3. UpdateMachineGroup	712
4.3.8.4. ListMachineGroup	713
4.3.8.5. GetMachineGroup	714
4.3.8.6. ApplyConfigToMachineGroup	716
4.3.8.7. RemoveConfigFromMachineGroup	717
4.3.8.8. ListMachines	718
4.3.8.9. GetAppliedConfigs	719
4.3.9. Logtail configuration operations	720
4.3.9.1. CreateConfig	720
4.3.9.2. ListConfig	722
4.3.9.3. GetAppliedMachineGroups	723
4.3.9.4. GetConfig	724
4.3.9.5. DeleteConfig	726
4.3.9.6. UpdateConfig	727
4.3.10. Consumer group operations	729
4.3.10.1. CreateConsumerGroup	729
4.3.10.2. DeleteConsumerGroup	730
4.3.10.3. GetCheckPoint	731

4.3.10.4. HeartBeat	732
4.3.10.5. ListConsumerGroup	733
4.3.10.6. UpdateCheckPoint	734
4.3.10.7. UpdateConsumerGroup	736
4.3.11. RAM or STS	737
4.3.11.1. Overview	737
4.3.11.2. Resource list	738
4.3.11.3. Action list	739
4.3.11.4. Authentication rules	740
4.3.12. Description of common resources	741
4.3.12.1. Encrypt data	741
4.3.12.2. Data model	742
4.3.12.3. Data encoding	744
4.3.12.4. Logstores	744
4.3.12.5. Shards	745
4.3.12.6. Logtail configurations	745
4.3.12.7. Logtail machine group	746
5.CSG	749
5.1. Preparations	749
5.1.1. Log on to the API Tool console	749
5.1.2. View the information about an API operation	749
5.1.3. Create an AccessKey pair	749
5.1.4. Obtain an STS AccessKey pair	750
5.1.5. Obtain the values of common header parameters	752
5.1.6. Obtain the endpoint of Cloud Storage Gateway	753
5.1.7. Obtain Cloud Storage Gateway SDKs	754
5.2. Call examples of SDK for Java	754
5.2.1. Quick start	754

5.2.2. Use CSG SDK for Java	755
5.2.2.1. Install SDK for Java	755
5.2.2.2. Configure authentication credentials	755
5.2.2.3. Configure HttpClientConfig	756
5.2.2.4. Initiate a request	756
5.2.2.5. Handle errors	757
5.3. Call examples of SDK for Python	757
5.3.1. Quick start	757
5.3.2. Use CSG SDK for Python	758
5.3.2.1. Install CSG SDK for Python	758
5.3.2.2. Configure authentication credentials	758
5.3.2.3. Initiate HTTPS requests	759
5.3.2.4. Initiate a request	759
5.3.2.5. Handle errors	759
5.4. API reference	760
5.4.1. List of operations by function	760
5.4.2. Common parameters	760
5.4.3. Gateway	761
5.4.3.1. CreateGateway	761
5.4.3.2. DescribeGateways	761
5.4.3.3. DescribeGateway	763
5.4.3.4. ModifyGateway	764
5.4.3.5. DeleteGateway	764
5.4.3.6. UpgradeGateway	765
5.4.3.7. RebootGateway	765
5.4.3.8. DescribeTasks	766
5.4.4. Shares	767
5.4.4.1. DeployCacheDisk	767

5.4.4.2. DescribeGatewayFileShares	767
5.4.4.3. ModifyGatewayFileShare	769
5.4.4.4. DeleteGatewayFileShares	770
5.4.4.5. CreateGatewaySMBUser	770
5.4.4.6. DescribeGatewaySMBUsers	771
5.4.4.7. DeleteGatewaySMBUser	772
5.4.4.8. CacheConfig parameter description	772
5.4.5. Cache disks	773
5.4.5.1. DescribeGatewayCaches	773
5.4.5.2. ExpandCacheDisk	774
5.4.5.3. DeleteGatewayCacheDisk	774

1.OSS

1.1. Usage notes

Before using an OSS API to implement a specific function, developers need to understand the two methods that are used to call OSS API operations and choose the appropriate method according to the actual situation.

 **Notice** If you encounter problems during use, contact Alibaba Cloud technical support.

OSS API operations can be called by using ASAPI or OSS SDKs. Select an appropriate calling method based on your actual scenarios.

- If you want to perform operations on OSS buckets in a resource set of a specified organization, such as creating buckets, use ASAPI SDKs to call the corresponding OSS API operation. For more information, see related topics in ASAPI Developer Guide of *OSS Developer Guide*.
- If you want to perform operations on objects (such as object upload) in a bucket without the resource set and organization specified, you can use an OSS SDK to call the specific operation. For more information, see related topics in SDK Reference of *OSS Developer Guide*.

 **Notice** To perform operations on objects in a bucket, you must have the required permissions.

1.2. API Reference

1.2.1. Overview

Apsara Stack Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service. It enables you to store a large amount of data in the cloud. You can upload data to and download data from OSS anytime, anywhere, and on any Internet device by calling the simple RESTful APIs. OSS enables you to develop a variety of services that are capable of processing large amounts of data. These services include multimedia sharing websites, online storage, and personal and corporate data backups.

 **Note** Before you call these operations, ensure that you understand the OSS service instructions and usage agreements.

Instructions

OSS API reference provides request syntax, parameters, sample requests, and sample responses for operations. To implement secondary development, use SDK packages. For more information about how to install and use SDK, see OSS SDK reference.

Terms

Term	Description
bucket	The container used to store objects in OSS. Every object is contained in a bucket.
object	Objects are the basic unit for data operations in OSS. Objects are also known as files. Each object has metadata, data, and a key. A key can uniquely identify an object in a bucket.
endpoint	The domain name used to access OSS. OSS uses HTTP RESTful APIs to provide services. Different regions are accessed by using different endpoints. A region has different endpoints for access over the internal network and for access over the Internet. For more information, see Obtain an endpoint .
AccessKey	The access credential that is used to identify the requester. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is used to verify the identity of the user, while the AccessKey secret is used to encrypt and verify the signature string. You must keep your AccessKey secret confidential. For more information, see Obtain an AccessKey pair .

1.2.2. Access control

1.2.2.1. Authenticate users by using signatures

OSS uses an AccessKey pair, which includes an AccessKey ID and an AccessKey secret to implement symmetric encryption and verify the identity of a requester. The AccessKey ID is used to verify the identity of the user, while the AccessKey secret is used to encrypt and verify the signature string. You must keep your AccessKey secret confidential.

 **Note** For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

AccessKey pairs can be categorized based on the account types.

- AccessKey pair of an Alibaba Cloud account: The AccessKey pair of an Alibaba Cloud account has full permissions on its resources.
- AccessKey pair of a RAM user: A RAM user is created and authorized by an Alibaba Cloud account. The AccessKey pair of a RAM user has limited permissions on specified resources.
- AccessKey pair of an STS temporary access credential: A temporary access credential is generated by an Alibaba Cloud account or a RAM user. The AccessKey pair of a temporary credential has limited permissions on specified resources within the validity period of the credential. An STS temporary access credential becomes invalid after its validity period is exceeded.

Before you send a request to OSS as an individual user, you must generate a signature string in the specified format for the request and use your AccessKey secret to encrypt the signature string and generate a verification code. After OSS receives the request, OSS finds the AccessKey secret based on your AccessKey ID, and uses the AccessKey secret to decrypt the signature string and verification code. Then, OSS calculates a verification code and compares it against the decrypted verification code. If the two verification codes are the same, OSS determines that the request is valid. Otherwise, OSS rejects the request and returns HTTP status code 403.

1.2.2.2. Add signatures to the Authorization header

You can include the Authorization header in the HTTP request to carry signature information and indicate that the requester has been authorized.

Calculation of the Authorization header

```
Authorization = "OSS " + AccessKeyId + ":" + Signature
Signature = base64(hmac-shal(AccessKeySecret,
    VERB + "\n"
    + Content-MD5 + "\n"
    + Content-Type + "\n"
    + Date + "\n"
    + CanonicalizedOSSHeaders
    + CanonicalizedResource))
```

- `AccessKeySecret` : the key required for a signature.
- `VERB` : the HTTP request method such as PUT, GET, POST, HEAD, or DELETE.
- `\n` : the line break.
- `Content-MD5` : optional. This parameter specifies the Content-MD5 value of the message content (excluding headers). The value is obtained by calculating the 128-bit hash value and then encoding the value in Base64. This request header can be used for the message validity check. The message content is valid if the received message content is the same as the content that is sent. Example: eB5eF1ptWaXm4bj5Pyxw=. For more information, visit [RFC 2616 Content-MD5](#).
- `Content-Type` : optional. This parameter specifies the type of the requested content. Example: application/octet-stream.
- `Date` : the time when the request is sent. The time must be in GMT. Example: Sun, 22 Nov 2015 08:16:38 GMT.
- `CanonicalizedOSSHeaders` : the HTTP headers in alphabetical order. The prefix of the HTTP headers is x-oss-.
- `CanonicalizedResource` : the OSS resource you want to access.

Specifically, the values of Date and CanonicalizedResource are required. If the difference between the time on the client when the request is sent and the time on the server when the request is received is greater than 15 minutes, the OSS server rejects the request and returns HTTP status code 403.

Create CanonicalizedOSSHeaders

All HTTP headers whose prefixes are x-oss- are called canonicalized OSS headers. Process of creating canonicalized OSS headers:

- The names of all HTTP request headers whose prefixes are x-oss- must be in lowercase letters. For example, `X-OSS-Meta-Name: TaoBao` is converted to `x-oss-meta-name: TaoBao`.
- If the request is sent by using the AccessKey ID and AccessKey secret obtained by using STS, you must add the obtained security-token value to the signature string in the `x-oss-security-token:security-token` format.
- Sort all obtained HTTP request headers in alphabetical order.
- Delete any space on either side of a delimiter between the request header and content. For example, `x-oss-meta-name: TaoBao` is converted to `x-oss-meta-name:TaoBao`.
- Separate all content and headers with the (`\n`) delimiter to form the final canonicalized OSS headers.

Note

- CanonicalizedOSSHeaders is optional. The (`\n`) delimiter at the end can be removed.
- To create only one canonicalized OSS header, add the (`\n`) delimiter to the end of the header. Example: `x-oss-meta-a\n`.
- To create multiple canonicalized OSS headers, add the (`\n`) delimiter to the end of each header. Example: `x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n`.

Create CanonicalizedResource

Destination OSS resources specified in the request sent by the user is called canonicalized resources. Process of creating canonicalized resources:

- Set CanonicalizedResource to empty character string ("").
- Specify the OSS resource you want to access in the following format: `/BucketName/ObjectName`. Set CanonicalizedResource to `/BucketName/` to access a bucket or to the forward slash (`/`) delimiter to access OSS.
- If the requested resource contains subresources, sort all subresources in alphabetical order and separate them with the ampersand (`&`) delimiter. The question mark (`?`) and subresource string are added to the end of the CanonicalizedResource string. In this case, CanonicalizedResource is in the format of `/BucketName/ObjectName? acl&uploadId=UploadId`.
- If query strings (also called HTTP request parameters) are specified in the request, sort these query strings and request values in alphabetical order. Separate each pair of parameters and values with the ampersand (`&`) delimiter. Add the concatenated content to CanonicalizedResource as parameters. In this case, CanonicalizedResource is in the format of `/BucketName/ObjectName? acl&response-content-type=ContentType&uploadId=UploadId`.

Note

- Subresources supported by OSS include but are not limited to: acl, uploads, location, cors, logging, website, referer, lifecycle, delete, append, tagging, objectMeta, uploadId, partNumber, security-token, position, img, style, styleName, replication, replicationProgress, replicationLocation, cname, bucketInfo, comp, qos, live, status, vod, startTime, endTime, symlink, x-oss-process, response-content-type, response-content-language, response-expires, response-cache-control, response-content-disposition, and response-content-encoding.
- Three types of subresources are available:
 - Resource identifiers such as acl, append, uploadId, and symlink subresources
 - Subresources that specify the response header fields. Example: `response=***`.
 - Methods to process objects. Example: `x-oss-process`.

Rules to calculate signatures for the Authorization header

- A signature string must be in the UTF-8 format. A signature string that contains Chinese characters must be encoded in UTF-8. The encoded signature string is used together with AccessKeySecret to calculate the final signature.
- The HMAC-SHA1 method defined in [RFC 2104](#) is used to calculate the final signature. In this method, the Key is `AccessKeySecret`.

- Content-Type and Content-MD5 are optional in a request. If the request requires signature verification, the null value must be replaced with a line break (`\n`).
- Among all non-standard HTTP headers, only headers that start with `x-oss-` require signature strings. Other non-standard HTTP headers are ignored by OSS. For example, the `x-oss-magic` header in the preceding example requires a signature string.
- Headers that start with `x-oss-` must comply with the following conventions before signature verification:
 - The header names must be in lowercase.
 - The headers are sorted in alphabetical order.
 - No spaces exist on either side of a colon (:) between the header name and value.
 - Each header is followed by a line break (`\n`). If no headers are used, `CanonicalizedOSSHeaders` is set to null.

Examples

Assume that the AccessKey ID is `4***7`, and the AccessKey secret is `0***v`.

Signature examples

Request	Signature string calculation formula	Signature string
<pre>PUT /nelson HTTP/1.0 Content-MD5: eB5eJFlptWaXm4biJSPyxw== Content-Type: text/html Date: Thu, 17 Nov 2005 18:49:58 GMT Host: oss-example.regionid.example.com X-OSS- Meta-Author: ***@bar.com X-OSS-Magic: a***a</pre>	<pre>Signature = base64(hmac- sha1(AccessKeySecret, VERB + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedOSSHeaders + CanonicalizedResource))</pre>	<pre>"PUT\n ***w==\n text/html\n Thu, 17 Nov 2005 18:49:58 GMT\n x-oss-magic:a***a\n x-oss-meta- author:***@bar.com\n/oss-example/nelson"</pre>

An example of the signature calculation method:

```
import base64
import hmac
import sha
h = hmac.new("0***v",
  "PUT\n***M=\ntext/html\nThu, 17 Nov 2005 18:49:58 GMT\nx-oss-magic:abracadabra\nx-oss-meta-author:***@bar.com\n/oss-example/nelson", sha)
Signature = base64.b64encode(h.digest())
print("Signature: %s" % Signature)
```

The signature calculation result must be `2***A=` based on `Authorization = "OSS" + AccessKeyId + ":" + Signature`. Authorization that consists of `OSS 4***7:2***A=` and the `Authorization` header is sent:

```
PUT /nelson HTTP/1.0
Authorization:OSS 4***7:2***A=
Content-Md5: e***w==
Content-Type: text/html
Date: Thu, 17 Nov 2005 18:49:58 GMT
Host: oss-example.regionid.example.com
X-OSS-Meta-Author: ***@bar.com
X-OSS-Magic: a***a
```

Detail analysis

- If the imported AccessKey ID does not exist or is inactive, 403 Forbidden is returned. Error code: `InvalidAccessKeyld`.
- If the Authorization value format in the request header is incorrect, 400 Bad Request is returned. Error code: `InvalidArgument`.
- All OSS requests must use GMT stipulated in HTTP/1.1. The date format is `date1 = 2DIGIT SP month SP 4DIGIT; day month year`. Example: 02 Jun 1982. In this format, day uses two digits. Therefore, Jun 2, 2 Jun 1982, and 2-Jun-1982 are all invalid formats.
- If no date is entered or the date is in the invalid format during signature verification, 403 Forbidden is returned. Error code: `AccessDenied`.
- The difference between the time on the client when the request is sent and the time on the server when the request is received must be within 15 minutes. Otherwise, 403 Forbidden is returned. Error code: `RequestTimeTooSkewed`.
- If the AccessKey ID is active but OSS determines that the signature of the request is invalid, 403 Forbidden is returned. The correct signature string for verification and encryption is returned in the response. You can check whether the signature string is valid based on the response of OSS. Sample response:


```
<? xml version="1.0" ? >
<Error>
  <Code>
    SignatureDoesNotMatch
  </Code>
  <Message>
    The request signature we calculated does not match the signature you provided. Check your key and signing method.
  </Message>
  <StringToSignBytes>
    47 45 54 0a 0a 0a 57 65 64 2c 20 31 31 20 4d 61 79 20 32 30 31 31 20 30 37 3a 35 39 3a 32 35 20 47 4d 54 0a 2f 75 73 72 65 61 6c 74 65 73 74 3f 61 6
    3 6c
  </StringToSignBytes>
  <RequestId>
    1***2
  </RequestId>
  <HostId>
    regionid.example.com
  </HostId>
  <SignatureProvided>
    y***8=
  </SignatureProvided>
  <StringToSign>
    GET
    Wed, 11 May 2011 07:59:25 GMT
    /oss-example? acl
  </StringToSign>
  <OSSAccessKeyId>
    A***Q
  </OSSAccessKeyId>
</Error>
```

OSS SDK automatically implements signatures in your requests. You do not need to manually calculate your signature when you use OSS SDK. For more information about the signature implementation for specific programming languages, see the OSS SDK code. The following table describes the sample code used to implement signatures in various SDKs.

Sample code used to implement signatures in various SDKs

SDK	Signature implementation
Java SDK	OSSRequestSigner.java
Python SDK	auth.py
.Net SDK	OssRequestSigner.cs
PHP SDK	OssClient.php
C SDK	oss_auth.c
JavaScript SDK	client.js
Go SDK	auth.go
Ruby SDK	util.rb
iOS SDK	OSSModel.m
Android SDK	OSSUtils.java

FAQ

Example: Calculate the Content-MD5 value of message content "123456789". Correct method:

1. Calculate the 128-bit MD5 hash.
2. Encode the 128-bit MD5 hash (but not the 32-bit string) in Base64.

Python is used in this example. Correct calculation code:

```
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789")
>>> base64.b64encode(hash.digest())
'eB5eJF1ptWaXm4bijSPyxw=='
```

Note that you must use `hash.digest()` to calculate the 128-bit MD5 hash.

```
>>> hash.digest()
'\x1e~$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'
```

A common incorrect method is to directly encode the 32-bit string in Base64. Example: Call the `hash.hexdigest()` function to obtain a printable 32-bit string.

```
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
```

Result of encoding the incorrect MD5 value in Base64:

```
>>> base64.b64encode(hashlib.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

1.2.2.3. Generate a signed URL

In addition to using an Authorization field in the header, you can also add signature information to a URL so that you can forward the URL to a third party for authorized access.

Implementation

- The following code provides an example on how to generate a signed URL:

```
http://oss-example.regionid.example.com/oss-api.pdf?
AccessKeyId=AccessKeyId&Expires=1141889120&Signature=vjb***%3D
```

- A signed URL must include at least the following parameters: Signature, Expires, and AccessKeyId.
 - Expires** specifies the timeout time of the URL. The value of this parameter follows the UNIX time format and must be in UTC. The value is the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970. If the time OSS receives the URL request is later than the value of Expires that is included in the signature, a request timeout error code is returned. For example, the current time is 1141889060. To create a URL that is scheduled to expire in 60 seconds, you can set the value of Expires to 1141889120.
 - AccessKeyId** specifies the AccessKey ID in the AccessKey pair.
 - Signature** specifies the signature information. For all requests and header fields that OSS supports, the algorithm of signing the string for a URL is basically the same as that of signing the string for a header.

```
Signature = urlencode(base64(hmac-sha1(AccessKeySecret,
VERB + "\n"
+ CONTENT-MD5 + "\n"
+ CONTENT-TYPE + "\n"
+ EXPIRES + "\n"
+ CanonicalizedOSSHeaders
+ CanonicalizedResource)))
```

The following differences are listed:

- When a signed URL is generated, the Expires parameter replaces the Date parameter.
- Signatures cannot be included in a URL and a header at the same time.
- If more than one Signature, Expires, or AccessKeyId value is imported, the first imported value is used.
- Before OSS authenticates a request by using a signature, OSS checks the request time to determine whether it is later than the time specified in Expires.
- When you add a signature string to a URL, you must encode the URL.

- When you add a signature to a URL for temporary access, you must include `security-token`. Format:

```
http://oss-example.regionid.example.com/oss-api.pdf?
AccessKeyId=AccessKeyId&Expires=1141889120&Signature=vjb***%3D&security-token=SecurityToken
```

Sample code

The following code provides an example on how to add a signature to a URL in Python:

```
import base64
import hmac
import sha
import urllib
h = hmac.new("O***V",
"GET\n\n1141889120\n/oss-example/oss-api.pdf",
sha)
urllib.quote(base64.encodestring(h.digest()).strip())
```

OSS SDK provides methods for adding a signature to a URL. For more information, see the "Authorized access" topic in SDK documents. The following table describes the implementation of a signed URL for OSS SDK.

URL signature implementation files for OSS SDK

SDK	URL signature method	Implementation file
Java SDK	OSSClient.generatePresignedUrl	OSSClient.java
Python SDK	Bucket.sign_url	api.py
.Net SDK	OssClient.GeneratePresignedUri	OssClient.cs
PHP SDK	OssClient.signUrl	OssClient.php
JavaScript SDK	signatureUrl	object.js
C SDK	oss_gen_signed_url	oss_object.c

Detail analysis

- If you adopt the approach of generating a signed URL, the authorized data is exposed on the Internet before the authorization period is exceeded. We recommend that you assess the use risks in advance.
- You can add a signature to a URL in PUT and GET requests.
- When a signature is added to a URL, the sequence of the Signature, Expires, and AccessKeyId parameters can be swapped. If one or more of the Signature, Expires,

and AccessKeyId parameters are missing, 403 Forbidden is returned. Error code: AccessDenied.

- If the current access time is later than the Expires value that is set in the request or the time format is invalid, 403 Forbidden is returned. Error code: AccessDenied.
- If a URL includes one or more of the Signature, Expires, and AccessKeyId parameters and the header includes signature information, 400 Bad Request is returned. Error code: InvalidArgument.
- When the signature string is generated, Date is replaced by Expires, but the headers such as content-type and content-md5 defined in the preceding section are still included. (Although Date still exists in the request header, you do not need to add Date to the signature string.)

1.2.2.4. Authorize temporary access

OSS allows you to use STS to authorize temporary access. You can use STS to grant a third-party application or your RAM user an access credential that specifies the custom validity period and permissions.

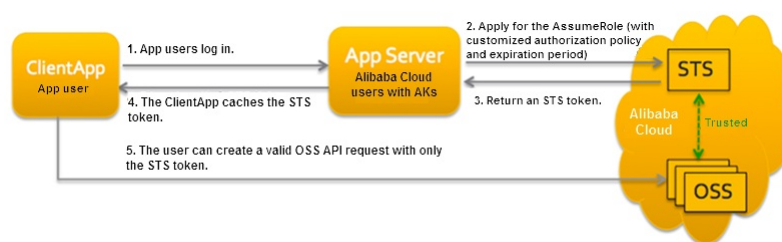
Introduction to STS

STS is a web service that provides temporary access tokens for cloud computing users. You can use STS to grant a third-party application or federated user (whose user ID is managed by yourself) an access credential with a custom validity period and permissions. The third-party application or federated user can use the access credential to directly call the operations provided by Alibaba Cloud services or use the Alibaba Cloud service SDKs to access the operations.

- You do not need to expose your long-term AccessKey pair to a third-party application. You need only to generate an access token and send it to the third-party application. You can customize the permissions and validity period of this token.
- The access token automatically becomes invalid when it expires.

The following figure uses an application as an example to show the authorization process.

Authorization process



Detailed solution description:

1. Log on as an application user. You can manage your application user ID. You can customize the ID management system or use an external web account or open ID. AppServer can define the principle of least privilege (POLP) of each valid application user.
2. AppServer requests a security token from STS. Before you call STS API operations, AppServer determines the POLP (described in policy syntax) of the app user and the validity period of the temporary security token. Then, AppServer calls the AssumeRole API operation of STS to obtain the security token. For more information about how to manage and use roles, see "Role management" in RAM User Guide.
3. STS returns a valid access credential to AppServer, including a security token, a temporary AccessKey pair (AccessKey ID and AccessKey secret), and the expiration time.
4. AppServer returns the access credential to ClientApp. ClientApp caches this credential. When the credential becomes invalid, ClientApp must request a new valid access credential from AppServer. For example, if the access credential is valid for one hour, ClientApp can request AppServer to update the access credential every 30 minutes.
5. ClientApp uses the cached access credential to call API operations provided by Alibaba Cloud services. Cloud services detect the STS access credential, verify the credential based on the STS service, and respond to the user request.

For more information about STS security tokens, see "Role management" in RAM User Guide. The key of the process is to call the AssumeRole operation of STS to obtain a valid access credential. You can also use STS SDK to call this API operation.

Use an STS credential to create a request that includes a signature

After the client of the user obtains an STS temporary credential, the client of the user uses the security token and temporary AccessKey pair (AccessKey ID and AccessKey secret) in the credential to create a signature. The method to create a signature for authorized access is the same as using the AccessKey pair of a root account to add a signature to the Authorization header. Take note of the following items:

- The signature key used by the user is the temporary AccessKey pair (AccessKey ID and AccessKey secret) that STS provides.
- The user needs to add the security token to the request header or in the URI as a request parameter. Only one of these two methods can be used. If both methods are selected, OSS returns InvalidArgument.
 - `x-oss-security-token:SecurityToken` is included in the request header. When CanonicalizedOSSHeaders of the signature is calculated, x-oss-security-token is taken into consideration.
 - `security-token=SecurityToken` is included in the URL. When CanonicalizedResource of the signature is calculated, security-token is taken into consideration as a subresource.

1.2.2.5. Bucket ACL

You can set the access control list (ACL) when you create a bucket, or modify the ACL for a created bucket. Only bucket owners can set or modify the ACL for buckets.

The following list describes the ACLs for buckets:

- Public read/write: Any users, including anonymous users can perform the PutObject, GetObject, and DeleteObject operations on the objects in the bucket. The bucket owner is charged for these operations. Exercise caution when you specify this ACL.
- Public read: Only the bucket owner can perform the PutObject and DeleteObject operations on the objects in the bucket. Other users, including anonymous users can perform only the GetObject operation on the objects in the bucket.
- Private: Only the bucket owner can perform the PutObject, GetObject, and DeleteObject operations on the objects in the bucket.

When you create a bucket, OSS sets the bucket ACL to private if you do not specify the bucket ACL. Only the bucket owner can call the PutBucketACL operation to modify the ACL of an existing bucket.

1.2.3. Common HTTP headers

This topic describes common request and response headers that are used for RESTful APIs.

Common request headers

In OSS, common HTTP request headers are used for RESTful APIs. These request headers can be used in all OSS requests. The following table lists the specific definitions of the request headers.

Common request headers

Header	Description
Authorization	The authentication information used to authenticate a request. Type: string Default value: null Scenario: non-anonymous requests
Content-Length	The content length of an HTTP request that is defined in RFC 2616. For more information, visit RFC 2616 . Type: string Default value: null Scenario: requests that need to submit data to OSS
Content-Type	The content type of an HTTP request that is defined in RFC 2616. For more information, visit RFC 2616 . Type: string Default value: null Scenario: requests that need to submit data to OSS
Date	The time in GMT stipulated in HTTP/ 1.1. Example: Wed, 05 Sep. 2012 23:00:00 GMT. Type: string Default value: null
Host	The value of the host to access. Format: <code><bucketname>.regionid.example.com</code> . Type: string Default value: null

Common response headers

In OSS, common HTTP response headers are used for RESTful APIs. These response headers can be used in all the OSS requests. The following table lists the specific definitions of the response headers.

Common response headers

Header	Description
Content-Length	The content length of an HTTP request that is defined in RFC 2616. For more information, visit RFC 2616 . Type: string Default value: null Scenario: requests that need to submit data to OSS
Connection	The connection status between the client and the OSS server. Type: enumeration Valid values: open and close Default value: null
Date	The time in GMT stipulated in HTTP/ 1.1. Example: Wed, 05 Sep. 2012 23:00:00 GMT. Type: string Default value: null
ETag	The entity tag (ETag) that is created to identify the content of the object when the object is created. If an object is created by using the PutObject request, the ETag value of the object is the MD5 hash of the object content. If an object is created by using another method, the ETag value of the content is the universally unique identifier (UUID) of the object content. The ETag value of the object can be used to check whether the object content is modified. Type: string Default value: null
Server	The server that generates a response. Type: string Default value: AliyunOSS

Header	Description
x-oss-request-id	The UUID that is created by the OSS server to identify the response. If you encounter any problems when you use OSS, you can contact OSS support personnel and provide this field to help them locate the problems. Type: string Default value: null

1.2.4. Service-relevant operations

1.2.4.1. GetService (ListBuckets)

You can call this operation to obtain all buckets that you own. The forward slash (/) in the request structure represents the root directory.

Usage notes

- The GetService operation is valid only for users who are authenticated.
- If no information for user authentication is provided in a request (anonymous access), OSS returns 403 Forbidden and error code AccessDenied.
- When all buckets are returned, the returned XML does not contain the Prefix, Marker, MaxKeys, IsTruncated, and NextMarker nodes. If not all results are returned, the preceding nodes are added, in which NextMarker is used to assign the marker for the subsequent query.

Request structure

```
GET / HTTP/1.1
Host: oss.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request parameters

When you call GetService (ListBucket), you can set the parameters described in the following table to limit the list of buckets returned so that only specified results are returned.

Parameter	Type	Required	Example	Description
prefix	String	No	example	The prefix that the names of returned buckets must contain. If this parameter is not specified, prefix information is not used to filter the returned buckets.
marker	String	No	examplebucket	The name of the object after which the list begins. If this parameter is not specified, all results are returned.
max-keys	String	No	100	The maximum number of buckets that can be returned this time. Valid values: 1 to 1000 Default value: 100

Response parameters

Parameter	Type	Example	Description
ListAllMyBucketsResult	Container	N/A	The container that stores the result of the GetService (ListBucket) request.
Prefix	String	example	The prefix that the names of returned buckets contain. This node is available only when not all buckets are returned.
Marker	String	examplebucket	The name of the bucket after which the list begins. This node is available only when not all buckets are returned.
MaxKeys	String	1	The maximum number of returned buckets in the response. This node is available only when not all buckets are returned.
IsTruncated	Enumerated string	true	Indicates whether all results are returned. Valid values: <code>true</code>: indicates that not all of the results are returned this time. <code>false</code>: indicates that all of the results are returned this time. This node is available only when not all buckets are returned.
NextMarker	String	xz02tphky6fjfiuc0	The marker for the next GetService (ListBucket) request to return the remaining results. This node is available only when not all buckets are returned.
Owner	Container	N/A	The container that stores the information about the bucket owner.

Parameter	Type	Example	Description
ID	String	17464*****02745	The user ID of the bucket owner.
DisplayName	String	17464*****02745	The name of the bucket owner, which is the same as the user ID.
Buckets	Container	N/A	The container that stores the information about multiple buckets.
Bucket	Container	N/A	The container that stores the bucket information.
Name	String	examplebucket	The name of the bucket.
CreateDate	Date	2011-12-01T12:27:13.000Z	The time when the bucket was created. Format: <code>yyyy-mm-ddThh:mm:ss.timezone</code> .
Location	String	oss-cn-shenzhen	The data center in which the bucket is located.
ExtranetEndpoint	String	oss-cn-shenzhen.aliyuncs.com	The public endpoint of the bucket.
IntranetEndpoint	String	oss-cn-shenzhen-internal.aliyuncs.com	The internal endpoint of the bucket.

Examples

• Sample Request 1

```
GET / HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: regionid.example.com
Authorization: OSS n***i:C***I=
```

• Sample Response 1

```
HTTP/1.1 200 OK
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 556
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5***4
<? xml version="1.0" encoding="UTF-8"? >
<ListAllMyBucketsResult>
  <Owner>
    <ID>17464*****02745</ID>
    <DisplayName>17464*****02745</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <CreateDate>2015-12-17T18:12:43.000Z</CreateDate>
      <ExtranetEndpoint>regionid.example.com</ExtranetEndpoint>
      <IntranetEndpoint>regionid-internal.example.com</IntranetEndpoint>
      <Location>oss-cn-shanghai</Location>
      <Name>app-base-oss</Name>
    </Bucket>
    <Bucket>
      <CreateDate>2014-12-25T11:21:04.000Z</CreateDate>
      <ExtranetEndpoint>regionid.example.com</ExtranetEndpoint>
      <IntranetEndpoint>regionid-internal.example.com</IntranetEndpoint>
      <Location>regionid</Location>
      <Name>examplebucket</Name>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

• Sample Request 2

```
GET /? prefix=xz02tphky6fjfiuc&max-keys=1 HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: regionid.example.com
Authorization: OSS n***i:C***I=
```

• Sample Response 2


```

HTTP/1.1 200 OK
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 545
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5***5
<? xml version="1.0" encoding="UTF-8"? >
<ListAllMyBucketsResult>
  <Prefix>xz02tphky6fjfiuc</Prefix>
  <Marker></Marker>
  <MaxKeys>1</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <NextMarker>xz02tphky6fjfiuc0</NextMarker>
  <Owner>
    <ID>17464*****02745</ID>
    <DisplayName>17464*****02745</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <CreationDate>2014-05-15T11:18:32.000Z</CreationDate>
      <ExtranetEndpoint>regionid.example.com</ExtranetEndpoint>
      <IntranetEndpoint>oss-cn-hangzhou-internal.example.com</IntranetEndpoint>
      <Location>oss-cn-hangzhou</Location>
      <Name>examplebucket</Name>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>

```

1.2.5. Bucket-relevant operations

1.2.5.1. Basic operations

1.2.5.1.1. PutBucket

You can call this operation to create a bucket. The region where the new bucket is located is consistent with the region in the endpoint that is contained in a request. After the data center of the bucket is determined, all objects in this bucket are stored in the corresponding region.

Request structure

```

PUT / HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
x-oss-acl: Permission
Authorization: SignatureValue

```

Usage notes

When you call the PutBucket operation, take note of the following items:

- Anonymous access is not supported.
- You can create up to 100 buckets in the same region by using an Alibaba Cloud account.

Request headers

Header	Type	Required	Example	Description
x-oss-acl	String	No	private	The access control list (ACL) for a bucket. Default value: private. Valid values: <i>public-read-write</i> <i>public-read</i> <i>private</i>

For more information about other common request headers included in PutBucket requests, see [Common request headers](#).

Request elements

Element	Type	Required	Example	Description
StorageClass	String	No	Standard	The storage class of the bucket. Default value: Standard. Valid storage: <i>Standard</i>

Response headers

The response headers involved in the PutBucket operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
PUT / HTTP/1.1
Host: BucketName.regionid.example.com
Date: Fri, 24 Feb 2012 03:15:40 GMT
x-oss-acl: private
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:77Dvh5wQgIjWjwO/KyRt8dOP****
<? xml version="1.0" encoding="UTF-8"? >
<CreateBucketConfiguration>
  <StorageClass>Standard</StorageClass>
</CreateBucketConfiguration>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Fri, 24 Feb 2012 03:15:40 GMT
Location: /**
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.5.1.2. GetBucket (ListObjects)

You can call this operation to list the information about all objects in a bucket.

Detail analysis

- User metadata of objects is not returned for the GetBucket request.
- If listing cannot be complete at a time because of the max-keys setting, `<NextMarker>` is added to the returned result, which indicates that the subsequent listing can start from this marker. The value of NextMarker is still in the list result.
- During a conditional query, even if the marker does not exist in the list, items after the marker are returned in alphabetical order. If the value of max-keys is smaller than 0 or greater than 1000, OSS returns 400 Bad Request and error code InvalidArgument.
- The prefix and marker parameters are used to list the returned objects by page. The parameter value must be smaller than 1,024 bytes in length.
- If prefix is set to a folder name in the request, the objects whose names contain this prefix are listed, including all objects and subfolders in the folder. If prefix is set to a folder name and delimiter is set to a forward slash (/) in the request, the names of the objects and subfolders in the folder are listed in the CommonPrefixes element. Objects and subfolders are not listed. For example, a bucket contains the following objects: fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. If prefix is set to fun/, the three objects are returned. If prefix is set to fun/ and delimiter is set to a forward slash (/), fun/test.jpg and fun/movie/ are returned.

Request structure

```
GET / HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request parameters

Parameter	Type	Required	Example	Description
delimiter	String	No	/	The character used to group objects by name. If you specify the Delimiter parameter in the request, the response contains the CommonPrefixes element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Default value: null
marker	String	No	test1.txt	The name of the object after which the list begins. If this parameter is specified, objects whose names are alphabetically greater than the marker parameter value are returned. The marker parameter is used to list the returned objects by page, and this parameter value must be smaller than 1,024 bytes in length. Even if the specified marker does not exist in the list during a conditional query, the list starts from the object whose name is alphabetically greater than the marker parameter value. Default value: null
max-keys	String	No	200	The maximum number of objects to return. If the listing cannot be complete at a time because max-keys is specified, the NextMarker element is included in the response as the marker for next listing. Valid values: 1 to 1000 Default value: 100

Parameter	Type	Required	Example	Description
prefix	String	No	fun	The prefix that the returned object names must contain. - The value of prefix must be smaller than 1,024 bytes in length. - If you specify a prefix to query objects, the names of the returned objects still contain the prefix. If prefix is set to a folder name in the request, the objects whose names contain this prefix are listed, including all objects and subfolders in the folder. If prefix is specified and delimiter is set to a forward slash (/), only the objects in the folder are listed. The subfolders are grouped together as a single result in CommonPrefixes. Objects and folders in the subfolders are not listed. For example, a bucket contains the following objects: fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. If prefix is set to fun/, the three objects are returned. If prefix is set to fun/ and delimiter is set to a forward slash (/), fun/test.jpg and fun/movie/ are returned. Default value: null
encoding-type	String	No	URL	The encoding type of the object name in the response. Default value: null Valid value: URL  Notice The delimiter, marker, prefix, NextMarker, and Key values are UTF-8 encoded. If delimiter, marker, prefix, NextMarker, and Key contain control characters that are not supported by the XML 1.0 standard, you can specify encoding-type to encode Delimiter, Marker, Prefix, NextMarker, and Key in the response.

Response elements

Element	Type	Example	Description
Contents	Container	N/A	The container that stores the returned object metadata. Parent nodes: ListBucketResult
CommonPrefixes	String	N/A	If the Delimiter parameter is specified in the request, the response contains the CommonPrefixes element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Parent nodes: ListBucketResult
Delimiter	String	/	The character used to group objects by name. If you specify the Delimiter parameter in the request, the response contains the CommonPrefixes element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Parent nodes: ListBucketResult
EncodingType	String	N/A	The encoding type of the object name in the response. In addition, you can use the encoding-type parameter to encode the following elements in the response: Delimiter, Marker, Prefix, NextMarker, and Key. Parent nodes: ListBucketResult
DisplayName	String	user_example	The name of the object owner. Parent nodes: ListBucketResult, Contents, and Owner
ETag	String	5B3C1A2E053D763E1B002CC607C5A0FE1****	The entity tag (ETag). An ETag is created when the object is created to identify the content of an object. - If an object is created using a PutObject request, the ETag value is the MD5 hash of the object content. - If an object is created using other methods, the ETag value is the UUID of the object content. - The ETag value of an object can be used to check whether the object content is changed. However, we recommend that you do not use the ETag of an object as the MD5 hash of the object to verify data integrity. Parent nodes: ListBucketResult and Contents
ID	String	0022012****	The user ID of the bucket owner. Parent nodes: ListBucketResult, Contents, and Owner
IsTruncated	Enumerated string	false	Indicates whether the returned results are truncated. Valid values: <i>true</i> and <i>false</i> - true indicates that not all of the results are returned this time. - false indicates that all of the results are returned this time. Parent nodes: ListBucketResult

Element	Type	Example	Description
Key	String	fun/test.jpg	The key of the object. Parent nodes: ListBucketResult and Contents
LastModified	Date	2012-02-24T08:42:32.000Z	The last modified time of the object. Parent nodes: ListBucketResult and Contents
ListBucketResult	Container	N/A	The container that stores the result of the GetBucket request. Child nodes: Name, Prefix, Marker, MaxKeys, Delimiter, IsTruncated, NextMarker, and Contents Parent node: none
Marker	String	test1.txt	The name of the object after which the listing begins. Parent nodes: ListBucketResult
MaxKeys	String	100	The maximum number of returned objects in the response. Parent nodes: ListBucketResult
Name	String	oss-example	The name of the bucket. Parent nodes: ListBucketResult
Owner	Container	N/A	The container that stores the information about the bucket owner. Child nodes: DisplayName and ID Parent nodes: ListBucketResult
Prefix	String	fun/	The prefix that the names of returned objects contain. Parent nodes: ListBucketResult
Size	String	344606	The size of the returned object. Unit: byte. Parent nodes: ListBucketResult and Contents
StorageClass	String	Standard	The storage class of the object. Parent nodes: ListBucketResult and Contents

For more information about other common response headers involved in the GetBucket operation, such as `x-oss-request-id` and `Content-Type`, see [Common response headers](#).

Examples

- Sample requests for the simple GetBucket operation

```
GET / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:BC+oQIXVR2/ZghT7cGa0ykbo****
```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1866
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Name>oss-example</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>100</MaxKeys>
  <Delimiter></Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>fun/movie/001.avi</Key>
    <LastModified>2012-02-24T08:43:07.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user-example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/movie/007.avi</Key>
    <LastModified>2012-02-24T08:43:27.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user-example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/test.jpg</Key>
    <LastModified>2012-02-24T08:42:32.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user-example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>oss.jpg</Key>
    <LastModified>2012-02-24T06:07:48.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user-example</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>

```

- Sample requests that have the prefix parameter specified

```

GET /? prefix=fun HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:BC+oQIXVR2/ZghT7cGa0ykbo****

```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1464
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Name>oss-example</Name>
  <Prefix>fun</Prefix>
  <Marker></Marker>
  <MaxKeys>100</MaxKeys>
  <Delimiter></Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>fun/movie/001.avi</Key>
    <LastModified>2012-02-24T08:43:07.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/movie/007.avi</Key>
    <LastModified>2012-02-24T08:43:27.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/test.jpg</Key>
    <LastModified>2012-02-24T08:42:32.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>

```

- Sample requests that have the prefix and delimiter parameters specified

```

GET /? prefix=fun/&delimiter=/ HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:DNrnX7xHk3sgysx7I8U9I9IY****

```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 712
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Name>oss-example</Name>
  <Prefix>fun/</Prefix>
  <Marker></Marker>
  <MaxKeys>100</MaxKeys>
  <Delimiter></Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>fun/test.jpg</Key>
    <LastModified>2012-02-24T08:42:32.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
  <CommonPrefixes>
    <Prefix>fun/movie/</Prefix>
  </CommonPrefixes>
</ListBucketResult>

```

- Sample requests that have the marker parameter specified

In this example, the value of max-keys is set to 2, which indicates that the maximum number of objects to return is two.

```

GET /? max-keys=2&marker=test1.txt HTTP/1.1
Host: test-bucket-xxx.oss-cn-shenzhen.aliyuncs.com
Accept-Encoding: identity
Accept: */*
Connection: keep-alive
User-Agent: aliyun-sdk-python/2.11.0 (Darwin/18.2.0/x86_64;3.4.1)
date: Tue, 26 May 2020 08:39:48 GMT
authorization: OSS LTAIBlVW9xxxxxxx:MmY11jLlO8U1AgjgHk3Ckp****

```

Sample responses

The value of NextMarker in the response is the marker for next listing.


```

HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 26 May 2020 08:39:48 GMT
Content-Type: application/xml
Content-Length: 1032
Connection: keep-alive
x-oss-request-id: 5ECCD5D4881816373582xxx
x-oss-server-time: 3
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult>
  <Name>test-bucket-xxx</Name>
  <Prefix></Prefix>
  <Marker>test1.txt</Marker>
  <MaxKeys>2</MaxKeys>
  <Delimiter></Delimiter>
  <EncodingType>url</EncodingType>
  <IsTruncated>true</IsTruncated>
  <NextMarker>test100.txt</NextMarker>
  <Contents>
    <Key>test10.txt</Key>
    <LastModified>2020-05-26T07:50:18.000Z</LastModified>
    <ETag>"C4CA4238A0B923820DCC509A6F75*****"</ETag>
    <Type>Normal</Type>
    <Size>1</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>1305433xxx</ID>
      <DisplayName>1305433xxx</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>test100.txt</Key>
    <LastModified>2020-05-26T07:50:20.000Z</LastModified>
    <ETag>"C4CA4238A0B923820DCC509A6F75*****"</ETag>
    <Type>Normal</Type>
    <Size>1</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>1305433xxx</ID>
      <DisplayName>1305433xxx</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>

```

1.2.5.1.3. GetBucketInfo

Queries information about a bucket.

You can view the following information about a bucket by calling GetBucketInfo:

- The time when the bucket was created
- The public endpoint of the bucket
- The internal endpoint of the bucket
- The information about the bucket owner
- The access control list (ACL) of the bucket

Request syntax

```

GET /?bucketInfo HTTP/1.1
Host: BucketName.oss.example.com
Date: GMT Date
Authorization: SignatureValue

```

Request headers

For more information about the common request headers involved in the GetBucketInfo operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
BucketInfo	Container	N/A	The container that stores the bucket information content.
Bucket	Container	N/A	The container that stores the specific information about the bucket.
CreationDate	Time	2013-07-31T10:56:21.000Z	The time when the bucket was created.
ExtranetEndpoint	String	oss-cn-hangzhou.aliyuncs.com	The public endpoint of the bucket.
IntranetEndpoint	String	oss-cn-hangzhou-internal.aliyuncs.com	The internal endpoint of the bucket.
Location	String	cn-hangzhou	The region of the data center to which the bucket belongs.

Element	Type	Example	Description
Name	String	oss-example	The name of the bucket.
Owner	Container	N/A	The container that stores the information about the bucket owner.
ID	String	2***3	The user ID of the bucket owner.
DisplayName	String	2***3	The name of the bucket owner, which is the same as the user ID.
AccessControlList	Container	N/A	The container that stores the ACL information.
Grant	Enumerated string	private	The ACL of the bucket. Valid values: private, public-read, and public-read-write
Versioning	String	Enabled	The versioning state of the bucket. Valid values: Enabled and Suspended

Examples

Sample requests

```
Get /?bucketInfo HTTP/1.1
Host: BucketName.oss.example.com
Date: Sat, 12 Sep 2015 07:51:28 GMT
Authorization: OSS q***c: B***=
```

Sample responses when the information about a bucket is returned

```
HTTP/1.1 200
x-oss-request-id: 5***B
Date: Sat, 12 Sep 2015 07:51:28 GMT
Connection: keep-alive
Content-Length: 531
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<BucketInfo>
  <Bucket>
    <CreationDate>2013-07-31T10:56:21.000Z</CreationDate>
    <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
    <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</IntranetEndpoint>
    <Location>cn-hangzhou</Location>
    <Name>oss-example</Name>
    <Owner>
      <DisplayName>2***3</DisplayName>
      <ID>2***3</ID>
    </Owner>
    <AccessControlList>
      <Grant>private</Grant>
    </AccessControlList>
  </Bucket>
</BucketInfo>
```

Sample responses when the specified information about a bucket is not found

```
HTTP/1.1 404
x-oss-request-id: 5***B
Date: Sat, 12 Sep 2015 07:51:28 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchBucket</Code>
  <Message>The specified bucket does not exist.</Message>
  <RequestId>5***4</RequestId>
  <HostId>nosuchbucket.oss.example.com</HostId>
  <BucketName>nosuchbucket</BucketName>
</Error>
```

Sample responses when you are not authorized to access information about a bucket

```

HTTP/1.1 403
x-oss-request-id: 5***C
Date: Sat, 12 Sep 2015 07:51:28 GMT
Connection: keep-alive
Content-Length: 209
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>AccessDenied</Code>
  <Message>AccessDenied</Message>
  <RequestId>5***E</RequestId>
  <HostId>BucketName.oss.example.com</HostId>
</Error>

```

1.2.5.1.4. GetBucketLocation

You can call this operation to view the location information about the data center to which a bucket belongs.

Request structure

```

GET /? location HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue

```

Request headers

For more information about the common request headers involved in the GetBucketLocation operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
LocationConstraint	String	oss-cn-hangzhou	The region to which the bucket belongs.

Examples

Sample requests

```

Get /? location HTTP/1.1
Host: *.regionid.example.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS q***c:c***=

```

Sample responses

```

HTTP/1.1 200
x-oss-request-id: 5***B
Date: Fri, 15 Mar 2013 05:31:04 GMT
Connection: keep-alive
Content-Length: 90
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<LocationConstraint xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">oss-cn-hangzhou</LocationConstraint >

```

1.2.5.1.5. DeleteBucket

You can call this operation to delete a bucket.

Request structure

```

DELETE / HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue

```

Usage notes

- To prevent unexpected data loss, OSS does not allow users to delete non-empty buckets.
- If you delete a non-empty bucket, OSS returns 409 Conflict and error code BucketNotEmpty.
- Only the bucket owner can delete the bucket. If you delete a bucket when you are not authorized, OSS returns 403 Forbidden and error code AccessDenied.

Request headers

The request headers involved in the DeleteBucket operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

The response headers involved in the DeleteBucket operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
DELETE / HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 05:31:04 GMT
Authorization: OSS g***:c***=
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

1.2.5.2. ACL

1.2.5.2.1. PutBucketACL

You can call this operation to modify the access control list (ACL) of a bucket.

Request structure

```
PUT /? acl HTTP/1.1
x-oss-acl: Permission
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Usage notes

When you call the PutBucketAcl operation, take note of the following items:

- To call this operation, you must have write permissions on the bucket.
- PutBucketAcl uses the overwriting semantics. A new ACL overwrites the existing one.
- If the specified bucket for which you want to set ACL does not exist when you call this operation, a new bucket is created.

Request headers

Header	Type	Required	Example	Description
x-oss-acl	String	Yes	private	The ACL that you want to set for the bucket. This header is included in PutBucketAcl requests to set the ACL of the bucket. If this header is not included, the ACL settings do not take effect. Valid values: public-read-write, public-read, and private • public-read-write: Any users, including anonymous users can read and write objects in the bucket. Exercise caution when you set the ACL of a bucket to public-read-write. • public-read: Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read objects in the bucket. Exercise caution when you set the ACL of a bucket to public-read. • private: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.

For more information about other common request headers involved in the PutBucketAcl operation, see [Common request headers](#).

Response headers

For more information about other common response headers involved in the PutBucketAcl operation, see [Common response headers](#).

Examples

Sample requests

```
PUT /? acl HTTP/1.1
x-oss-acl: public-read
Host: BucketName.regionid.example.com
Date: Fri, 24 Feb 2012 03:21:12 GMT
Authorization: OSS g***:K***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Fri, 24 Feb 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

If the specified ACL does not exist, OSS returns HTTP status code 400.

Sample error responses

```
HTTP/1.1 400 Bad Request
x-oss-request-id: 5***9
Date: Fri, 24 Feb 2012 03:55:00 GMT
Content-Length: 309
Content-Type: text/xml; charset=UTF-8
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <Code>InvalidArgument</Code>
  <Message>no such bucket access control exists</Message>
  <RequestId>5***9</RequestId>
  <HostId>***-test.example.com</HostId>
  <ArgumentName>x-oss-acl</ArgumentName>
  <ArgumentValue>error-acl</ArgumentValue>
</Error>
```

1.2.5.2.2. GetBucketAcl

You can call this operation to obtain the access control list (ACL) of a bucket.

Request structure

```
GET /? acl HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

For more information about the common request headers involved in the GetBucketACL operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
AccessControlPolicy	Container	N/A	The container that contains the result of the GetBucketACL request.
Owner	Container	N/A	The container that stores the information about the bucket owner.
ID	String	0022012****	The user ID of the bucket owner.
DisplayName	String	0022012****	The name of the bucket owner, which is the same as the user ID.
AccessControlList	Container	N/A	The container that contains the ACL information.
Grant	Enumerated string	public-read	The ACL for the bucket. Valid values: private, public-read, and public-read-write

Examples

Sample requests

```
GET /? acl HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 04:11:23 GMT
Authorization: OSS q***c:C***=
```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Fri, 24 Feb 2012 04:11:23 GMT
Content-Length: 253
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" ? >
<AccessControlPolicy>
  <Owner>
    <ID>0022012****</ID>
    <DisplayName>0022012****</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read</Grant>
  </AccessControlList>
</AccessControlPolicy>

```

1.2.5.3. Lifecycle

1.2.5.3.1. PutBucketLifecycle

A bucket owner has permissions to call PutBucketLifecycle to configure lifecycle for a bucket. After a lifecycle rule is enabled, OSS automatically deletes objects that match the rule or converts the storage class for the objects based on the rule on a regular basis.

Usage notes

When you call the PutBucketLifecycle operation, take note of the following items:

- Only bucket owners or RAM users who have the PutBucketLifecycle permission can initiate a PutBucketLifecycle request.
- If no lifecycle rules are configured for a bucket, a lifecycle rule is created after you call PutBucketLifecycle. If a lifecycle rule is configured for the bucket, a lifecycle rule is created and the existing lifecycle rule is overwritten.
- PutBucketLifecycle uses the overwriting semantics. New lifecycle rules overwrite the existing rules. To add lifecycle rules for a bucket, call GetBucketLifecycle to obtain the existing lifecycle configurations of the bucket, add new lifecycle configurations, and then call PutBucketLifecycle to update the lifecycle configurations for the bucket.
- When you call the PutBucketLifecycle operation, you can set a validity period or expiration date for objects or parts that are generated in incomplete multipart upload tasks.

Request structure

```

PUT /? lifecycle HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Authorization: SignatureValue
Host: BucketName.oss.example.com
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
  <Rule>
    <ID>RuleID</ID>
    <Prefix>Prefix</Prefix>
    <Status>Status</Status>
    <Expiration>
      <Days>Days</Days>
    </Expiration>
    <AbortMultipartUpload>
      <Days>Days</Days> </AbortMultipartUpload>
    </Rule>
</LifecycleConfiguration>

```

Request headers

Request headers involved in the PutBucketLifecycle operation contain only common request headers. For more information, see [Common request headers](#).

Request elements

Element	Type	Required	Example	Description
CreatedBeforeDate	String	Either Days or CreatedBeforeDate required	2002-10-11T00:00:00.000Z	The date based on which to run lifecycle rules. OSS performs the specified operation on data whose last modified date is before this date. Specify the time in the ISO 8601 standard. The time must be at UTC 00:00:00. For example, 2002-10-11T00:00:00.000Z indicates that objects last updated before 2002-10-11T00:00:00.000Z are deleted or the storage classes of these objects are converted to another storage class. Objects last updated at or after this time are not deleted or their storage classes are not converted to another storage class.
Days	Positive integer	Either Days or CreatedBeforeDate required	1	The number of days within which objects can be retained after they are last modified.

Element	Type	Required	Example	Description
Expiration	Container	No	N/A	The delete operation on the objects for the lifecycle rule.
AbortMultipartUpload	Container	No	N/A	The delete operation on the multipart upload tasks that are incomplete.
ID	String	No	rule1	The unique ID of a lifecycle rule. The ID can be a maximum of 255 bytes in length. If the value of this parameter is null or not specified, OSS automatically generates a unique ID for the lifecycle rule.
LifecycleConfiguration	Container	Yes	N/A	The container that stores lifecycle configurations. The container can contain up to 1,000 rules.
Prefix	String	Yes	data	The prefix of the names of objects to which the rule applies. The prefixes specified by different rules cannot overlap. - If Prefix is specified, this rule applies only to objects whose names contain the specified prefix in the bucket. - If Prefix is not specified, this rule applies to all objects in the bucket.
Rule	Container	Yes	N/A	Identifies the rule.
Status	String	Yes	Enabled	Specifies whether to run the rule. A value of Enabled indicates that OSS runs the rule. A value of Disabled indicates that OSS ignores the rule.

Response headers

The response headers involved in the PutBucketLifecycle operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
PUT /? lifecycle HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 443
Date: Mon, 14 Apr 2014 01:08:38 GMT
Authorization: OSS q**c:K***=
<? xml version="1.0" encoding="UTF-8"? >
</LifecycleConfiguration>
<Rule>
<ID>rule1</ID>
<Prefix>logs</Prefix>
<Status>Enabled</Status>
<Expiration>
<Days>1</Days>
</Expiration>
<AbortMultipartUpload>
<Days>1</Days>
</AbortMultipartUpload>
</Rule>
<Rule>
<ID>rule2</ID>
<Prefix>backup</Prefix>
<Status>Enabled</Status>
<Expiration>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</Expiration>
<AbortMultipartUpload>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</AbortMultipartUpload>
</Rule>
</LifecycleConfiguration>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Mon, 14 Apr 2014 01:17:10 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.5.3.2. GetBucketLifecycle

You can call this operation to view the lifecycle configurations of a bucket.

```
GET /? lifecycle HTTP/1.1
Host: BucketName.oss.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

For more information about the common request headers involved in the GetBucketLifecycle operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
CreatedBeforeDate	String	2002-10-11T00:00:00.000Z	The date based on which to run lifecycle rules. OSS performs the specified operation on data whose last modified date is before this date. Specify the time in the ISO 8601 standard. The time must be at UTC 00:00:00. For example, 2002-10-11T00:00:00.000Z indicates that objects last updated before 2002-10-11T00:00:00.000Z are deleted or the storage classes of these objects are converted to another storage class. Objects last updated at or after this time are not deleted or their storage classes are not converted to another storage class.
Days	Positive integer	1	The number of days within which objects can be retained after the objects are last modified.
Expiration	Container	N/A	The delete operation on the objects based on the lifecycle rule.
AbortMultipartUpload	Container	N/A	The delete operation on the multipart upload tasks that are incomplete.
ID	String	rule1	The unique ID of a lifecycle rule. The ID can be a maximum of 255 bytes in length. If the value of this parameter is null or not specified, OSS generates a unique ID for the lifecycle rule.
LifecycleConfiguration	Container	N/A	The container that stores lifecycle configurations. The container can contain up to 1,000 rules.
Prefix	String	logs/	The prefix of names of objects to which the rule applies. - If Prefix is specified, this rule applies only to objects whose names contain the specified prefix in the bucket. - If Prefix is not specified, this rule applies to all objects in the bucket.
Rule	Container	N/A	Identifies the rule.
Status	String	Enabled	Specifies whether to run the rule. A value of Enabled indicates that OSS runs the rule. A value of Disabled indicates that OSS ignores the rule.

Examples

Sample requests

```
Get /? lifecycle HTTP/1.1
Host: BucketName.oss.example.com
Date: Mon, 14 Apr 2014 01:17:29 GMT
Authorization: OSS q***:c***=
```

Sample responses for buckets that have lifecycle rules configured

```
HTTP/1.1 200
x-oss-request-id: 5***B
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: keep-alive
Content-Length: 255
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
<Rule>
<ID>rule1</ID>
<Prefix>logs</Prefix>
<Status>Enabled</Status>
<Expiration>
<Days>1</Days>
</Expiration>
<AbortMultipartUpload>
<Days>1</Days>
</AbortMultipartUpload>
</Rule>
<Rule>
<ID>rule2</ID>
<Prefix>backup</Prefix>
<Status>Enabled</Status>
<Expiration>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</Expiration>
<AbortMultipartUpload>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</AbortMultipartUpload>
</Rule>
</LifecycleConfiguration>
```

Sample responses for buckets that have no lifecycle rules configured


```

HTTP/1.1 404
x-oss-request-id: 5***B
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: keep-alive
Content-Length: 278
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <BucketName>oss-example</BucketName>
  <Code>NoSuchLifecycle</Code>
  <Message>No Row found in Lifecycle Table. </Message>
  <RequestId>5***9</RequestId>
  <HostId> BucketName.oss.example.com</HostId>
</Error>

```

1.2.5.3.3. DeleteBucketLifecycle

You can call this operation to delete the lifecycle configurations for a specific bucket.

Request structure

```

DELETE /? lifecycle HTTP/1.1
Host: BucketName.oss.example.com
Date: GMT Date
Authorization: SignatureValue

```

Request headers

Request headers involved in the DeleteBucketLifecycle operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

Response headers involved in the DeleteBucketLifecycle operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```

DELETE /? lifecycle HTTP/1.1
Host: BucketName.oss.example.com
Date: Mon, 14 Apr 2014 01:17:35 GMT
Authorization: OSS g***:6***=

```

Sample responses

```

HTTP/1.1 204 No Content
x-oss-request-id: 5***B
Date: Mon, 14 Apr 2014 01:17:35 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS

```

1.2.5.4. Logging

1.2.5.4.1. PutBucketLogging

Bucket owners can enable logging for their buckets. When this feature is enabled, OSS automatically records detailed information about the requests to this bucket, and follows the user-specified rules to write access logs as an object to the user-specified bucket on an hourly basis.

 **Notice** OSS provides logging for bucket owners to understand and analyze access to buckets. OSS does not guarantee that the bucket access logs contain all access records.

Usage notes

- The source bucket to which an object belongs and the destination bucket where the processed object is stored must belong to the same user.
- To disable logging for a bucket, you need only to send a request that contains an empty BucketLoggingStatus element. For more information about the specific method, see the following sample request.
- All PutBucketLogging requests must include signatures. Anonymous access is not supported.
- When the source bucket is deleted, the corresponding logging rules are also deleted.
- OSS generates bucket access log objects on an hourly basis. However, requests in the previous hour may be recorded in the log object generated for the subsequent hour.
- In the naming conventions for log objects generated by the OSS, UniqueString is just a UUID that OSS generates to uniquely identify the object.
- Each time OSS generates a bucket access log object, OSS performs a PUT operation and records the storage space that the object uses. However, OSS does not record the traffic generated by the PUT operation. After log objects are generated, you can perform operations on these objects as common objects.
- OSS ignores all query string parameters whose values contain the x- prefix. However, these parameters are recorded in access log objects. To identify a special request from a large number of access logs, you can add a query string parameter whose value contains the x- prefix to the URL of the request. Example

- `http://oss-example.regionid.example.com/aliyun-logo.png`
- `http://oss-example.regionid.example.com/aliyun-logo.png?x-user=admin`

When OSS processes the preceding two requests, the results are the same. However, you can search for access logs by `x-user=admin` to locate the marked request.

- A hyphen (-) may be contained in any field in OSS logs. The hyphen (-) indicates that data is unknown or the field is invalid for the current request.
- More fields will be added to OSS logs. We recommend that developers consider potential compatibility issues when you develop log processing tools.

Request structure

```
PUT /? logging HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Authorization: SignatureValue
Host: BucketName.regionid.example.com
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus>
  <LoggingEnabled>
    <TargetBucket>TargetBucket</TargetBucket>
    <TargetPrefix>TargetPrefix</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Request elements

Element	Type	Required	Example	Description
BucketLoggingStatus	Container	Yes	N/A	The container that stores the logging status information.
LoggingEnabled	Container	No	N/A	The container that stores the access log information. This element is required only when logging is enabled.
TargetBucket	String	This element is required when logging is enabled.	doc-log	The bucket that stores access logs.
TargetPrefix	String	No	MyLog-	The prefix of the access log object.

Response headers

Response headers involved in the PUTBucketLogging operation contain only common response headers. For more information, see [Common response headers](#).

The name format of the object that stores access logs

```
<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString
```

TargetPrefix is specified by the user. YYYY-mm-DD-HH-MM-SS indicates the time when an access log object is created. YYYY indicates the year (four-digit). mm indicates the month (two-digit). DD indicates the day (two-digit). HH indicates the hour (two-digit). MM indicates the minute (two-digit). SS indicates the second (two-digit). UniqueString indicates the string generated by OSS. An example for the name of an access log object:

```
MyLog-oss-example-2012-09-10-04-00-00-0000
```

In the preceding example, MyLog- is the object name prefix specified by the user. oss-example is the name of the source bucket. 2012-09-10-04-00-00 is the object creation time (UTC+8). 0000 is the string generated by OSS.

Log object format

Element	Example	Description
Remote IP	119.140.142.11	The IP address from which a request originates. The proxy or your firewall may block this field.
Reserved	N/A	A reserved field.
Reserved	N/A	A reserved field.
Time	[02/May/2012:00:00:04 +0800]	The time when OSS received a request.
Request-URI	"GET /aliyun-logo.png HTTP/1.1"	The requested URI, including query string parameters.
HTTP Status	200	The HTTP status code that OSS returned.
SentBytes	5576	The traffic consumed when a user downloads logs from OSS.
RequestTime (ms)	71	The length of time in milliseconds used to complete the request.
Referer	http://www.aliyun.com/product/oss	The HTTP header that identifies the address of the web page which is linked to the resource being requested.
User-Agent	curl/7.15.5	The user-agent header in an HTTP request.
HostName	oss-example.regionid.example.com	The domain name that is requested.

Element	Example	Description
Request ID	505B01695037C2AF032593A4	The UUID used to uniquely identify this request.
LoggingFlag	true	Indicates whether the access logging feature is enabled.
Requester Aliyun ID	1657136103983691	The Alibaba Cloud ID of the requester. The value of this field is a hyphen (-) for anonymous access.
Operation	GetObject	The type of a request.
Bucket	oss-example	The name of the bucket that you requested.
Key	/aliyun-logo.png	The key of the object that is requested.
ObjectSize	5576	The size of an object.
Server Cost Time (ms)	17	The length of time for the OSS server to process this request. Unit: milliseconds.
Error Code	NoSuchBucket	The error code that OSS returned.
Request Length	302	The length of the request. Unit: bytes.
UserID	1657136103983691	The ID of a bucket owner.
Delta DataSize	280	The variation of the size of a bucket. The value of this field is a hyphen (-) if the bucket size does not change.
Sync Request	-	Indicates whether the request is forwarded to the origin by using Content Delivery Network (CDN). The value of this field is a hyphen (-) if this is not a CDN back-to-origin request.
Reserved	-	A reserved field.

Examples

Sample requests where bucket logging is enabled:

```
PUT /? logging HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS q***c:K***=
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus>
<LoggingEnabled>
<TargetBucket>doc-log</TargetBucket>
<TargetPrefix>MyLog</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Sample requests where bucket logging is disabled:

```
PUT /? logging HTTP/1.1
Host: ***.regionid.example.com
Content-Type: application/xml
Content-Length: 86
Date: Fri, 04 May 2012 04:21:12 GMT
Authorization: OSS q***c:K***=
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus>
</BucketLoggingStatus>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Fri, 04 May 2012 04:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.5.4.2. GetBucketLogging

You can call this operation to view the logging configurations of a bucket.

Request structure

```
GET /? logging HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

For more information about the common request headers involved in the GetBucketLogging operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
BucketLoggingStatus	Container	N/A	The container that stores the logging status information.
LoggingEnabled	Container	N/A	The container that stores the access log information. This element is required only when logging is enabled.
TargetBucket	String	mybucketlogs	The bucket that stores access logs.
TargetPrefix	String	mybucket-access_log/	The prefix of the access log object.

Examples

Sample requests

```
Get /? logging HTTP/1.1
Host: *.regionid.example.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS g***c:c***=
```

Sample responses for buckets that have logging rules configured

```
HTTP/1.1 200
x-oss-request-id: 5***B
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 210
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access_log</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Sample responses for buckets that have no logging rules configured

```
HTTP/1.1 200
x-oss-request-id: 5***B
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 110
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
</BucketLoggingStatus>
```

1.2.5.4.3. DeleteBucketLogging

You can call this operation to disable logging for a bucket.

Request structure

```
DELETE /? logging HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

The request headers involved in the DeleteBucketLogging operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

The response headers involved in the DeleteBucketLogging operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
DELETE /? logging HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 05:35:24 GMT
Authorization: OSS q**c:6***=
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-request-id: 5***B
Date: Fri, 24 Feb 2012 05:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

1.2.5.5. Static website hosting

1.2.5.5.1. PutBucketWebsite

You can call this operation to set the static website hosting mode for a bucket.

Usage notes

- Static websites are the websites where all web pages consist of static content, such as scripts in JavaScript run on the client. OSS does not support content that needs to be processed by the server, such as PHP, JSP, and APS.NET content.
- After a bucket is set to the static website hosting mode, OSS returns the index page for anonymous access to the root domain name of the static website, and returns the result of GetBucket for signed access to the root domain name of the static website.

Request structure

```
PUT /? website HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Host: BucketName.regionid.example.com
Authorization: SignatureValue
<? xml version="1.0" encoding="UTF-8"? >
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>errorDocument.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

Request parameters

- The following table describes the parameters that you can configure in the WebsiteConfiguration field in a request.

Parameter	Type	Required	Description
WebsiteConfiguration	Container	Yes	The root node. Parent nodes: none

- The following table describes the parameters that you can configure in the IndexDocument field in a request.

Element	Type	Required	Example	Description
IndexDocument	Container	Conditional. You must specify at least one of the following containers: IndexDocument, ErrorDocument, and RoutingRules.	N/A	The container for the default homepage. Parent nodes: WebsiteConfiguration
Suffix	String	Conditional. This parameter must be specified if IndexDocument is specified.	index.html	The default homepage. If this parameter is set, the default homepage is returned for access to objects whose names end with a forward slash (/). Parent nodes: IndexDocument

Element	Type	Required	Example	Description
SupportSubDir	String	No	false	Specifies whether a request sent to access a subfolder is redirected to the default homepage of the subfolder. Valid values: - A value of true indicates that the request is redirected to the default homepage of the subfolder. - A value of false indicates that the request is redirected to the default homepage of the root folder instead of that of the subfolder. If the default homepage for access to bucket.oss-cn-hangzhou.aliyuncs.com/subdir/ is set to index.html. A value of false indicates that the request is redirected to bucket.oss-cn-hangzhou.aliyuncs.com/index.html, and a value of true indicates that the request is redirected to bucket.oss-cn-hangzhou.aliyuncs.com/subdir/index.html. Default value: false Parent nodes: IndexDocument
Type	Enumeration	No	0	Specifies the operation to perform when the default homepage is set, the name of the accessed object does not end with a forward slash (/), and the object does not exist.  Note This parameter takes effect only when SupportSubDir is set to true. It takes effect after RoutingRule and before ErrorFile. If the default homepage for access to bucket.oss-cn-hangzhou.aliyuncs.com/abc is set to index.html and the abc object does not exist. The following content describes the operations that correspond to the valid values of Type: 0: OSS checks whether the object named abc/index.html, which is in the <code>Object + Forward slash (/) + Homepage</code> format exists. If the object exists, OSS returns 302 and the Location header <code>/abc/</code>, which is in the <code>Forward slash (/) + Object + Forward slash (/)</code> format, in the response is URL-encoded. If the object does not exist, OSS returns 404 and continues to check ErrorFile. 1: OSS returns 404 and NoSuchKey and continues to check ErrorFile. 2: OSS checks whether abc/index.html exists. If abc/index.html exists, the content of the object is returned. If abc/index.html does not exist, OSS returns 404 and continues to check ErrorFile. Default value: 0 Parent nodes: IndexDocument

- The following table describes the parameters that you can configure in the ErrorDocument field in a request.

Parameter	Type	Required	Description
ErrorDocument	Container	Conditional. You must specify at least one of the following containers: IndexDocument, ErrorDocument, and RoutingRules.	The container used to store the default 404 page. Parent nodes: WebsiteConfiguration
Key	Container	Conditional. This parameter must be specified if ErrorDocument is specified.	The default 404 page. If this parameter is specified, the 404 page is returned when the target object does not exist. Parent nodes: ErrorDocument

- The following table describes RoutingRules, RoutingRule, and RuleNumber you can configure in a request.

Parameter	Type	Required	Example	Description
RoutingRules	Container	Conditional. You must specify at least one of the following containers: IndexDocument, ErrorDocument, and RoutingRules.	N/A	The container used to store RoutingRule. Parent nodes: WebsiteConfiguration

Parameter	Type	Required	Example	Description
RoutingRule	Container	No	N/A	Specifies redirection-based back-to-origin rules or mirroring-based back-to-origin rules. You can specify a maximum of 20 rules. Parent nodes: RoutingRules
RuleNumber	String	Conditional. This parameter must be specified if RoutingRule is specified.	1	The sequence number used to match and execute redirection rules. Redirection rules are matched based on this parameter. If a match succeeds, the rule is executed and the subsequent rules are not executed. Parent nodes: RoutingRule

- The following table describes RoutingRules, RoutingRule, and Condition in a request.

Parameter	Type	Required	Example	Description
Condition	Container	Conditional. This parameter must be specified if RoutingRule is specified.	N/A	The matching conditions. If all of the specified conditions are met, the rule is executed. A rule is determined as matched only when the rule meets the conditions specified by all nodes in Condition. Parent nodes: RoutingRule
KeyPrefixEquals	String	No	abc/	Specifies that only objects whose names contain the specified prefix match the rule. Parent nodes: Condition
HttpErrorCodeReturnedEquals	HTTP status code	No	404	Specifies that the rule is matched only when the specified object is accessed and the specified status code is returned. If the redirection rule is a mirroring-based back-to-origin rule, the parameter value must be 404. Parent nodes: Condition
IncludeHeader	Container	No	N/A	Specifies that the rule is matched only when the specified header is included in the request and the header value is equal to the specified value. You can specify up to five headers. Parent nodes: Condition
Key	String	Conditional. This parameter must be specified if IncludeHeader is specified.	host	Specifies that the rule is matched only when the specified header is included in the request and the header value is equal to the value specified by Equals. Parent nodes: IncludeHeader
Equals	String	Conditional. This parameter must be specified if IncludeHeader is specified.	test.oss-cn-beijing-internal.aliyuncs.com	Specifies that the rule is matched only when the header specified by Key is included in the request and the header value is equal to the specified value. Parent nodes: IncludeHeader
KeySuffixEquals	String	No	zip	Specifies that only objects whose names contain the specified suffix match the rule. The default value is null, which indicates that no suffix is specified. Parent nodes: Condition

- The following table describes RoutingRules, RoutingRule, and Redirect you can configure in a request.

Parameter	Type	Required	Example	Description
Redirect	Container	Conditional. This parameter must be specified if RoutingRule is specified.	N/A	The operation to perform after the rule is matched. Parent nodes: RoutingRule

Parameter	Type	Required	Example	Description
RedirectType	String	Conditional. This parameter must be specified if Redirect is specified.	Mirror	The redirection type. Valid values: - Mirror: mirroring-based back-to-origin - External: external redirection. OSS returns the 3xx HTTP redirect code and the Location header for you to redirect the request to another IP address. - Internal: internal redirection. OSS redirects the access from Object1 to Object2 based on the rule. In this case, the user accesses Object2 instead of Object1. - AliCDN: redirection based on Alibaba Cloud CDN. OSS adds an additional header to the request, which is different from the External type. After CDN identifies the header, CDN redirects the access to the specified IP address and returns the obtained data instead of the redirect request to the user. Parent nodes: Redirect
PassQueryString	Boolean	No	true	Specifies whether parameters in the original request are contained in the redirect request when redirection-based or mirroring-based back-to-origin is performed. For example, if the PassQueryString parameter is set to true and the <code>?a=b&c=d</code> parameter is contained in the request sent to OSS, this parameter is added to the Location header in the redirect request when the HTTP redirect code is 302. For example, if the request is <code>Location:www.test.com? a=b&c=d</code> and the redirecting type is mirroring-based back-to-origin, the <code>?a=b&c=d</code> parameter is also contained in the back-to-origin request. Valid values: true and false Default value: false Parent nodes: Redirect
MirrorURL	String	Conditional. This parameter must be specified if RedirectType is set to Mirror.	<code>http://www.example.com/</code>	Specifies the IP address of the origin in mirroring-based back-to-origin. This parameter takes effect only when the value of RedirectType is Mirror. URLs that start with <code>http://</code> or <code>https://</code> must end with a forward slash (/). OSS adds the object name to the string to form the back-to-origin URL. For example, the object to access is myobject. If the specified URL is <code>http://www.example.com/</code>, the back-to-origin URL is <code>http://www.test.com/myobject</code>. If the specified URL is <code>http://www.test.com/dir1/</code>, the back-to-origin URL is <code>http://www.test.com/dir1/myobject</code>. Parent nodes: Redirect
MirrorPassQueryString	Boolean	No	true	This parameter plays the same role as PassQueryString and has a higher priority than PassQueryString. This parameter takes effect only when the value of RedirectType is Mirror. Default value: false Parent nodes: Redirect
MirrorFollowRedirect	Boolean	No	true	Specifies whether the access is redirected to the specified Location if the origin returns a 3XX HTTP status code and when the origin receives a mirroring-based back-to-origin request. For example, when a mirroring-based back-to-origin request is initiated, the origin returns 302, and Location is specified. - In this case, if the value of MirrorFollowRedirect is true, OSS continues to send requests to the IP address specified by Location. A request can be redirected for a maximum of 10 times. If a request is redirected for more than 10 times, a mirroring-based back-to-origin failure is returned. - If the value of MirrorFollowRedirect is false, OSS returns 302 and passes through the Location. This parameter takes effect only when the value of RedirectType is Mirror. Default value: true Parent node: Redirect

Parameter	Type	Required	Example	Description
MirrorCheckMd5	Boolean	No	false	Specifies whether OSS checks the MD5 hash of the body of the response returned by the origin. When the value of this parameter is true and the response returned by the origin includes the Content-Md5 header, OSS checks whether the MD5 hash of the obtained data matches the header value. If the MD5 hash of the obtained data does not match the header value, OSS does not store the data. This parameter takes effect only when the value of RedirectType is Mirror. Default value: false Parent nodes: Redirect
MirrorHeaders	Container	No	N/A	Specifies the headers contained in the response that is returned when you use mirroring-based back-to-origin. This parameter takes effect only when the value of RedirectType is Mirror. Parent nodes: Redirect
PassAll	Boolean	No	true	Specifies whether OSS passes through all request headers - except for reserved headers and headers that start with oss-, x-oss-, and x-drs-, such as content-length, authorization2, authorization, range, and date - to the origin. This parameter takes effect only when the value of RedirectType is Mirror. Default value: false Parent nodes: MirrorHeaders
Pass	String	No	myheader-key1	Specifies the headers to pass through to the origin. The header can be up to 1,024 bytes in length and can contain only letters, digits, and hyphens (-). This parameter takes effect only when the value of RedirectType is Mirror. You can specify up to 10 headers. Parent nodes: MirrorHeaders
Remove	String	No	myheader-key3	Specifies the headers that are not allowed to be passed through to the origin. The header can be up to 1,024 bytes in length. The character set of this parameter is the same as that of Pass. This parameter takes effect only when the value of RedirectType is Mirror. You can specify up to 10 headers. This parameter is used together with PassAll. Parent nodes: MirrorHeaders
Set	Container	No	N/A	Specifies the headers that are sent to the origin. The specified headers are configured in the data returned by the origin regardless of whether the headers are contained in the request. You can specify up to 10 header sets. This parameter takes effect only when the value of RedirectType is Mirror. Parent nodes: MirrorHeaders
Key	String	Conditional. This parameter must be specified if Set is specified.	myheader-key5	Specifies the key of the header. The key can be up to 1,024 bytes in length. The character set of this parameter is the same as that of Pass. This parameter takes effect only when the value of RedirectType is Mirror. Parent nodes: Set
Value	String	Conditional. This parameter must be specified if Set is specified.	myheader-value5	Specifies the value of the header. The value can be up to 1,024 bytes in length and cannot contain <code>\r\n</code>. This parameter takes effect only when the value of RedirectType is Mirror. Parent nodes: Set
Protocol	String	Conditional. This parameter must be specified if the value of RedirectType is External or AliCDN.	http	The protocol used for redirection. For example, if you access the object named test, Protocol is set to https, and Hostname is set to <code>www.example.com</code>, the Location header is <code>https://www.example.com/test</code>. This parameter takes effect only when the value of RedirectType is External or AliCDN. Valid values: <code>http</code> and <code>https</code>. Parent nodes: Redirect

Parameter	Type	Required	Example	Description
HostName	String	Conditional. This parameter must be specified if the value of RedirectType is External or AliCDN.	www.example.com	The domain name used for redirection, which must comply with the naming conventions for domain names. For example, if you access the object named test, Protocol is set to https, and Hostname is set to www.example.com, the Location header is https://www.example.com/test. Parent nodes: Redirect
ReplaceKeyPrefixWith	String	No	abc/	Specifies the string that is used to replace the prefix of the object name in the redirect request. If the prefix of the object is empty, this string is added before the object name. Note You can specify only one of the ReplaceKeyWith and ReplaceKeyPrefixWith nodes in a rule. If KeyPrefixEquals is set to abc/ and ReplaceKeyPrefixWith is set to def/, when you access the abc/test.txt object, the Location header is http://www.example.com/def/test.txt. Parent nodes: Redirect
EnableReplacePrefix	Boolean	No	true	If this parameter is set to true, the prefix of object names is replaced with the value specified by ReplaceKeyPrefixWith. If this parameter is not specified or empty, the prefix of object names is truncated. Note When the ReplaceKeyWith parameter is not empty, this parameter cannot be set to true. Default value: false Parent nodes: Redirect
ReplaceKeyWith	String	No	prefix/\${key}.suffix	Specifies the string that is used to replace the requested object name when the request is redirected. This parameter can be a variable. The \${key} variable that indicates the object name in the request is supported. If ReplaceKeyWith is set to prefix/\${key}.suffix, when you access the object named test, the Location header is http://www.example.com/prefix/test.suffix. Parent nodes: Redirect
HttpRedirectCode	HTTP status code	Conditional. This parameter must be specified if the value of RedirectType is External or AliCDN.	301	The HTTP redirect code in the response. This parameter takes effect only when the value of RedirectType is External or AliCDN. Valid values: 301, 302, and 307 Parent nodes: Redirect

For more information about other common request elements of PutBucketWebsite, see [Common request headers](#).

Response elements

For more information about the response elements of PutBucketWebsite, see [Common response headers](#).

Examples

• Sample requests

```
PUT /? website HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 209
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqx*****k53otfjbyc:KU5h8YM*****0dXqf3JxrTZHiA=
<? xml version="1.0" encoding="UTF-8"? >
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
    <SupportSubDir>true</SupportSubDir>
    <Type>0</Type>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

• Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS

```

- Complete sample code

```

PUT /? website HTTP/1.1
Date: Fri, 27 Jul 2018 09:03:18 GMT
Content-Length: 2064
Host: test.oss-cn-hangzhou-internal.aliyuncs.com
Authorization: OSS alnBN*****QMf8u:sNKIHT6ci/z231yIT5vYnetD****
User-Agent: aliyun-sdk-python-test/0.4.0
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
    <SupportSubDir>true</SupportSubDir>
    <Type>0</Type>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <RuleNumber>1</RuleNumber>
      <Condition>
        <KeyPrefixEquals>abc</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <RedirectType>Mirror</RedirectType>
        <PassQueryString>true</PassQueryString>
        <MirrorURL>http://www.test.com</MirrorURL>
        <MirrorPassQueryString>true</MirrorPassQueryString>
        <MirrorFollowRedirect>true</MirrorFollowRedirect>
        <MirrorCheckMd5>false</MirrorCheckMd5>
        <MirrorHeaders>
          <PassAll>true</PassAll>
          <Pass>myheader-key1</Pass>
          <Pass>myheader-key2</Pass>
          <Remove>myheader-key3</Remove>
          <Remove>myheader-key4</Remove>
          <Set>
            <Key>myheader-key5</Key>
            <Value>myheader-value5</Value>
          </Set>
        </MirrorHeaders>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <RuleNumber>2</RuleNumber>
      <Condition>
        <KeyPrefixEquals>abc</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
        <IncludeHeader>
          <Key>host</Key>
          <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
        </IncludeHeader>
      </Condition>
      <Redirect>
        <RedirectType>AliCDN</RedirectType>
        <Protocol>http</Protocol>
        <HostName>www.test.com</HostName>
        <PassQueryString>false</PassQueryString>
        <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
        <HttpRedirectCode>301</HttpRedirectCode>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 27 Jul 2018 09:03:18 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5B5ADFD6ED3CC49176CBE29D
x-oss-server-time: 47

```

1.2.5.5.2. GetBucketWebsite

You can call this operation to view the static website hosting status of a bucket.

Request structure

```
GET /? website HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

For more information about the common request headers involved in the GetBucketWebsite operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
ErrorDocument	Container	N/A	The parent element of the Key child element.
IndexDocument	Container	N/A	The parent element of the Suffix child element.
Key	String	error.html	The file name used when the 404 HTTP status code is returned.
Suffix	String	index.html	The index file name added when a folder URL is returned. This element cannot be empty or contain a slash (/). For example, if the index file is set to index.html, regionid.example.com/mybucket/mydir/ included in a request is converted into regionid.example.com/mybucket/index.html by default.
WebsiteConfiguration	Container	N/A	The container for the request.

Examples

Sample requests

```
Get /? website HTTP/1.1
Host: BucketName.regionid.example.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS q***c:B***=
```

Sample success responses

```
HTTP/1.1 200
x-oss-request-id: 5***B
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<WebsiteConfiguration xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

Sample error responses

```
HTTP/1.1 404
x-oss-request-id: 5***B
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<Error xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Code>NoSuchWebsiteConfiguration</Code>
  <Message>The specified bucket does not have a website configuration. </Message>
  <BucketName>oss-example</BucketName>
  <RequestId>5***F</RequestId>
  <HostId>BucketName.regionid.example.com</HostId>
</Error>
```

1.2.5.5.3. DeleteBucketWebsite

You can call this operation to disable static website hosting for a bucket.

Request structure

```
DELETE /? website HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

Request headers involved in the DeleteBucketWebsite operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

Response headers involved in the DeleteBucketWebsite operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
DELETE /? website HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Authorization: OSS q***:L***=
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-request-id: 5***B
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

1.2.5.6. Hotlink protection

1.2.5.6.1. PutBucketReferer

You can call this operation to configure a Referer whitelist and specify whether to allow a request whose Refer field is empty.

Usage notes

PutBucketReferer replaces the existing configured whitelist with the whitelist specified by RefererList. When empty RefererList that contains no Referer request elements is uploaded, this operation overwrites the configured whitelist. The existing RefererList is deleted.

Request structure

```
PUT /? referer HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Host: BucketName.oss.example.com
Authorization: SignatureValue
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
  <RefererList>
    <Referer> http://www.aliyun.com</Referer>
    <Referer> https://www.aliyun.com</Referer>
    <Referer> http://www. *.com</Referer>
    <Referer> https://www?.aliyuncs.com</Referer>
  </RefererList>
</RefererConfiguration>
```

Request elements

Element	Type	Required	Example	Description
RefererConfiguration	Container	Yes	N/A	The container that contains the Referer configurations. Type: container Child nodes: AllowEmptyReferer and RefererList Parent node: none
AllowEmptyReferer	Boolean	Yes	true	Specifies whether to allow a request whose Refer field is empty. Default value: true
RefererList	Container	Yes	N/A	The container that contains the Referer whitelist.
Referer	String	No	http://www.aliyun.com	The Referer whitelist.

Response headers

The response headers involved in the PutBucketReferer operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests that exclude the Referer element

```
PUT /? referer HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS g***c:K***=
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList />
</RefererConfiguration>
```

Sample requests that include the Referer element

```
PUT /? referer HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS g***c:K***=
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList>
<Referer> http://www.aliyun.com</Referer>
<Referer> https://www.aliyun.com</Referer>
<Referer> http://www. *.com</Referer>
<Referer> https://www?.aliyuncs.com</Referer>
</ RefererList>
</RefererConfiguration>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.5.6.2. GetBucketReferer

You can call this operation to view the Referer configurations of a bucket.

Request structure

```
GET /? referer HTTP/1.1
Host: BucketName.oss.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

For more information about the common request headers involved in the GetBucketReferer operation, see [Common request headers](#).

Response elements

Element	Type	Example	Description
RefererConfiguration	Container	N/A	The container that contains the Referer configurations.
AllowEmptyReferer	Enumerated string	true	Specifies whether to allow a request whose Refer field is empty. Valid values: true and false Default value: true
RefererList	Container	N/A	The container that contains the Referer whitelist.
Referer	String	http://www.aliyun.com	The Referer whitelist.

Examples

Sample requests

```
Get /? referer HTTP/1.1
Host: BucketName.oss.example.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS g***: B***=
```

Sample responses to requests that include the Referer element

```

HTTP/1.1 200
x-oss-request-id: 5***B
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
  <RefererList>
    <Referer>http://www.aliyun.com</Referer>
    <Referer>https://www.aliyun.com</Referer>
    <Referer>http://www. *.com</Referer>
    <Referer>https://www.?.aliyuncs.com</Referer>
  </RefererList>
</RefererConfiguration>

```

Sample responses to requests that exclude the Referer element

```

HTTP/1.1 200
x-oss-request-id: 5***B
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList />
</RefererConfiguration>

```

1.2.5.7. Tags

1.2.5.7.1. PutBucketTags

Adds tags to a bucket or modifies the tags of a bucket. Object Storage Service (OSS) allows you to classify and manage buckets by using bucket tags.

Usage notes

- Only the owner of a bucket and authorized RAM users can configure tags for the bucket. If other users attempt to configure tags for the bucket, 403 Forbidden is returned together with the AccessDenied error code.
- You can configure up to 20 tags for a bucket. A tag is a key-value pair.
- When you call PutBucketTags to add tags to a bucket, the new tags overwrite the existing tags that have the same key.

Request syntax

```

PUT /?tagging HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Authorization: SignatureValue
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
<?xml version="1.0" encoding="UTF-8"?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>key1</Key>
      <Value>value1</Value>
    </Tag>
    <Tag>
      <Key>key2</Key>
      <Value>value2</Value>
    </Tag>
  </TagSet>
</Tagging>

```

Request headers

A PutBucketLifecycle request contains only common request headers. For more information, see [Common request headers](#).

Request elements

Element	Type	Required	Description
Tagging	Container	Yes	The container used to store TagSet. Child nodes: TagSet Parent nodes: none

Element	Type	Required	Description
TagSet	Container	Yes	The container used to store a set of Tags. Child nodes: Tag Parent nodes: Tagging
Tag	Container	Yes	The container used to store the tag that you want to add or modify. Child nodes: Key and Value Parent nodes: TagSet
Key	String	Yes	The key of the tag that you want to add or modify. - A tag key can be up to 64 bytes in length. - A tag key cannot start with <code>http://</code>, <code>https://</code>, or <code>Aliyun</code>. - A tag key must be UTF-8 encoded. - This element cannot be left empty. Child nodes: none Parent nodes: Tag
Value	String	No	The value of the tag that you want to add or modify. - A tag value can be up to 128 bytes in length. - A tag value must be UTF-8 encoded. - This element can be left empty. Child nodes: none Parent nodes: Tag

Response headers

The response to a PutObject request contains only common response headers. For more information, see [Common response headers](#).

Examples

• Sample requests

```
PUT /?tagging
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 20 Dec 2018 11:49:13 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
<Tagging>
  <TagSet>
    <Tag>
      <Key>testa</Key>
      <Value>testv1</Value>
    </Tag>
    <Tag>
      <Key>testb</Key>
      <Value>testv2</Value>
    </Tag>
  </TagSet>
</Tagging>
```

• Sample responses

```
200 (OK)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5C1B138A109F4E405B2D****
date: Thu, 20 Dec 2018 11:59:06 GMT
x-oss-server-time: 148
connection: keep-alive
```

1.2.5.7.2. GetBucketTags

Queries the tags of a bucket.

Request syntax

```
GET /?tagging
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Element	Type	Description

Element	Type	Description
Tagging	Container	The container that stores the returned tags of the bucket. Parent nodes: none  Note If no tags are configured for the bucket, an XML message body is returned in which the Tagging element is empty.
TagSet	Container	The container that stores the returned tags of the bucket. Parent nodes: Tagging
Tag	Container	The container that stores the returned tags of the bucket. Parent nodes: TagSet
Key	String	The key of a tag. Parent nodes: Tag
Value	String	The value of a tag. Parent nodes: Tag

Examples

• Sample requests

```
GET /?tagging
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 20 Dec 2018 13:09:13 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
```

• Sample responses

```
200 (OK)
content-length: 237
server: AliyunOSS
x-oss-request-id: 5C1B2D24B90AD5490CFE368E
date: Thu, 20 Dec 2018 13:12:21 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>testa</Key>
      <Value>value1-test</Value>
    </Tag>
    <Tag>
      <Key>testb</Key>
      <Value>value2-test</Value>
    </Tag>
  </TagSet>
</Tagging>
```

Error codes

Error code	HTTP status code	Description
NoSuchBucket	404	The error message returned because the bucket whose tags you want to query does not exist.
AccessDenied	403	The error message returned because you do not have permissions to query the tags of the bucket. Only the bucket owner can query the tags of the bucket.

1.2.5.7.3. DeleteBucketTags

Deletes the tags of a bucket.

 **Note** If no tag is configured for the bucket or the tags you want to delete do not exist, OSS returns 204.

Request syntax

```
DELETE /?tagging HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

All headers in a DeleteBucketTags request are common request headers. For more information about common request headers, see [Common request headers](#).

Response headers

All headers in the response to a DeleteBucketTags request are common response headers. For more information about common response headers, see [Common response headers](#).

Examples

- Sample requests sent to delete all tags of a bucket

```
DELETE /?tagging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 25 Dec 2018 17:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:6ZVHOehYzxoClyxRydPQs/Cn****
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-request-id: 5C22E0EFD127F6810B1A****
Date: Tue, 25 Dec 2018 17:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

- Sample request sent to delete tags of which the keys are k1 and k2

```
DELETE /?tagging=k1,k2 HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 25 Dec 2018 17:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:6ZVHOehYzxoClyxRydPQs/Cn****
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-request-id: 5C22E0EFD127F6810B1A92A8
Date: Tue, 25 Dec 2018 17:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Error codes

Error code	HTTP status code	Description
NoSuchBucket	404 No Content	The error message returned because the bucket does not exist.
AccessDenied	403 Forbidden	The error message returned because you do not have permissions to delete the tags of the bucket. Only the owner of a bucket can delete the tags of the bucket.

1.2.5.8. CORS

1.2.5.8.1. Overview

Cross-origin resource sharing (CORS) allows web applications to access resources across origins. OSS allows you to use CORS to develop more flexible web applications. OSS provides CORS-related API operations for developers to control cross-origin access.

1.2.5.8.2. PutBucketcors

You can call this operation to configure a cross-origin resource sharing (CORS) rule for a specific bucket. If a CORS rule is configured for the bucket, PutBucketCORS overwrites the existing rule. By default, CORS is disabled for buckets. All cross-origin requests are not allowed.

Usage notes

When you call PutBucketCORS, take note of the following items:

- Disabled-by-default design

By default, CORS is disabled for buckets. All cross-origin requests are not allowed.

- Overwriting mechanism

When you call PutBucketCORS to configure a new CORS rule for a bucket for which a CORS rule is already configured, the existing rule is overwritten.

- Usage in applications

When you use CORS in applications, you must call PutBucketCORS to configure a CORS rule to enable CORS.

For example, if you want to use the `XMLHttpRequest` feature of your browser to access OSS from the domain name `www.a.com`, you must call PutBucketCORS to configure a CORS rule that is described by an XML message body.

- CORS rule matching

When OSS receives a cross-origin request or an OPTIONS request sent for a bucket, OSS reads the CORS rules configured for the bucket and checks the permission of the request. OSS matches the request with the rules one by one. When OSS finds the first match, OSS returns a corresponding header in the response. If no match is found, OSS does not include any CORS header in the response.

The following conditions must be met before OSS determines that a CORS rule matches the request:

- The origin from which the request is initiated must match the value of one `AllowedOrigin` element in the CORS rule.
- The method of the request such as GET or PUT or the method corresponding to the `Access-Control-Request-Method` header in an OPTIONS request must match the value of one `AllowedMethod` element in the CORS rule.
- Each header included in `Access-Control-Request-Headers` in an OPTIONS request must match the value of one `AllowedHeader` element in the CORS rule.

Request structure

```
PUT /? cors HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Host: BucketName.regionid.example.com
Authorization: SignatureValue
<? xml version="1.0" encoding="UTF-8"? >
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>the origin you want allow CORS request from</AllowedOrigin>
    <AllowedOrigin>... </AllowedOrigin>
    <AllowedMethod>HTTP method</AllowedMethod>
    <AllowedMethod>... </AllowedMethod>
    <AllowedHeader> headers that allowed browser to send</AllowedHeader>
    <AllowedHeader>... </AllowedHeader>
    <ExposeHeader> headers in response that can access from client app</ExposeHeader>
    <ExposeHeader>... </ExposeHeader>
    <MaxAgeSeconds>time to cache pre-flight response</MaxAgeSeconds>
  </CORSRule>
  <CORSRule>
    ...
  </CORSRule>
  ...
</CORSConfiguration >
```

Request elements

Element	Type	Required	Example	Description
CORSRule	Container	Yes	N/A	The container for CORS rules. A maximum of 10 rules can be configured for each bucket.
AllowedOrigin	String	Yes	*	The origin from which a request is allowed. You can use multiple elements to specify multiple origins. Each rule can contain only one asterisk (*) as the wildcard. If AllowedOrigin is set to an asterisk (*), all cross-origin requests are allowed.
AllowedMethod	Enumerated string	Yes	GET	The cross-origin request methods that are allowed. Valid values: GET, PUT, DELETE, POST, and HEAD
AllowedHeader	String	No	Authorization	Specifies whether the headers specified by Access-Control-Request-Headers in the preflight (OPTIONS) request are allowed. Each header specified by Access-Control-Request-Headers must match the value of an AllowedHeader element. Each rule can contain only one asterisk (*) as the wildcard.
ExposeHeader	String	No	x-oss-test	The list of headers that can be exposed to the browser. The headers are the response headers that allow access from an application such as XMLHttpRequest in JavaScript. Asterisks (*) cannot be included as wildcards in the value of this parameter.
CORSConfiguration	Container	Yes	N/A	The container that stores the CORS rules configured for a bucket.
MaxAgeSeconds	Integer	No	100	The time the browser can cache the response to a preflight (OPTIONS) request to a specific resource. Unit: seconds. Only one MaxAgeSeconds element can be configured for each CORS rule.

Response headers

The response headers involved in the PutBucketcors operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
PUT /? cors HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS g***A=
<? xml version="1.0" encoding="UTF-8"? >
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>Authorization</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>http://www.a.com</AllowedOrigin>
    <AllowedOrigin>http://www.b.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader> Authorization</AllowedHeader>
    <ExposeHeader>x-oss-test</ExposeHeader>
    <ExposeHeader>x-oss-test1</ExposeHeader>
    <MaxAgeSeconds>100</MaxAgeSeconds>
  </CORSRule>
</CORSConfiguration >
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***F
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.5.8.3. GetBucketcors

You can call this operation to query the existing CORS rules of a specified bucket.

Request structure

```
GET /? cors HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

The request headers involved in the GetBucketcors operation contain only common request headers. For more information, see [Common request headers](#).

Response elements

Element	Type	Example	Description
CORSRule	Container	N/A	The container that stores CORS rules. Up to 10 rules can be configured for a bucket.
AllowedOrigin	String	*	The sources from which cross-origin requests are allowed. If AllowedOrigin is set to an asterisk (*), cross-origin requests from all sources are allowed.
AllowedMethod	Enumerated string	GET	The cross-origin request methods that are allowed. Valid values: GET, PUT, DELETE, POST, and HEAD
AllowedHeader	String	*	Indicates whether the headers specified by Access-Control-Request-Headers in the OPTIONS preflight request are allowed. Each header specified by Access-Control-Request-Headers must match the value of an AllowedHeader element.
ExposeHeader	String	x-oss-test	The response headers for allowed access requests from applications, such as an XMLHttpRequest object in JavaScript.
MaxAgeSeconds	Integer	100	The period of time within which the browser can cache the response for an OPTIONS preflight request to specific resources. A CORS rule can contain only one MaxAgeSeconds parameter.
CORSConfiguration	Container	N/A	The container that stores the CORS rules configured for a bucket.

Examples

Sample requests

```
Get /? cors HTTP/1.1
Host: ***.regionid.example.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS q***w=
```

Sample responses if CORS rules exist

```

HTTP/1.1 200
x-oss-request-id: 5***F
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>*/</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <ExposeHeader>x-oss-test</ExposeHeader>
    <MaxAgeSeconds>100</MaxAgeSeconds>
  </CORSRule>
</CORSConfiguration>

```

1.2.5.8.4. DeleteBucketcors

You can call this operation to delete CORS rules.

Request structure

```

DELETE /? cors HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue

```

Request headers

Request headers involved in the DeleteBucketcors operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

Response headers involved in the DeleteBucketcors operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```

DELETE /? cors HTTP/1.1
Host: *.regionid.example.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Authorization: OSS q***g=

```

Sample responses

```

HTTP/1.1 204 No Content
x-oss-request-id: 5***C
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS

```

1.2.5.8.5. OptionObject

Before a cross-origin request is sent, the browser sends a preflight (OPTIONS) request that includes a specific origin, HTTP method, and header information to OSS to determine whether to send the cross-origin request.

OSS allows you to call the PutBucketcors operation to enable CORS for buckets. OSS determines whether to allow the preflight request of the browser based on the specified rules after CORS is enabled. If OSS rejects this request or CORS is disabled, 403 Forbidden is returned.

Request structure

```

OPTIONS /ObjectName HTTP/1.1
Host: BucketName.regionid.example.com
Origin:Origin
Access-Control-Request-Method:HTTP method
Access-Control-Request-Headers:Request Headers

```

Request headers

Header	Type	Required	Example	Description
Origin	String	Yes	http://www.example.com	The origin of the request, used to identify a cross-origin request. Default value: null

Header	Type	Required	Example	Description
Access-Control-Request-Method	String	Yes	PUT	The method to use in the cross-origin request. Default value: null
Access-Control-Request-Headers	String	Yes	x-oss-test	The request headers, except for standard headers, to use in the cross-origin request. Default value: null

Response headers

Header	Type	Example	Description
Access-Control-Allow-Origin	String	http://www.example.com	The origin that is included in the request. If the request is denied, the response does not contain the header.
Access-Control-Allow-Methods	String	PUT	The HTTP method of the request. If the request is denied, the response does not contain the header.
Access-Control-Allow-Headers	String	x-oss-test	The list of headers included in the request. If the request includes headers that are not allowed, the response does not contain the headers and the request is denied.
Access-Control-Expose-Headers	String	Content-Length	The list of headers that are accessible to JavaScript applications on a client.
Access-Control-Max-Age	Integer	100	The maximum duration for the browser to cache preflight results. Unit: seconds.

Examples

Sample requests

```
OPTIONS /testobject HTTP/1.1
Host: *.regionid.example.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Origin:http://www.example.com
Access-Control-Request-Method:PUT
Access-Control-Request-Headers:x-oss-test
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***C
Date: Fri, 24 Feb 2012 05:45:34 GMT
Access-Control-Allow-Origin: http://www.example.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-oss-test
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

1.2.5.9. Versioning

1.2.5.9.1. PutBucketVersioning

Configures the versioning state of a bucket.

Usage notes

- Before you configure the versioning state of a bucket, you must have the PutBucketVersioning permission.
- A bucket can be in one of the following versioning state: disabled, enabled, and suspended. By default, versioning is disabled for a bucket.
- If versioning is enabled for a bucket, OSS generates unique version IDs for all objects that are added to the bucket. In this case, OSS stores multiple versions of the objects.
- If versioning is suspended for a bucket, OSS generates the version ID null for all objects that are added to the bucket. In this case, OSS does not store new previous versions for objects that is deleted or overwritten.

Request syntax

```
PUT /?versioning HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Examples

- Sample requests for enabling versioning for a bucket

```
PUT /?versioning HTTP/1.1
Host: bucket-versioning.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:20:12 GMT
Authorization: OSS e7thre3jj5mlvqk:12ztptkaR8a74gIGFzOaZZQe****
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC015CB7AEADE01700****
Date: Tue, 09 Apr 2019 02:20:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

• Sample requests for suspending versioning for a bucket

```
PUT /?versioning HTTP/1.1
Host: bucket-versioning.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:28:18 GMT
Authorization: OSS m2qa99e9tpkaehr:DWAzr2EkqDwFJNke1Nuaogn7****
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Suspended</Status>
</VersioningConfiguration>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC0342B7AEADE01700****
Date: Tue, 09 Apr 2019 02:28:18 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Error codes

Error code	HTTP status code	Description
AccessDenied	403	The error message returned because you do not have permissions to configure the versioning state of the bucket.
InvalidArgument	400	The error message returned because the versioning state that you want to configure is invalid. You can set the versioning state of a bucket to only Enabled or Suspended.

1.2.5.9.2. GetBucketVersioning

Queries the versioning state of a bucket.

Request syntax

```
GET /?versioning HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Element	Type	Description
VersioningConfiguration	Container	The container that stores the versioning state of the bucket. Child nodes: Status Parent nodes: none
Status	String	The versioning state of the bucket. Parent nodes: VersioningConfiguration Valid values: - Enabled: indicates that versioning is enabled for the bucket. - Suspended: indicates that versioning is suspended for the bucket.

Examples**Sample requests**

```
GET /?versioning HTTP/1.1
Host: bucket-versioning.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:28:18 GMT
Authorization: OSS pu4kyewh6qk6nut:2TeHbmWY36GwdwS0JFiRb+NV****
```

Sample responses

- Sample response for a bucket for which versioning is enabled

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC0342B7AEADE01700****
Date: Tue, 09 Apr 2019 02:28:18 GMT
Content-Length: 121
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

- Sample response for a bucket for which versioning is disabled

If versioning is disabled for the requested bucket, the versioning state of the bucket is not contained in the XML body of the response.

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC015CB7AEADE01700****
Date: Tue, 09 Apr 2019 02:20:12 GMT
Content-Length: 74
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<VersioningConfiguration xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com"/>
```

Error codes

Error code	HTTP status code	Description
AccessDenied	403	The error message returned because you do not have permissions to query the versioning state of the bucket. Only the bucket owner or RAM users that have the GetBucketVersioning permission can query the versioning state of a bucket.
NoSuchBucket	404	The error message returned because the specified bucket does not exist.

1.2.5.9.3. GetBucketVersions(ListObjectVersions)

Lists the versions of all objects and delete markers in a bucket.

Usage notes

If you call the GetBucketVersions(ListObjectVersions) and GetBucket(ListObjects) operations on a versioned bucket, the following different results are returned:

- In the response to a GetBucket(ListObjects) request, the current versions of objects in the bucket are returned, excluding delete markers.
- In the response to a GetBucketVersions(ListObjectVersions) request, all versions of objects in the bucket are returned, including delete markers.

Objects are returned in alphabetic order, but the versions of each object are returned by creation time in descending order.

Request headers

A GetBucketVersions(ListObjectVersions) request contains only common request headers such as `Authorization` and `Host`. For more information, see [Common request headers](#).

Request parameters

When you call the GetBucketVersions(ListObjectVersions) operation, you can specify the following parameters to filter the returned results: prefix, key-marker, version-id-marker, delimiter, and max-keys.

Parameter	Type	Required	Example	Description
delimiter	String	No	/	The character that is used to group objects by name. If you specify the delimiter parameter in the request, the response contains the CommonPrefixes element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. If you set prefix to a directory name and delimiter to a forward slash (/), only objects in the directory are returned. The names of the subdirectories within the directory are returned in the CommonPrefixes element. However, objects and directories within the subdirectories are not listed. This parameter is empty by default.
key-marker	String	The key-marker parameter is required if the version-id-marker parameter is specified.	example	Specifies that objects whose names are alphabetically after the value of the key-marker parameter are returned. This parameter can be specified together with version-id-marker. The value of key-marker must be less than 1,024 bytes in length. This parameter is empty by default.

Parameter	Type	Required	Example	Description
version-id-marker	String	No	CAEQMxiBgICbof2D0BYiGRhZjg wMzjiMjA3MjQ0ODE5MWYxZDY wMzjIZjU1****	Specifies that the versions created before the version specified by version-id-marker for the object whose name is specified by key-marker are returned by creation time in descending order. By default, if this parameter is not specified, the results are returned from the latest version of the object whose name is alphabetically after the value of key-marker. This parameter is empty by default. Valid values: version IDs
max-keys	String	No	100	The maximum number of objects to return. If the number of returned objects exceeds the value of the max-keys parameter, the response contains the <code>NextKeyMarker</code> and <code>NextVersionIdMarker</code> elements as the markers for the next <code>GetBucketVersions(ListObjectVersions)</code> request. The values of <code>NextKeyMarker</code> and <code>NextVersionIdMarker</code> are included in the response. Valid values: 1 to 1000 Default value: 100.
prefix	String	No	fun	The prefix that the names of returned objects must contain. - The value of the prefix parameter must be less than 1,024 bytes in length. - If you specify a prefix in the request, the names of the returned objects contain the prefix. If you set prefix to a directory name in the request, the objects whose names contain the prefix are listed, including all objects and subdirectories within the directory. This parameter is empty by default.
encoding-type	String	No	URL	The encoding type of the content to return. This parameter is empty by default. Valid value: URL  Notice The values of delimiter, marker, prefix, NextMarker, and Key are UTF-8 encoded. If the value of delimiter, marker, prefix, NextMarker, or Key contains a control character that is not supported by the XML 1.0 standard, you can specify encoding-type to encode the value in the response.

Response elements

Element	Type	Example	Description
ListVersionsResult	Container	N/A	The container that stores the response to the <code>GetBucketVersions(ListObjectVersions)</code> request. Child nodes: Name, Prefix, Marker, MaxKeys, Delimiter, IsTruncated, NextMarker, Version, and DeleteMarker Parent nodes: none
CommonPrefixes	String	N/A	If the delimiter parameter is specified in the request, the response contains the <code>CommonPrefixes</code> element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in <code>CommonPrefixes</code> . Parent nodes: ListVersionsResult
Delimiter	String	/	The character that is used to group objects by name. If you specify the delimiter parameter in the request, the response contains the <code>CommonPrefixes</code> element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in <code>CommonPrefixes</code> . Parent nodes: ListVersionsResult
EncodingType	String	URL	The encoding type of the content in the response. If you specify encoding-type in the request, the values of Delimiter, Marker, Prefix, NextMarker, and Key are encoded in the response. Parent nodes: ListVersionsResult

Element	Type	Example	Description
IsTruncated	String	true	Indicates whether the returned results are truncated. - true: indicates that not all of the results are returned for the request. - false: indicates that all of the results are returned for the request. Valid values: true and false Parent nodes: ListVersionsResult
KeyMarker	String	example	Indicates the object from which the GetBucketVersions(ListObjectVersions) operation starts. Parent nodes: ListVersionsResult
VersionIdMarker	String	CAEQMxiBgICbof2D0BYiIGRhZjgwMzJiMjA3MjQ0ODE5MWYxZDYwMzJlZjU1****	This element is returned together with KeyMarker to indicate the version from which the GetBucketVersions(ListObjectVersions) operation starts. Parent nodes: ListVersionsResult
NextKeyMarker	String	test	If not all of the results are returned for the request, the NextKeyMarker element is contained in the response to indicate the key-marker value of the next GetBucketVersions(ListObjectVersions) request. Parent nodes: ListVersionsResult
NextVersionIdMarker	String	CAEQGBiBgIC_jq7P9xYiIDRiZWJkNjY2Y2Q4NDQ5ZTl5ZGE5ODIxMTIyZThl**	If not all of the results are returned for the request, the NextVersionIdMarker element is contained in the response to indicate the version-id-marker value of the next GetBucketVersions(ListObjectVersions) request. Parent nodes: ListVersionsResult
MaxKeys	String	1000	The maximum number of returned objects in the response. Parent nodes: ListVersionsResult
Name	String	examplebucket-1250000000	The name of the bucket. Parent nodes: ListVersionsResult
Owner	Container	N/A	The container that stores the information about the bucket owner. Parent nodes: ListVersionsResult
Prefix	String	fun	The prefix that must be included in the names of the listed objects. Parent nodes: ListVersionsResult
Version	Container	N/A	The container that stores the versions of objects, excluding delete markers. Parent nodes: ListVersionsResult
DeleteMarker	Container	N/A	The container that stores delete markers. Parent nodes: ListVersionsResult
ETag	String	250F8A0AE989679A22926A875F0A2****	The entity tag (ETag) that is generated when an object is created. ETags are used to identify the content of objects. - For an object that is created by calling the PutObject operation, the ETag value of the object is the MD5 hash of the object content. - For an object that is not created by calling the PutObject operation, the ETag value of the object is the UUID of the object content.  Note The ETag value of an object can be used only to check whether the object content is modified. We recommend that you use the MD5 hash of an object rather than the ETag value of the object to verify data integrity. Parent nodes: ListVersionsResult.Version
Key	String	example	The name of the object. Parent nodes: ListVersionsResult.Version and ListVersionsResult.DeleteMarker
LastModified	Time	2019-04-09T07:27:28.000Z	The time when the object was last modified. Parent nodes: ListVersionsResult.Version and ListVersionsResult.DeleteMarker
VersionId	String	CAEQMxiBgMDNoP2D0BYiIDe3MWUxNzgxDQxNTRIODl5OGYwZGMwNGY3MzZjN****	The version ID of the object. Parent nodes: ListVersionsResult.Version and ListVersionsResult.DeleteMarker

Element	Type	Example	Description
IsLatest	String	true	Indicates whether the version is the current version. Valid values: - true: The version is the current version. - false: The version is a previous version. Parent nodes: ListVersionsResult.Version and ListVersionsResult.DeleteMarker
Size	String	93731	The size of the returned object. Unit: bytes. Parent nodes: ListVersionsResult.Version and ListVersionsResult.DeleteMarker
StorageClass	String	Standard	The storage class of the object. Parent nodes: ListVersionsResult.Version and ListVersionsResult.DeleteMarker
DisplayName	String	12345125285864390	The name of the object owner. Parent nodes: ListVersionsResult.Version.Owner and ListVersionsResult.DeleteMarker.Owner
ID	String	1234512528586****	The user ID of the bucket owner. Parent nodes: ListVersionsResult.Version.Owner and ListVersionsResult.DeleteMarker.Owner

For more information about the common response headers contained in the response to a `GetBucketVersions(ListObjectVersions)` request such as `x-oss-request-id` and `Content-Type`, see [Common response headers](#).

Examples

- List the versions of objects in an unversioned bucket

Sample requests

```
GET /?versions HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 07:27:48 GMT
Authorization: OSS ami4tq0x76ov9cu:Wfx4*****e7Rc0jawCsh7h1k****
```

Sample responses

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1262
Connection: keep-alive
Date: Thu, 09 Apr 2019 07:27:48 GMT
Server: AliyunOSS
x-cos-request-id: 534B371674E88A4D8906****
<ListVersionsResult>
  <Name>examplebucket-1250000000</Name>
  <Prefix/>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>example-object-1.jpg</Key>
    <VersionId/>
    <IsLatest>true</IsLatest>
    <LastModified>2019-08-5T12:03:10.000Z</LastModified>
    <ETag>5B3C1A2E053D763E1B669CC607C5A0FE1****</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
  </Version>
  <Version>
    <Key>example-object-2.jpg</Key>
    <VersionId/>
    <IsLatest>true</IsLatest>
    <LastModified>2019-08-9T12:03:09.000Z</LastModified>
    <ETag>5B3C1A2E053D763E1B002CC607C5A0FE1****</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
  </Version>
  <Version>
    <Key>example-object-3.jpg</Key>
    <VersionId/>
    <IsLatest>true</IsLatest>
    <LastModified>2019-08-10T12:03:08.000Z</LastModified>
    <ETag>4B3F1A2E053D763E1B002CC607C5AGTRF****</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
  </Version>
</ListVersionsResult>

```

- List the versions of objects in a versioned bucket

In this example, two objects named example and pic.jpg are stored in a bucket named oss-example. The object named example has the following three versions that are displayed by creation time in descending order: 111222, 000123 (delete marker), and 222333. The object named pic.jpg has only one version whose ID is 232323.

If you set key-marker to example and version-id-marker to 111222, the following three versions are returned in sequence: 000123 of example, 222333 of example, and 232323 of pic.jpg.

Sample requests

```

GET /?versions&key-marker=example&version-id-marker=CAEQMxiBgICbof2D0BYiIGRhZjgwMzJiMjA3MjQ0ODE5MWYxZDYwMzJlZjU1**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 07:27:48 GMT
Authorization: OSS ami4tq0x76o****:Wfx4kLpx+e7Rc0jawCsh7h1k****

```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 5CAC4974B7AEADE01700****
Date: Tue, 09 Apr 2019 07:27:48 GMT
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Name>oss-example</Name>
  <Prefix></Prefix>
  <KeyMarker>example</KeyMarker>
  <VersionIdMarker>CAEQMxiBgICbof2D0BYiIGRhZjgwMzJmMjA3MjQ0DE5MWYxZDYwMzJlZjU1****</VersionIdMarker>
  <MaxKeys>100</MaxKeys>
  <Delimiter></Delimiter>
  <IsTruncated>false</IsTruncated>
  <DeleteMarker>
    <Key>example</Key>
    <VersionId>CAEQMxiBgICaof2D0BYiIDJhMGE3N2M1YTI1NDQzOGY5NTkyNTI3MGYyMzJm****</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2019-04-09T07:27:28.000Z</LastModified>
    <Owner>
      <ID>1234512528586****</ID>
      <DisplayName>12345125285864390</DisplayName>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>example</Key>
    <VersionId>CAEQMxiBgMDNoP2D0BYiIDE3MmUxNzgxZDQxNTRiODI5OGYwZGMwNGY3MzJm****</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2019-04-09T07:27:28.000Z</LastModified>
    <ETag>"250F8A0AE989679A22926A875F0A2****"</ETag>
    <Type>Normal</Type>
    <Size>93731</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>1234512528586****</ID>
      <DisplayName>12345125285864390</DisplayName>
    </Owner>
  </Version>
  <Version>
    <Key>pic.jpg</Key>
    <VersionId>CAEQMxiBgMCZov2D0BYiIDY4MDllOTc2YmY5MjQxMzdiOGI3OTlhNTU0ODIx****</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2019-04-09T07:27:28.000Z</LastModified>
    <ETag>"3663F7B0B9D3153F884C821E7CF4****"</ETag>
    <Type>Normal</Type>
    <Size>574768</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>1234512528586****</ID>
      <DisplayName>12345125285864390</DisplayName>
    </Owner>
  </Version>
</ListVersionsResult>

```

Error codes

Error code	HTTP status code	Description
NoSuchBucket	404	The error message returned because the bucket that you want to access does not exist. This error message is also returned when the bucket that you want to access cannot be created because the bucket name does not comply with the naming conventions.
AccessDenied	403	The error message returned because you are not authorized to access the bucket.
InvalidArgument	400	- The error message returned because the value of max-keys is less than 0 or greater than 1000. - The error message returned because the length of the specified prefix, key-marker, or delimiter value is invalid. - The error message returned because the value of version-id-marker is invalid.

1.2.6. Object-relevant operations

1.2.6.1. Basic operations

1.2.6.1.1. PutObject

Uploads objects.

Usage notes

- The object that you want to upload cannot be larger than 5 GB in size.
- By default, if you upload an object that has the same name as that of an accessible existing object, the existing object is overwritten by the uploaded object and 200 OK is returned.
- OSS does not use directories. All elements are stored as objects. You can create a simulated directory by creating an object whose name ends with a forward slash (/) and that is 0 KB in size.
- To ensure data integrity, OSS provides multiple methods to verify the MD5 hash of the data that you upload. To perform MD5 verification based on the Content-MD5 header, add the header to the request. When the Content-MD5 header value is specified in a request, OSS calculates the MD5 hash of the uploaded data and compares the calculated MD5 hash with the Content-MD5 value specified in the request. If the two values do not match, the operation fails and 400 Bad Request is returned.

Versioning

If you call the PutObject operation to upload an object to a versioning-enabled bucket, OSS generates a unique version ID for the uploaded object and returns the version ID as the x-oss-version-id header in the response.

If you call the PutObject operation to upload an object to a versioning-suspended bucket, OSS generates a version ID of null for the uploaded object. An object can have only one version whose ID is null.

Request syntax

```
PUT /ObjectName HTTP/1.1
Content-Length: ContentLength
Content-Type: ContentType
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

Header	Type	Required	Example	Description
Cache-Control	String	No	no-cache	The web page caching behavior that is performed when the object is downloaded. For more information, visit RFC 2616 .
Content-Disposition	String	No	attachment;filename=oss_download.jpg	The name of the object when the object is downloaded. For more information, visit RFC 2616 .
Content-Encoding	String	No	utf-8	The content encoding format of the object when the object is downloaded. For more information, visit RFC 2616 . This parameter is empty by default.
Content-MD5	String	No	uk80s/oShU2MMhcajpeAVA==	The Content-MD5 value of the message content (excluding headers). The value is obtained by calculating the 128-bit hash value and then encoding the value in Base64 based on RFC 1864. This request header can be used to check the message validity. The message content is valid if the received message content is the same as the content that is sent. Although this request header is optional, we recommend that you use it for end-to-end checks.
Expires	String	No	Fri, 28 Feb 2012 05:38:42 GMT	The expiration time of the request. For more information, visit RFC 2616 .  Note OSS does not limit or verify this value.
x-oss-server-side-encryption	String	No	KMS	The method that is used to encrypt the object on the OSS server when the object is created. Valid values: <i>AES256</i> and <i>KMS</i> If you specify this header, it is returned in the response and the uploaded object is encrypted by using the method specified by this header. When you download the encrypted object, the x-oss-server-side-encryption header is included in the response and the header value is set to the algorithm used to encrypt the object.
x-oss-server-side-data-encryption	String	No	SM4	The algorithm used to encrypt objects. If this header is not specified, objects are encrypted by using AES-256. This parameter is valid only when x-oss-server-side-encryption is set to KMS. Valid values: <i>SM4</i>
x-oss-server-side-encryption-key-id	String	No	3dea0cae-5fd6-4bf8-8574-87c557*****	The ID of the customer master key (CMK) that is managed by Key Management Service (KMS). This parameter is valid only when x-oss-server-side-encryption is set to KMS.
x-oss-object-acl	String	No	private	The access control list (ACL) of the object. Valid values: <i>public-read</i> , <i>private</i> , and <i>public-read-write</i>

Header	Type	Required	Example	Description
x-oss-tagging	String	No	xx=1	The object tag. You can configure multiple tags for the object. Example: TagA=A&TagB=B.  Note The key and value of a tag must be URL-encoded. If a tag does not contain an equal sign (=), the value of the tag is considered an empty string.

Response headers

All headers in the response to a PutObject request are common response headers. For more information about common response headers, see [Common response headers](#).

Examples

- Upload an object to an unversioned bucket

Sample requests

```
PUT /oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Cache-control: no-cache
Expires: Fri, 28 Feb 2012 05:38:42 GMT
Content-Encoding: utf-8
Content-Disposition: attachment;filename=oss_download.jpg
Date: Fri, 24 Feb 2012 06:03:28 GMT
Content-Type: image/jpeg
Content-Length: 344606
Authorization: OSS q***:k***=
[344606 bytes of object data]
```

Sample responses

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Sat, 21 Nov 2015 18:52:34 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5***A
x-oss-bucket-version: 1***9
ETag: "A***2"
```

- Upload an object to a versioned bucket

Sample requests

```
PUT /test HTTP/1.1
Content-Length: 362149
Content-Type: text/html
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 02:53:24 GMT
Authorization: OSS lkojgn8ylexic6e:6yYhX+BuuEqzIltAMW0wgIyl****
```

Sample responses

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 09 Apr 2019 02:53:24 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5CAC0A3DB7AEADE01700****
x-oss-version-id: CAEQNhIbGMDJgZCA0BYiIDc4MGZjZGI2OTBjOTRmNTE5NmU5NmFhZjhjYmY0****
ETag: "4F345B1F066DB1444775AA97D5D2****"
```

1.2.6.1.2. GetObject

Obtains or queries an object. Before you call this operation, you must have read permissions on the object you want to obtain or query.

Limits

- You can specify the Range parameter in GetObject requests to implement resumable download. We recommend that you specify this parameter when you download large objects.
- If the Range parameter is specified in the request, the response contains the length of the entire object and the range of bytes returned for this operation. For example, Content-Range: bytes 0-9/44 indicates that the length of the entire object is 44 bytes and the range of bytes returned in response to the request is from byte 0 to byte 9. If the value of Range is invalid, the entire object is returned and the response returned by OSS does not include Content-Range.
- Anonymous users are not allowed to customize parameters in GetObject requests to specify the headers to be returned in the response.
- If this object is encrypted by using a server-side encryption algorithm based on entropy encoding, OSS automatically returns the decrypted object and x-oss-server-side-encryption in the response when OSS receives the GetObject request. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm of the object.
- To obtain an object compressed in the GZIP format, add Accept-Encoding:gzip to the display mode in the request. OSS determines whether to return data compressed in the GZIP format based on the Content-Type value and size of the object. If the object is compressed in the GZIP format, the response returned by OSS does not include the ETag value of the object. OSS supports GZIP for objects whose Content-Type header is set to one of the following values: HTML,

JavaScript, CSS, XML, RSS, and JSON. The object must be at least 1 KB in size.

Versioning

By default, only the current version of an object is returned after `GetObject` is called.

If the version ID of the object is specified in the request, OSS returns the specified version of the object. If the version ID is set to null in the request, OSS returns the version of the object whose version ID is null.

Request syntax

```
GET /ObjectName HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
Range: bytes=ByteRange (optional)
```

Request parameters

You can customize parameters in a signed `GetObject` request to specify the headers to be returned in the response. The following table describes the parameters.

Parameter	Type	Required	Example	Description
response-content-type	String	No	text	The content-type header in the response.
response-content-language	String	No	en	The content-language header in the response.
response-expires	String	No	Fri, 24 Feb 2012 17:00:00 GMT	The expires header in the response.
response-cache-control	String	No	No-cache	The cache-control header in the response. Type: string. This parameter is empty by default.
response-content-disposition	String	No	attachment; filename:testing.txt	The content-disposition header in the response. Type: string. This parameter is empty by default.
response-content-encoding	String	No	utf-8	The content-encoding header in the response. Type: string. This parameter is empty by default.

Request headers

Header	Type	Required	Example	Description
Range	String	No	bytes=100-900	The range of data of the object that you want to obtain. For example, if you specify bytes=0-9, OSS transfers byte 0 to byte 9, a total of 10 bytes.
If-Modified-Since	String	No	Fri, 13 Nov 2015 14:47:53 GMT	The object transfer condition. If the specified time is earlier than the actual modified time of the object, the system transfers the object and returns 200 OK. Otherwise, the system returns 304 Not Modified. The time must be in GMT.
If-Unmodified-Since	String	No	Fri, 13 Nov 2015 14:47:53 GMT	The object transfer condition. If the specified time is later than or equal to the actual modified time of the object, OSS transfers the object and returns 200 OK. Otherwise, OSS returns 412 Precondition Failed. The time must be in GMT.
If-Match	String	No	5B3C1A2E0563E1B002CC607C***	The object transfer condition. If the specified ETag value of the object matches the actual ETag value, OSS transfers the object and returns 200 OK. Otherwise, OSS returns 412 Precondition Failed.
If-None-Match	String	No	5B3C1A2E0563E1B002CC607C***	The object transfer condition. If the specified ETag value does not match the actual ETag value, OSS transfers the object and returns 200 OK. Otherwise, OSS returns 304 Not Modified. Type: string. This parameter is empty by default.

Response headers

If the requested object is a symbolic link, the content of the object is returned. Response headers `Content-Length`, `ETag`, and `Content-Md5` indicate the metadata of the requested object. The `Last-Modified` header indicates the later one of the time points when the requested object and the symbolic link are last modified. All other headers indicate the metadata of the symbolic link.

Header	Type	Example	Description
<code>x-oss-server-side-encryption</code>	String	<code>x-oss-server-side-encryption</code>	If the requested object is encrypted by using a server-side encryption algorithm based on entropy encoding, OSS automatically decrypts the object and returns the decrypted object after OSS receives the <code>GetObject</code> request. OSS includes <code>x-oss-server-side-encryption</code> in the response to indicate the encryption algorithm that is used to encrypt the object on the server.
<code>x-oss-tagging-count</code>	String	10	The number of tags added to the object. This header is returned only if you have permissions to query the tags of the object.

Examples

- Simple `GetObject` requests

Sample requests

```
GET /oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 06:38:30 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJkde6OhZ9J*****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 3a8f-2e2d-7965-3ff9-51c875b*****
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 06:38:30 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E0563E1B002CC607C*****"
Content-Type: image/jpeg
Content-Length: 344606
Server: AliyunOSS
[344606 bytes of object data]
```

- `GetObject` requests that include the `Range` parameter

Sample requests

```
GET /oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 28 Feb 2012 05:38:42 GMT
Range: bytes=100-900
Authorization: OSS qn6qrrqxo2oawuk5jbyc:qZzjF3DUtd+yK16BdhGtFcC*****
```

Sample responses

```
HTTP/1.1 206 Partial Content
x-oss-request-id: 28f6-15ea-8224-234e-c0ce407*****
x-oss-object-type: Normal
Date: Fri, 28 Feb 2012 05:38:42 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E05E1B002CC607C*****"
Accept-Ranges: bytes
Content-Range: bytes 100-900/344606
Content-Type: image/jpeg
Content-Length: 801
Server: AliyunOSS
[801 bytes of object data]
```

- `GetObject` requests that specify custom response headers

Sample requests

```
GET /oss.jpg?response-expires=Thu%2C%2001%20Feb%202012%2017%3A00%3A00%20GMT& response-content-type=text&response-cache-control=No-cache&response-content-disposition=attachment%253B%2520filename%253Dtesting.txt&response-content-encoding=utf-8&response-content-language=%E4%B8%AD%E6%96%87 HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 06:09:48 GMT
```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC75A644*****
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 06:09:48 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D1B002CC607*****"
Content-Length: 344606
Connection: keep-alive
Content-disposition: attachment; filename:testing.txt
Content-language: en
Content-encoding: utf-8
Content-type: text
Cache-control: no-cache
Expires: Fri, 24 Feb 2012 17:00:00 GMT
Server: AliyunOSS
[344606 bytes of object data]

```

- GetObject request sent to query an object for which the version ID is specified

Sample requests

```

GET /example?versionId=CAEQNhIBgMDJgZCA0BYiIDc4MGZjZGI2OTBjOTRmNTE5NmU5NmFhZjhjYmY0**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 02:58:06 GMT
Authorization: OSS lkojgxic6e:8wcOrEDt4iSxpBPfQW9OJNw*****

```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 5CAC0A3EDE0170*****
x-oss-version-id: CAEQNhIBgM0BYiIDc4MGZjZGI2OTBjOTRmNTE5NmU5NmFhZjhjYmY*****
x-oss-object-type: Normal
Date: Tue, 09 Apr 2019 02:58:06 GMT
Last-Modified: Fri, 22 Mar 2018 08:07:50 GMT
ETag: "5B3C1A2E053D7002CC607C5A*****"
Content-Type: text/html
Content-Length: 362149
Server: AliyunOSS
[362149 bytes of object data]

```

- GetObject requests sent to query an object for which the version ID is not specified and of which the current version is a delete marker

Sample requests

```

GET /example HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 03:22:33 GMT
Authorization: OSS duagpvtn35:taVLDvAJMhEumrR+oLMwtQp*****

```

Sample responses

```

HTTP/1.1 404 Not Found
x-oss-request-id: 5CAC0FEADE0170*****
x-oss-delete-marker: true
x-oss-version-id: CAEQNxIBgyA0BYiIDc4ZDdmNTA2MG5iZTRiNjE5NzZlZW40WM5OT*****
Date: Tue, 09 Apr 2019 03:22:33 GMT
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The specified key does not exist.</Message>
  <RequestId>5CAC0FEADE0170*****</RequestId>
  <HostId>versioning-get.oss-cn-hangzhou.aliyun*****</HostId>
  <Key>example</Key>
</Error>

```

- GetObject requests sent to query an object for which the version ID is specified and of which the version is a delete marker

Sample requests

```

GET /example?versionId=CAEQMxiBgMCFqaWA0BYiIDliMwi4MGQ0MTVmMjQ3MmE5MDNlMmY4YmFkYTk3**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 03:09:44 GMT
Authorization: OSS tvqm50uz4y:51UaP+wQt5klRQang/U6Eeq*****

```

Sample responses

```

HTTP/1.1 405 Method Not Allowed
x-oss-request-id: 5CAC0CF8DE01700*****
x-oss-delete-marker: true
x-oss-version-id: CAEQMxiBgMcfqaWADliMwI4MGQ0MTVnMjQ3MmE5MDNlMmY4YmFkYTk*****
Allow: DELETE
Date: Tue, 09 Apr 2019 03:09:44 GMT
Content-Type: application/xml
Content-Length: 318
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>MethodNotAllowed</Code>
  <Message>The specified method is not allowed against this resource.</Message>
  <RequestId>5CAC0CF8DE01700*****</RequestId>
  <HostId>versioning-get.oss-cn-hangzhou.aliyuncs.com</HostId>
  <Method>GET</Method>
  <ResourceType>DeleteMarker</ResourceType>
</Error>

```

1.2.6.1.3. CopyObject

Copies objects within a bucket or between different buckets in the same region.

To perform the copy operation on an existing object in Object Storage Service (OSS), you can send a PUT request to OSS, and add x-oss-copy-source to the PUT request to specify the source object that you want to copy. OSS determines that this is a copy operation and performs the copy operation on the server side. If the copy operation is successful, OSS returns the information of the destination object to you. You can call this operation to copy objects up to 1 GB in size. To copy an object larger than 1 GB in size, you must use UploadPartCopy. For more information, see [UploadPartCopy](#).

Usage notes

- If the source object URL is the same as the destination object URL during the CopyObject operation, OSS replaces the metadata of the source object regardless of the value of x-oss-metadata-directive.
- Request headers for the CopyObject operation start with x-oss- and must be added to a signature string.
- If the x-oss-server-side-encryption header is specified in the CopyObject request and the value of this header is valid, the destination object is encrypted on the server after the CopyObject operation is performed regardless of whether the source object is encrypted on the server. In addition, the response to a CopyObject request contains the x-oss-server-side-encryption header. The value of this header is set to the algorithm used to encrypt the destination object. When the destination object is downloaded, the response also contains the x-oss-server-side-encryption header, and the value of this header is set to the algorithm used to encrypt the destination object. If the x-oss-server-side-encryption header is not specified in the CopyObject request, the destination object is not encrypted on the server regardless of whether the source object is encrypted on the server.
- Objects created by using the AppendObject operation cannot be copied.

Versioning

By default, if you do not specify a version ID in the x-oss-copy-source header in the CopyObject request, the current version of the object is copied. You can add a version ID to x-oss-copy-source to copy a specified version of the object. If the specified version of the source object is a delete marker, OSS returns 404 Not Found, which indicates that the object does not exist.

To recover a previous version of an object, you can copy the previous version of the object to the bucket in which the object is stored. Then, the previous version of the object is stored as the current version.

If versioning is enabled for the destination bucket, OSS generates a unique version ID for the object that is copied to the destination bucket. The version ID is returned as the value of the x-oss-version-id header in the response. If versioning is disabled or suspended for the destination bucket, OSS generates a version whose ID is null for the object that is copied to the destination bucket, and overwrites the existing version whose ID is null.

Request syntax

```

PUT /DestObjectName HTTP/1.1
Host: DestBucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
x-oss-copy-source: /SourceBucketName/SourceObjectName

```

Request headers

Header	Type	Required	Example	Description
x-oss-copy-source	String	Yes	/oss-example/oss.jpg	The path of the source object. This parameter is empty by default.
x-oss-copy-source-if-match	String	No	5B3C1A2E053D763E1B002CC607C5***	If the ETag value of the source object is the same as the ETag value specified in the request, OSS copies the object and returns 200 OK. Otherwise, OSS returns 412 Precondition Failed, which indicates that the preprocessing has failed. This parameter is empty by default.
x-oss-copy-source-if-none-match	String	No	5B3C1A2E053D763E1B002CC607C5***	If the ETag value of the source object is different from the ETag value specified in the request, OSS copies the object and returns 200 OK. Otherwise, OSS returns 304 Not Modified, which indicates that the preprocessing has failed. This parameter is empty by default.

Header	Type	Required	Example	Description
x-oss-copy-source-if-unmodified-since	String	No	2019-04-09T07:01:56.000Z	If the specified time is equal to or later than the actual modified time of the object, OSS copies the object and returns 200 OK. Otherwise, OSS returns 412 Precondition Failed, which indicates that the preprocessing has failed. This parameter is empty by default.
x-oss-copy-source-if-modified-since	String	No	2019-04-09T07:01:56.000Z	If the source object is modified after the specified time, OSS copies the object. Otherwise, OSS returns 304 Not Modified, which indicates that the preprocessing has failed. This parameter is empty by default.
x-oss-metadata-directive	String	No	COPY	The method used to configure the metadata of the destination object. Default value: COPY. Valid values: COPY: The metadata of the source object is copied to the destination object. The x-oss-server-side-encryption attribute of the source object is not copied to the destination object. The x-oss-server-side-encryption parameter in the CopyObject request specifies the method used to encrypt the destination object on the server. REPLACE: The metadata specified in the request instead of the metadata of the source object is used as the metadata of the destination object.
x-oss-server-side-encryption	String	No	AES256	The algorithm that OSS uses to encrypt an object when you create the object. Valid values: <i>AES256</i> and <i>KMS</i> . - If the x-oss-server-side-encryption header is not specified in the CopyObject request, the destination object is not encrypted on the server regardless of whether the source object is encrypted on the server. - If the x-oss-server-side-encryption header is specified in the CopyObject request, the destination object is encrypted on the server after the CopyObject operation is performed regardless of whether the source object is encrypted on the server. In addition, the response to a CopyObject request contains the x-oss-server-side-encryption header. The value of this header is set to the algorithm used to encrypt the destination object. When the destination object is downloaded, the x-oss-server-side-encryption header is included in the response and the value of this header is set to the algorithm used to encrypt the object.
x-oss-server-side-encryption-key-id	String	No	9468da86-3509-4f8d-a61e-6eab1eac****	The ID of the customer master key (CMK) managed by Key Management Service (KMS). This parameter is valid only when x-oss-server-side-encryption is set to KMS.
x-oss-object-acl	String	No	private	The access control list (ACL) of the destination object when the object is created. Valid values: <i>public-read</i> , <i>private</i> , <i>public-read-write</i> , and <i>default</i> .
x-oss-tagging	String	No	a:1	The object tag. You can set multiple tags for the object. Example: TagA=A&TagB=B.  Note The tag key and value must be URL-encoded. If a key-value pair does not contain an equal sign (=), the tag value is considered an empty string.
x-oss-tagging-directive	String	No	Copy	The method used to configure tags for the destination object. Default value: Copy. Valid values: Copy: The tags of the source object are copied to the destination object. Replace: The tags specified in the request instead of the tags of the source object are copied to the destination object.

The common request headers involved in the CopyObject operation include the Host and Date headers. For more information about other common request headers involved in the CopyObject operation, see [Common request headers](#).

Response headers

The common response headers involved in the CopyObject operation include the Content-Length and Connection headers. For more information about other common response headers involved in the CopyObject operation, see [Common response headers](#).

Response elements

Element	Type	Example	Description

Element	Type	Example	Description
CopyObjectResult	Container	N/A	The result of the CopyObject operation. This parameter is empty by default.
ETag	String	5B3C1A2E053D763E1B002CC607C5****	The ETag value of the destination object. Parent nodes: CopyObjectResult
LastModified	String	Fri, 24 Feb 2012 07:18:48 GMT	The time when the destination object was last modified. Parent nodes: CopyObjectResult

Examples

- Copy an object in an unversioned bucket

Sample requests

```
PUT /copy_oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 07:18:48 GMT
x-oss-copy-source: /oss-example/oss.jpg
Authorization: OSS q***:g***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***1
Content-Type: application/xml
Content-Length: 193
Connection: keep-alive
Date: Fri, 24 Feb 2012 07:18:48 GMT
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <LastModified>Fri, 24 Feb 2012 07:18:48 GMT</LastModified>
  <ETag>"5***E"</ETag>
</CopyObjectResult>
```

- Copy an object in a versioned bucket without specifying a version ID.

Sample requests

```
PUT /dest-object-example HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 03:45:32 GMT
Authorization: OSS qeyxjc9arppwa0t:Qpw0jq7U7j04NVpPqdfcVkJI****
x-oss-copy-source: /versioning-copy-source/source-object
```

Sample responses

`x-oss-copy-source-version-id` in the sample response indicates the version ID of the source object. In this example, `x-oss-copy-source-version-id` indicates the current version of the source object. `x-oss-version-id` indicates the version ID of the destination object.

```
HTTP/1.1 200 OK
x-oss-copy-source-version-id: CAEQNRiBgIC28uaA0BYiIDY5OGIwNmNlNjYyMTRjNTc4N2M2OGNiMjZkZTQ2****
x-oss-version-id: CAEQNRiBgIDG8uaA0BYiIGZhZDRkZTk5Zjg3YzRhNzdiMWEwZGVhNDM1NTFh****
x-oss-request-id: 5CAC155CB7AEADE01700****
Content-Type: application/xml
Content-Length: 184
Connection: keep-alive
Date: Tue, 09 Apr 2019 03:45:32 GMT
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <ETag>"C81E728D9D4C2F636F067F89CC14****"</ETag>
  <LastModified>2019-04-09T03:45:32.000Z</LastModified>
</CopyObjectResult>
```

- Copy an object in a versioned bucket by specifying a version ID.

Sample requests

```
PUT /dest-object-example HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 03:45:32 GMT
Authorization: OSS qeyxjc9arppwa0t:5qG4DLahjxDtpLlf2e8fBfX****
x-oss-copy-source: /versioning-copy-source/source-object?versionId=CAEQNRiBgICv8uaA0BYiIDl1ZDc3MTc1NjE5MjRkMDI4ZGU4MTZkYjY1ZDgy****
```

Sample responses

`x-oss-copy-source-version-id` in the sample response indicates the version ID of the source object. In this example, `x-oss-copy-source-version-id` indicates the version ID specified by `x-oss-copy-source` in the request. `x-oss-version-id` indicates the version ID of the destination object.

```

HTTP/1.1 200 OK
x-oss-copy-source-version-id: CAEQNRiBgICv8uaA0BYiIDliZDc3MTc1NjE5MjRkMDI4ZGU4MTZkYjY1ZDgy****
x-oss-version-id: CAEQNxiBgMDP8uaA0BYiIDiYNGNhZDQ1M2M3NzRkZThiNzE0N2I3ZDkxOWY4****
x-oss-request-id: 5CAC155CB7AEADE01700****
Content-Type: application/xml
Content-Length: 184
Connection: keep-alive
Date: Tue, 09 Apr 2019 03:45:32 GMT
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <ETag>"C4CA4238A0B923820DCC509A6F75****"</ETag>
  <LastModified>2019-04-09T03:45:32.000Z</LastModified>
</CopyObjectResult>

```

1.2.6.1.4. AppendObject

Uploads an object by appending the object to an existing object. Objects created by using the AppendObject operation are appendable objects. Objects uploaded by using the PutObject operation are normal objects.

Usage notes

- You can perform the AppendObject operation only on an appendable object. For example, if a normal object that shares the same name already exists and AppendObject is called, Object Storage Service (OSS) returns HTTP status code 409 and error code ObjectNotAppendable.
- If you perform the PutObject operation on an existing appendable object, the appendable object is overwritten by a new normal object.
- If you perform the HeadObject operation on an object, OSS returns x-oss-object-type to indicate the type of the object. If you perform the HeadObject operation on an appendable object, the value of x-oss-object-type is Appendable. If you perform the HeadObject operation on a normal object, OSS also returns x-oss-next-append-position and x-oss-hash-crc64ecma.
- In the XML response to a GetBucket (ListObjects) request, the type of an appendable object is set to Appendable.
- You can neither use CopyObject to copy an appendable object nor change the server-side encryption attribute of this object. You can use CopyObject to modify the user metadata of this object.
- The append and position parameters belong to CanonicalizedResource that must be included in signatures.

Versioning

When you perform the AppendObject operation on appendable objects in a bucket for which versioning is enabled or suspended, take note of the following items:

- You can perform the AppendObject operation only on an object whose current version is an appendable object. When you perform this operation, OSS does not generate a previous version for the appendable object.

When you perform the PutObject or DeleteObject operation on an object whose current version is an appendable object, OSS stores the appendable object as a previous version. However, the appendable object cannot be further appended.

- Appendable objects cannot be copied.
- The AppendObject operation cannot be performed on non-appendable objects such as normal objects and delete markers.

Request syntax

```

POST /ObjectName?append&position=Position HTTP/1.1
Content-Length: ContentLength
Content-Type: ContentType
Host: BucketName.oss.example.com
Date: GMT Date
Authorization: SignatureValue

```

Request headers

Header	Type	Required	Example	Description
append	String	Yes	N/A	Specifies that the request is an AppendObject request. Each time an AppendObject operation is performed on an object, the time when the object is last modified is updated.
position	String	Yes	0	The position from which the AppendObject operation starts. Each time an AppendObject operation succeeds, the x-oss-next-append-position header is included in the response to specify the position from which the next AppendObject operation starts. The value of position in the first AppendObject operation performed on an object must be 0. The value of position in subsequent AppendObject operations performed on the object is the current length of the object. For example, if the value of position specified in the first AppendObject request is 0 and the value of Content-Length is 65536, the value of position in the second AppendObject request must be set to 65536. - If the value of position in an AppendObject request is 0 and no object that has the same name as the appended object exists, you can set headers such as x-oss-server-side-encryption in the AppendObject request in the same way as you do in a PutObject request. If you add a valid x-oss-server-side-encryption header to an AppendObject request in which the value of position is 0, the x-oss-server-side-encryption header is included in the response to the request. To modify the metadata specified by the headers, you can initiate a CopyObject request. - If you initiate an AppendObject request in which the value of position is valid to append an object that is 0 KB in size to an existing appendable object, the status of the appendable object is not changed.

Header	Type	Required	Example	Description
Cache-control	String	No	no-cache	The web page caching behavior for the object. For more information, see RFC 2616 . This parameter is empty by default.
Content-Disposition	String	No	attachment;filename=oss_download.jpg	The name of the object when the object is downloaded. For more information, see RFC 2616 . This parameter is empty by default.
Content-Encoding	String	No	utf-8	The encoding format of the object. For more information, see RFC 2616 . This parameter is empty by default.
Content-MD5	String	No	ohhnqLBJFikKPSBO1eNaUA==	The value of Content-MD5 is a string calculated by using the MD5 algorithm. This header is used to check whether the content of the received message is the same as that of the sent message. To obtain the value of the Content-MD5 header, calculate a 128-bit number based on the message content except for the header, and then encode the number in Base64. This parameter is empty by default. Limits: none
Expires	GMT	No	Wed, 08 Jul 2015 16:57:01 GMT	The time when the cache expires. For more information, see RFC 2616 . This parameter is empty by default.
x-oss-server-side-encryption	String	No	AES256	The method used to encrypt objects on the OSS server. Valid values: AES256: OSS-managed keys (SSE-OSS) are used for encryption and decryption. KMS: Customer master keys (CMKs) managed by Key Management Service (KMS) are used for encryption and decryption. SM4: The SM4 block cipher algorithm is used for encryption and decryption.
x-oss-object-acl	String	No	private	The access control list (ACL) of the object. Valid values: public-read: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. private: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. public-read-write: All users, including anonymous users, can perform read and write operations on objects in the bucket. Fees incurred by these operations are paid by the bucket owner. Exercise caution when you set the object ACL to this value.
x-oss-meta-*	String	No	x-oss-meta-location	You can add parameters prefixed with x-oss-meta- when you create an appendable object by performing the AppendObject operation. These parameters cannot be included in the requests when you append objects to an existing appendable object. Parameters prefixed with x-oss-meta-* are considered the user metadata of the object. You can configure multiple parameters prefixed with x-oss-meta- for an object. However, the total size of the metadata cannot exceed 8 KB. The names of parameters prefixed with x-oss-meta- can contain hyphens (-), digits, and letters. Uppercase letters are converted to lowercase letters. Other characters such as underscores (_) are not supported.

For more information about the common request headers included in an AppendObject request, see [Common request headers](#).

Response headers

Header	Type	Example	Description
x-oss-next-append-position	64-bit integer	1717	The position that must be provided in the next request, which is the current length of the object. This header is returned when a success message is returned for an AppendObject request, or when HTTP status code 409 is returned because the position and the object length do not match.
x-oss-hash-crc64ecma	64-bit integer	3231342946509354535	The 64-bit CRC value of the object. This value is calculated based on the ECMA-182 standard.

For more information about the common response headers included in the response to an AppendObject request, see [Common response headers](#).

Methods used to calculate the 64-bit CRC value

The CRC value of an appendable object is calculated based on the [ECMA-182](#) standard. You can calculate the 64-bit CRC value of an object by using the following methods:

- Use the Boost CRC module

```
typedef boost::crc_optimal<64, 0x42F0E1EBA9EA3693ULL, 0xffffffffffffffffULL, 0xffffffffffffffffULL, true, true> boost_ecma;
uint64_t do_boost_crc(const char* buffer, int length)
{
    boost_ecma crc;
    crc.process_bytes(buffer, length);
    return crc.checksum();
}
```

- Use the `crcmod` function in Python

```
do_crc64 = crcmod.mkCrcFun(0x142F0E1EBA9EA3693L, initCrc=0L, xorOut=0xffffffffffffffffL, rev=True)
print do_crc64("123456789")
```

Examples

- Append an object to an existing object in a bucket for which versioning is disabled

Sample requests

```
POST /oss.jpg?append&position=0 HTTP/1.1
Host: ***.regionid.example.com
Cache-control: no-cache
Expires: Wed, 08 Jul 2015 16:57:01 GMT
Content-Encoding: utf-8
Content-Disposition: attachment;filename=oss_download.jpg
Date: Wed, 08 Jul 2015 06:57:01 GMT
Content-Type: image/jpeg
Content-Length: 1717
Authorization: OSS qn6qrrqxo2oawuk53otf****:kZoYNv66bsmc10+dcGKw5x2P****
[1717 bytes of object data]
```

Sample responses

```
HTTP/1.1 200 OK
Date: Wed, 08 Jul 2015 06:57:01 GMT
ETag: "0F7230CAA4BE94CCBDC99C550000****"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-hash-crc64ecma: 14741617095266562575
x-oss-next-append-position: 1717
x-oss-request-id: 559CC9BDC755F95A6448****
```

- Append an object to an existing object in a bucket for which versioning is enabled or suspended

Sample requests

If you perform the `AppendObject` operation on an object in a bucket for which versioning is enabled or suspended, the `x-oss-version-id` header is included in the response and its value must be null.

```
POST /example?append&position=0 HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 03:59:33 GMT
Content-Length: 3
Content-Type: application/octet-stream
Authorization: OSS bwo4j5l8d3j****:MCY5nnfgfJU/f3Xe0odqBtG5****
```

Sample responses

```
HTTP/1.1 200 OK
Date: Tue, 09 Apr 2019 03:59:33 GMT
ETag: "2776271A4A09D82CA518AC5C0000****"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-version-id: null
x-oss-hash-crc64ecma: 3231342946509354535
x-oss-next-append-position: 3
x-oss-request-id: 5CAC18A5B7AEADE01700****
```

1.2.6.1.5. DeleteObject

Deletes an object.

Limits

- You must have write permissions on the object that you want to delete.
- HTTP status code 204 is returned when the `DeleteObject` operation succeeds, regardless of whether the object exists.
- If the object you want to delete is a symbolic link, the `DeleteObject` operation deletes only the symbolic link but not the content to which the symbolic link points.

Versioning

When you call `DeleteObject` to delete an object from a versioned bucket, you must determine whether to specify a version ID in the request.

- Delete an object without specifying a version ID (temporary deletion)

By default, if you do not specify the version ID of the object that you want to delete in the request, OSS does not delete the current version of the object but adds a delete marker to the object as the new current version. In this case, if you perform the `GetObject` operation on the object without specifying a version ID in the request, OSS identifies that the current version of the object is a delete marker and returns `404 Not Found`. In addition, the `x-oss-delete-marker=true` header and the `x-oss-version-id` header that indicates the version ID of the created delete marker are included in the response.

The value of the `x-oss-delete-marker` header in the response is true, which indicates that the `x-oss-version-id` value is the version ID of a delete marker.

- Delete an object by specifying a version ID (permanent deletion)

If you specify the version ID of the object that you want to delete in the request, OSS permanently deletes the version specified by the `versionId` field in the `params` parameter. To delete a version whose ID is null, add `params['versionId'] = "null"` to `params` in the request. OSS identifies the string "null" as the ID of the version to delete and deletes the version whose ID is null.

Request syntax

```
DELETE /ObjectName HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

All headers in a `DeleteObject` request are common request headers. For more information about common request headers, see [Common request headers](#).

Response headers

Header	Type	Example	Description
<code>x-oss-delete-marker</code>	Boolean	true	Indicates whether the object is a delete marker. • If you do not specify the version ID of the object that you want to delete in the <code>DeleteObject</code> request, OSS creates a delete marker as the current version of the object and includes this header with a value of true in the response. • If you specify the version ID of the object you want to delete in the <code>DeleteObject</code> request and the specified version is a delete marker, OSS includes this header in the response and the value of this header is true. Valid value: <i>true</i>
<code>x-oss-version-id</code>	String	CAEQMxiBgIDh3ZCB0BYiIGe4YjlyMjE xZDhhYjQxNzZINGUyZTI4ZjZjZDcz*	The version ID of the deleted object. • If you do not specify the version ID of the object that you want to delete in the <code>DeleteObject</code> request, OSS creates a delete marker as the current version of the object and includes this header in the response to indicate the version ID of the created delete marker. • If you specify the version ID of the object you want to delete in the <code>DeleteObject</code> request, OSS includes this header in the response to indicate the ID of the deleted object.

The response to this request contains common response headers. For more information about common response headers, see [Common response headers](#).

Examples

- Delete an object from an unversioned bucket

Sample requests

```
DELETE /AK.txt HTTP/1.1
Host: *.regionid.example.com
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: text/html
Connection: keep-alive
date: Wed, 02 Jan 2019 13:28:38 GMT
authorization: OSS qn6qrrqxo2oawuk53otfjbyc:zUglwRPGkbByZxml+y4eyu+NIUs=zV0****
Content-Length: 0
```

Sample responses

```
HTTP/1.1 204 No Content
Server: AliyunOSS
Date: Wed, 02 Jan 2019 13:28:38 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5C2CBC8653718B5511EF4535
x-oss-server-time: 134
```

- Delete an object from a versioned bucket without specifying a version ID

In this case, OSS adds a delete marker to the object and includes the `x-oss-delete-marker=true` header in the response.

Sample requests

```
DELETE /example HTTP/1.1
Host: *.regionid.example.com
Date: Tue, 09 Apr 2019 04:08:23 GMT
Authorization: OSS twnetzwjkr9eq6:z73SSKA6t2tNTP4GuPjPiyV/****
```

Sample responses

```
HTTP/1.1 204 NoContent
x-oss-delete-marker: true
x-oss-version-id: CAEQMxiBgIDh3ZCB0BYiIGE4YjIyMjExZDhhYjQxNzZiNGUyZTI4ZjljZDcz****
x-oss-request-id: 5CAC1AB7B7AEADE01700****
Date: Tue, 09 Apr 2019 04:08:23 GMT
Connection: keep-alive
Server: AliyunOSS
```

- Delete a version of the object from a versioned bucket by specifying the version ID

In this case, the specified version of the object is permanently deleted.

Sample requests

```
DELETE /example?versionId=CAEQOBiBgIDNlJeB0BYiIDAwYjJlNDQ4YjJkMzQxMmY5NTM5N2UzZWNiZTQ2**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 04:11:54 GMT
Authorization: OSS gb3m2qiwirupd6v:UjOXBmIbJD3qXL+DP1EDNyCI****
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-version-id: CAEQOBiBgIDNlJeB0BYiIDAwYjJlNDQ4YjJkMzQxMmY5NTM5N2UzZWNiZTQ2****
x-oss-request-id: 5CAC1B8AB7AEADE01700****
Date: Tue, 09 Apr 2019 04:11:54 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- Delete a delete marker from a versioned bucket by specifying the version ID of the delete marker.

In the following example, the specified version is a delete marker. The `x-oss-delete-marker=true` header is included in the response.

Sample requests

```
DELETE /example?versionId=CAEQOBiBgIDNlJeB0BYiIDAwYjJlNDQ4YjJkMzQxMmY5NTM5N2UzZWNiZTQ2**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 04:16:25 GMT
Authorization: OSS jh475i54ffozhoy:4tX6Z+fnhtINhp0g+sRiLEqb****
```

Sample responses

```
HTTP/1.1 204 No Content
x-oss-delete-marker: true
x-oss-version-id: CAEQNhiBgIDFtp.B0BYiIDk4NzgwMmU4NDMyOTQyM2NiMDQxOTcxYWNhMjc1****
x-oss-request-id: 5CAC1C99B7AEADE01700****
Date: Tue, 09 Apr 2019 04:16:25 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.6.1.6. DeleteMultipleObjects

Deletes multiple objects in the same bucket by sending a single HTTP request.

A maximum of 1,000 objects can be deleted when you call the DeleteMultipleObjects operation. This operation supports the following return modes:

- Verbose mode: The message body of the response returned by Object Storage Service (OSS) contains all deleted objects.
- Quiet mode: The message body of the response returned by OSS contains only objects that fail to be deleted. If all objects are deleted, no message body is returned.

Usage notes

- You must specify the Content-Length and Content-MD5 headers in the DeleteMultipleObjects request. OSS verifies that the received message body is correct based on the values of the preceding headers before the delete operation is performed.
- To obtain the Content-MD5 value, encrypt the message body of the DeleteMultipleObjects request by using the MD5 algorithm to obtain a 128-bit byte array, and then encode the byte array in Base64.
- If DeleteMultipleObjects is used to delete an object that does not exist, the operation is considered to be successful.
- The message body of the DeleteMultipleObjects request can be up to 2 MB in size. If the size of the message body exceeds 2 MB, OSS returns the MalformedXML error code.
- A maximum of 1,000 objects can be deleted when you call the DeleteMultipleObjects operation. If more than 1,000 objects are requested, OSS returns the MalformedXML error code.
- If the Content-MD5 header is specified in the request, OSS calculates the MD5 hash of the received content and checks whether the calculated MD5 hash is the same as the Content-MD5 value specified in the request. If the two MD5 hashes are different, OSS returns InvalidDigest.

Versioning

You can call the DeleteMultipleObjects operation to delete multiple objects from a versioned bucket. If you do not specify the version IDs of the objects that you want to delete in your request, delete markers are added to the objects. If you specify the version IDs of the objects that you want to delete in your request, objects with the specified version IDs are permanently deleted.

Request syntax

```

POST /?delete HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Content-Length: ContentLength
Content-MD5: MD5Value
Authorization: SignatureValue
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>true</Quiet>
  <Object>
    <Key>key</Key>
  </Object>
  ...
</Delete>

```

Request headers

When you call the DeleteMultipleObjects operation, you can use the Encoding-type parameter to encode the object names in the response.

Header	Type	Required	Example	Description
Encoding-type	String	No	url	The encoding type of the object names in the response. The object names are UTF-8 encoded. If an object name includes control characters that are not supported by the XML 1.0 standard, you can specify this header to encode the object name in the response. Valid value: <i>url</i>
Content-Length	String	Yes	151	The length of the HTTP message body. OSS verifies the message body of the request and deletes the specified objects only when the length of the message body is the same as the value of this header.
Content-MD5	String	Yes	IVaXz0QX0UTCXr aY9Z61mQ==	The value of the Content-MD5 header is a string calculated by using the MD5 algorithm. The header is used to check whether the content of the received message is the same as that of the sent message. After the Content-MD5 header is specified in the request, OSS calculates the MD5 hash based on the message body and checks whether the calculated MD5 hash is the same as the Content-MD5 value specified in the request. Note To obtain the value of the Content-MD5 header, encrypt the message body of the DeleteMultipleObjects request by using the MD5 algorithm to obtain a 128-bit byte array, and then encode the byte array in Base64.

Request elements

Element	Type	Required	Example	Description
Delete	Container	Yes	N/A	The container that stores the DeleteMultipleObjects request content.
Object	Container	Yes	N/A	The container that stores the information about an object.
Key	String	Yes	test.jpg	The name of the object that you want to delete.
Quiet	Enumerated string	Yes	true	Specifies whether to enable the Quiet return mode. The DeleteMultipleObjects operation provides the following return modes: - Quiet mode: The message body of the response returned by OSS contains only objects that fail to be deleted. If all objects are deleted, no message body is returned. - Verbose mode: The message body of the response returned by OSS contains all deleted objects. By default, this mode is used. Valid values: <i>true</i> and <i>false</i> Default value: <i>false</i>
VersionId	String	No	CAEQMhiBgIDXiaa B0BYiIGQzYmRkZ GUxMTM1ZDRjOT ZhNjk4YjRjMTAyZ jhl*****	The version ID of the object that you want to delete. Parent nodes: Object

Response headers

All headers in the response to a PutObject request are common response headers. For more information, see [Common response headers](#).

Response elements

Element	Type	Description
Deleted	Container	The container that stores the deleted objects. Child nodes: Key Parent nodes: DeleteResult

Element	Type	Description
DeleteResult	Container	The container that stores the response to the DeleteMultipleObjects request. Child nodes: Deleted Parent nodes: none
Key	String	The name of the deleted object. Parent nodes: Deleted
EncodingType	String	The encoding type of deleted object names in the response. If the Encoding-type header is specified in the request, object names are encoded in the response. Parent nodes: container
DeleteMarker	Boolean	Indicates whether the specified version is a delete marker. Valid values: true and false.  Note This element is returned only if a delete marker is created or permanently deleted. In this case, the value of this element is true.
DeleteMarkerVersionId	String	The version ID of the delete marker. Parent nodes: Deleted

Examples

- Sample requests in which the Verbose mode is enabled

```
POST /?delete HTTP/1.1
Host: *.regionid.example.com
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length:151
Content-MD5: ohhnqLBJFiKkPSB01eNaUA==
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:z3gBfnFAxBcBDgx27Y/jEfB****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
  </Object>
  <Object>
    <Key>test.jpg</Key>
  </Object>
  <Object>
    <Key>demo.jpg</Key>
  </Object>
</Delete>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 78320852-7eee-b697-75e1-b6db0f4849e7
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length: 244
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Deleted>
    <Key>multipart.data</Key>
  </Deleted>
  <Deleted>
    <Key>test.jpg</Key>
  </Deleted>
  <Deleted>
    <Key>demo.jpg</Key>
  </Deleted>
</DeleteResult>
```

- Sample requests in which the Quiet mode is enabled

```
POST /?delete HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length:151
Content-MD5: ohhngLBJFiKkPSB01eNaUA==
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:WuV0Jks8RyGSNqrBoa64kEE****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>true</Quiet>
  <Object>
    <Key>multipart.data</Key>
  </Object>
  <Object>
    <Key>test.jpg</Key>
  </Object>
  <Object>
    <Key>demo.jpg</Key>
  </Object>
</Delete>
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- Sample requests in which the VersionId element is not specified

```
POST /?delete HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 04:20:03 GMT
Content-MD5: xSLOYWaPC86RpWwXNiFeXg==
Authorization: OSS xq39jyxYrddzvvh:/G5kxMIwlilMQjPp2HkJEp5q****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
  </Object>
  <Object>
    <Key>test.jpg</Key>
  </Object>
</Delete>
```

Sample responses

In the following example, the version IDs of the two objects that you want to delete (multipart.dat and test.jpg) are not specified. Therefore, OSS adds delete markers to the two objects, and returns <DeleteMarker>true</DeleteMarker> and <DeleteMarkerVersionId>XXXXXX</DeleteMarkerVersionId>.

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC1D73B7AEADE01700****
Date: Tue, 09 Apr 2019 04:20:03 GMT
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult>
  <Deleted>
    <Key>multipart.data</Key>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>CAEQMhiBgIDXiab0BYiIGQzYmRkZGUxMTM1ZDRjOTZhNjk4YjRjMTAyZjhl****</DeleteMarkerVersionId>
  </Deleted>
  <Deleted>
    <Key>test.jpg</Key>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>CAEQMhiBgIDB3aWB0BYiIGUzYTA3YzliMzVmNzRkZGM5NjllYTVMjYyYWEy****</DeleteMarkerVersionId>
  </Deleted>
</DeleteResult>
```

- Sample requests in which the VersionId element is specified

 **Note** You must specify both the Key and VersionId elements.

```

POST /?delete HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 06:09:34 GMT
Content-Length: 151
Content-MD5: 2Tpk+dL/tGyuSA+YCEuSVg==
Authorization: OSS qac50zy2vbbq4z:/pingCxyqfxc0+50Bfi2SX9c****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
    <VersionId>CAEQNRiBgIDyz.6C0BYiIGQ2NWEwNmVhNTA3ZTQ3MzM5ODliYjM1ZTdYjA4****</VersionId>
  </Object>
</Delete>

```

Sample responses

In this example, the key and version ID of the deleted object are returned.

```

HTTP/1.1 200 OK
x-oss-request-id: 5CAC371EB7AEADE01700****
Date: Tue, 09 Apr 2019 06:09:34 GMT
Content-Length: 244
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Deleted>
    <Key>multipart.data</Key>
    <VersionId>CAEQNRiBgIDyz.6C0BYiIGQ2NWEwNmVhNTA3ZTQ3MzM5ODliYjM1ZTdYjA4****</VersionId>
  </Deleted>
</DeleteResult>

```

- Sample requests in which the VersionId element is specified to delete delete markers

```

POST /?delete HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 06:14:50 GMT
Content-Length: 178
Content-MD5: dX9IFePFgYhmINvAhG30Bg==
Authorization: OSS 9dfwrokza5gflp7:iyl6IU6TcfFXcu5p0ds5dUdo****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
    <VersionId>CAEQNRiBgICEoPiC0BYiIGMxZWJmMzYjE0OTQ0ZmZhYjgzNzkzYjc2NjZk****</VersionId>
  </Object>
</Delete>

```

Sample responses

- In this example, the Key element in the response indicates the name of the deleted object and the VersionId element in the response indicates the deleted version of the object.
- DeleteMarker indicates that the deleted version is a delete marker. DeleteMarkerVersionId indicates the ID of the deleted version. In this case, the values of VersionId and DeleteMarkerVersionId are the same, and DeleteMarker and DeleteMarkerVersionId are returned together.

```

HTTP/1.1 200 OK
x-oss-request-id: 5CAC385AB7AEADE01700****
Date: Tue, 09 Apr 2019 06:14:50 GMT
Content-Length: 364
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Deleted>
    <Key>demo.jpg</Key>
    <VersionId>CAEQNRiBgICEoPiC0BYiIGMxZWJmMzYjE0OTQ0ZmZhYjgzNzkzYjc2NjZk****</VersionId>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>111111</DeleteMarkerVersionId>
  </Deleted>
</DeleteResult>

```

1.2.6.1.7. HeadObject

Returns only the metadata of a specified object without returning the object content.

Usage notes

- After a HeadObject request is sent, no message body is returned regardless of whether Object Storage Service (OSS) returns the 200 OK message or an error message.
- If you specify user metadata prefixed with x-oss-meta- when you send a PutObject request, such as x-oss-meta-location, the user metadata is returned.
- If the object on which you want to perform the HeadObject operation does not exist, OSS returns 404 Not Found.
- If server-side encryption is configured for the object, OSS returns the x-oss-server-side-encryption header in the response. The value of the x-oss-server-side-

encryption header indicates the server-side encryption algorithm of the object.

Versioning

By default, you can call the HeadObject operation to query the metadata of the object of the current version. If the current version of the object is a delete marker, OSS returns 404 Not Found. If you specify a version ID in the request, OSS returns the metadata of the object of the specified version.

Request syntax

```
HEAD /ObjectName HTTP/1.1
Host: BucketName/regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

Header	Type	Required	Example	Description
If-Modified-Since	String	No	Fri, 24 Feb 2012 07:32:52 GMT	The information return condition. If the specified time is earlier than the last modified time of the object, OSS returns the metadata of the object and 200 OK. Otherwise, OSS returns 304 Not Modified. This parameter is empty by default.
If-Unmodified-Since	String	No	Fri, 24 Feb 2012 07:32:52 GMT	The information return condition. If the specified time is equal to or later than the last modified time of the object, OSS returns the metadata of the object and 200 OK. Otherwise, OSS returns 412 Precondition Failed. This parameter is empty by default.
If-Match	String	No	215697CF4417D144C25EB698F5*****	The information return condition. If the specified ETag value matches the ETag value of the object, OSS returns the metadata of the object and 200 OK. Otherwise, OSS returns 412 Precondition Failed. This parameter is empty by default.
If-None-Match	String	No	215697CF4417D144C25EB698F5*****	The information return condition. If the specified ETag value does not match the ETag value of the object, OSS returns the metadata of the object and 200 OK. Otherwise, OSS returns 304 Not Modified. This parameter is empty by default.

Response headers

If the requested object is a symbolic link, take note of the following headers included in the response:

- The Content-Length, ETag, and Content-Md5 headers are the metadata of the object to which the symbolic link points.
- The returned value of Last-Modified uses the last modified time of the symbolic link or the last modified time of the object to which the symbolic link points, whichever one is later.
- Other response headers are the metadata of the symbolic link.

Header	Type	Example	Description
x-oss-meta-*	String	x-oss-meta-test	The user metadata that is prefixed with x-oss-meta-. If you specify headers prefixed with x-oss-meta- in the PutObject request, the user metadata of the object is included in the response.
Custom headers that are not prefixed with x-oss-meta-	String	x-oss-persistent-headers	If you specify headers that are not prefixed with x-oss-meta in the PutObject request to configure the user metadata of the object, the headers are included in the response. Example: x-oss-persistent-headers: <i>key1:base64_encode(value1),key2:base64_encode(value2),....</i>
x-oss-server-side-encryption	String	KMS	This header is included in the response if the object is encrypted by using the server-side encryption algorithm based on entropy encoding. The value of this header is the algorithm used to encrypt the object on the server side.
x-oss-server-side-encryption-key-id	String	3dea0cae-5fd6-4bf8-8574-87c557*****	This header is included in the response if the object is encrypted by using KMS on the OSS server. The value of this header is the CMK ID used to encrypt the object.
x-oss-object-type	String	Normal	The type of the object. Valid values: - Normal: The object is uploaded by using PutObject. - Appendable: The object is uploaded by using AppendObject. - Multipart: The object is uploaded by using MultipartUpload.
x-oss-next-append-position	String	100	The position for the next append operation. If the type of the object is Appendable, this header is included in the response.
x-oss-hash-crc64ecma	String	15429415218160274758	The 64-bit CRC value of the object. This value is calculated based on the ECMA-182 standard. If an object is created before OSS supports CRC-64, this header may not be included in the response when you call HeadObject to query the metadata of the object.
Content-Md5	String	IVaXz0QX0UTCXraY9Z61mQ==	- This header is included in the response if the object type is Normal. The value of this header is calculated based on the following process: 1. Calculate the MD5 hash of the message content based on RFC 1864 to obtain a 128-bit digit. Do not include headers to calculate the MD5 hash. 2. Encode the value in Base64. - If the object type is Multipart or Appendable, this header is not included in the response.

Header	Type	Example	Description
Last-Modified	String	Fri, 24 Feb 2012 06:07:48 GMT	The time when the object was last modified. The time is in GMT specified in HTTP/1.1.
x-oss-tagging-count	String	10	The number of tags added to the object. This header is included in the response only when you have read permissions on tags.

Examples

- Query the metadata of an object in an unversioned bucket

Sample requests

```
HEAD /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanlJ5dLA092wpD****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A6448****
x-oss-object-type: Normal
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpeg
Connection: keep-alive
Server: AliyunOSS
```

- Query the metadata of an object of the specified version in a versioned bucket

Sample requests

```
HEAD /example?versionId=CAEQNRiBgICb8o6D0BYiIDNlNzk5NGE2M2Y3ZjRhZTViYTAXZGE0ZTEyMWYy****
Host: versioning-test.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 06:27:12 GMT
Authorization: OSS ryghu9rp3mqplya:RvyjGvKxaUhdF0ibyEwX5mQM****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-versionId: CAEQNRiBgICb8o6D0BYiIDNlNzk5NGE2M2Y3ZjRhZTViYTAXZGE0ZTEyMWYy****
x-oss-request-id: 5CAC3B40B7AEADE01700****
x-oss-object-type: Normal
Date: Fri, 7 Aug 2020 06:27:12 GMT
Last-Modified: Fri, 7 Aug 2020 06:27:12 GMT
ETag: "A082B659EF78733A5A042FA253B1****"
Content-Length: 481827
Content-Type: text/html
Connection: keep-alive
Server: AliyunOSS
```

- Query the metadata of an object of the current version in a versioned bucket

Sample requests

```
HEAD /example HTTP/1.1
Host: versioning-test.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 06:27:12 GMT
Authorization: OSS ryghu9rp3mqplya:RvyjGvKxaUhdF0ibyEwX5mQM****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-versionId: CAEQMxiBgMCZov2D0BYiIDY4MDllOTc2YmY5MjQxMzdiOGI3OTlhNTU0ODIx****
x-oss-request-id: 5CAC3B40B7AEADE01700****
x-oss-object-type: Normal
Date: Fri, 7 Aug 2020 06:27:12 GMT
Last-Modified: Fri, 7 Aug 2020 06:27:12 GMT
ETag: "3663F7B0B9D3153F884C821E7CF4****"
Content-Length: 485859
Content-Type: text/html
Connection: keep-alive
Server: AliyunOSS
```

- Query the metadata of an object that is encrypted by using SSE-OSS

Sample requests

```
HEAD /oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanlJ5dLA092wpD****
```

Sample responses


```

HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC75F95A6448****
x-oss-object-type: Normal
x-oss-server-side-encryption: AES256
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 7 Aug 2020 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpg
Connection: keep-alive
Server: AliyunOSS

```

- Query the metadata of an object that is encrypted by using SSE-KMS

Sample requests

```

HEAD /oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanlJ5dLA092wpD****

```

Sample responses

```

HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC75F95A64485981
x-oss-object-type: Normal
x-oss-server-side-encryption: KMS
x-oss-server-side-encryption-key-id: 9468da86-3509-4f8d-a61e-6eableac****
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 7 Aug 2020 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpg
Connection: keep-alive
Server: AliyunOSS

```

1.2.6.1.8. GetObjectMeta

Queries only the basic metadata but not the content of an object in a bucket. The basic metadata of an object includes the ETag value, size, and last modified time of the object.

Usage notes

- After a GetObjectMeta request is sent, no message body is contained in the response.
- You must include parameters related to the object metadata in the request when you call the GetObjectMeta operation. Otherwise, the request is used for GetObject.
- If the object on which you want to perform the GetObjectMeta operation does not exist, Object Storage Service (OSS) returns 404 Not Found.
- Compared with HeadObject, GetObjectMeta returns only some basic metadata of an object in a bucket. The basic metadata includes the ETag value, object size, and last modified time. The value of the Content-Length response header indicates the object size.

Versioning

By default, the GetObjectMeta operation obtains the metadata of an object of the current version. If the current version of the object is a delete marker, OSS returns 404 Not Found. If you specify a version ID in the request, OSS returns the metadata of the object of the specified version.

Request syntax

```

GET /ObjectName?objectMeta HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue

```

Request headers

A PutBucketLifecycle request contains only common request headers. For more information, see [Common request headers](#).

Response headers

Header	Type	Example	Description
Content-Length	String	344606	The size of the object.
ETag	String	215697CF4417D144C25EB698F59EB599	The entity tag (ETag) that is generated when an object is created. The ETag value is used to identify the content of the object. If an object is created by performing the PutObject operation, the ETag value of the object is the MD5 hash of the object content. If an object is not created by performing the PutObject operation, the ETag value is the UUID of the object content. You can use the ETag value of an object to check whether the object content is modified. However, we recommend that you do not use the ETag value of an object as the MD5 hash of the object to check data integrity. This parameter is empty by default.

Header	Type	Example	Description
Last-Modified	String	Fri, 24 Feb 2012 06:07:48 GMT	The time when the object was last modified. The time is in GMT specified in HTTP/1.1.

Examples

- Query the metadata of an object in an unversioned bucket

Sample requests

```
HEAD /oss.jpg?objectMeta HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLai4XZ+WwIfNm0Fmgb****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A6448****
Date: Wed, 29 Apr 2015 05:21:12 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5*****"
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 344606
Connection: keep-alive
Server: AliyunOSS
```

- Query the metadata of an object in a versioned bucket

Sample requests

```
GET /example?objectMeta&versionId=CAEQNRiBgIDMh4mD0BYiIDUzNDA4OGNmZjBjYTQ0YmI4Y2I4ZmVlYzJlNGVh**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 06:24:00 GMT
Authorization: OSS 5n4nrhyqrcs6ngn:i/M/c36KzrOEA/bBSHLlIIAt****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-version-id: CAEQNRiBgIDMh4mD0BYiIDUzNDA4OGNmZjBjYTQ0YmI4Y2I4ZmVlYzJlNGVh****
x-oss-request-id: 5CAC3A80B7AEADE0170005F6
Date: Tue, 09 Apr 2019 06:24:00 GMT
ETag: "1CF5A685959CA2ED8DE6E5F8ACC2*****"
Last-Modified: Tue, 09 Apr 2019 06:24:00 GMT
Content-Length: 119914
Connection: keep-alive
Server: AliyunOSS
```

1.2.6.1.9. PostObject

You can call this operation to upload an object to a specific bucket by using HTML forms.

Usage notes

When you call the PostObject operation, take note of the following items:

- To perform the PostObject operation on a bucket, you must have write permissions on the bucket. If the ACL of the bucket is public read/write, you do not need to upload the signature information. If the ACL of the bucket is not public read/write, signature verification is required for this operation.
- Unlike PutObject, PostObject uses an AccessKey secret to calculate the signature for a policy. The calculated signature string is used as the value of the Signature form field. OSS checks this value to verify the validity of the signature.
- The URL of the submitted form can be the domain name of the bucket. You do not need to specify the object in the URL. In other words, the request line is in the format of `POST / HTTP/1.1` instead of `POST /ObjectName HTTP/1.1`.
- OSS does not check the signature included in headers or URLs in PostObject requests.

Request structure

```

POST / HTTP/1.1
Host: BucketName.regionid.example.com
User-Agent: browser_data
Content-Length: ContentLength
Content-Type: multipart/form-data; boundary=9431149156168
--9431149156168
Content-Disposition: form-data; name="key"
key
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"
success_redirect
--9431149156168
Content-Disposition: form-data; name="Content-Disposition"
attachment;filename=oss_download.jpg
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-uuid"
myuuid
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-tag"
mytag
--9431149156168
Content-Disposition: form-data; name="OSSAccessKeyId"
access-key-id
--9431149156168
Content-Disposition: form-data; name="policy"
encoded_policy
--9431149156168
Content-Disposition: form-data; name="Signature"
signature
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg
file_content
--9431149156168
Content-Disposition: form-data; name="submit"
Upload to OSS
--9431149156168--

```

Request headers

Note The message body of a PostObject request is encoded in the multipart/form-data format. In PostObject operations, parameters are passed as form fields in the request message body, whereas parameters are passed as HTTP request headers in PutObject operations.

Header	Type	Required	Example	Description
OSSAccessKeyId	String	Conditional	LTAI4Fw2NbDUCV8zYU*****	The AccessKey ID of the bucket owner. Default value: null Constraint: This form field is required when the bucket ACL is not public read/write or when the policy or Signature form field is provided.
policy	String	Conditional	<pre>{ "expiration": "2014-12-01T12:00:00.000Z", "conditions": [{ "bucket": "johnsmith" }, { "starts-with": "\$key", "user/eric/" }] }</pre>	Specifies the validity of the form fields in the request. A request that does not contain the policy form field is considered to be an anonymous request, and can be used only to access buckets whose ACLs are public read/write. For more information about the policy example, see Post policy. Default value: null Constraint: This form field is required when the bucket ACL is not public read/write or when the OSSAccessKeyId or Signature form field is provided. Note The form and the policy form field must be encoded in UTF-8.
Signature	String	Conditional	None	Specifies the signature information that is calculated based on the AccessKey secret and the policy form field. OSS checks the signature information to verify the validity of the PostObject request. Default value: null Constraint: This form field is required when the bucket ACL is not public read/write or when the OSSAccessKeyId or policy form field is provided. Note Form fields are case-insensitive, but their values are case-sensitive.

Header	Type	Required	Example	Description
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, and Expires	String	No	None	The HTTP request headers. For more information, see PutObject. Default value: null The form submitted by the PostObject operation must be encoded in the <code>multipart/form-data</code> format. For example, the Content-Type header must be in the <code>multipart/form-data;boundary=xxxxxx</code> format, where boundary is a boundary string.
x-oss-content-type	String	No	text/h323	You can add the x-oss-content-type header in the message body of a PostObject request to specify the Content-Type of the object to upload. The Content-Type value specified by the x-oss-content-type header has the highest priority. The content type of the object is determined by three headers in the following priority in descending order: x-oss-content-type > Content-Type in the form field > Content-Type. Default value: null
key	String	Yes	/user/a/objectName.txt	The name of the object to upload. If the object name includes a path such as a/b/c/b.jpg, OSS automatically creates corresponding folders. Default value: null
success_action_redirect	String	No	None	The URL to which the client is redirected after the object is uploaded. If this form field is not specified, the returned result is specified by success_action_status. If the upload fails, OSS returns an error code, and the client is not redirected to a URL. Default value: null
success_action_status	String	No	200	The status code returned to the client when the success_action_redirect form field is not specified and the object is uploaded. Default value: 204 Valid values: 200, 201, and 204. - If the value of this form field is set to 200 or 204, OSS returns an empty file and the 200 or 204 status code. - If the value of this form field is set to 201, OSS returns an XML file and the 201 status code. - If the value of this form field is not specified or set to an invalid value, OSS returns an empty file and the 204 status code.
x-oss-meta-*	String	No	x-oss-meta-location	The metadata customized by the user. Default value: null If the request contains a form field prefixed with x-oss-meta-, the form field is considered to be the user metadata. Example: x-oss-meta-location.  Note An object may have multiple similar parameters prefixed with x-oss-meta-, but the total size of the user metadata cannot exceed 8 KB.
x-oss-server-side-encryption	String	No	KMS	The server-side encryption algorithm that is used when OSS creates the object. Valid values: <i>AES256</i> and <i>KMS</i> If you specify this parameter, it is returned in the response header and the uploaded object is encrypted and stored. When you download the encrypted object, the x-oss-server-side-encryption field is included in the response header and the field value is set to the algorithm used to encrypt the object.
x-oss-server-side-encryption-key-id	String	No	3dea0cae-5fd6-4bf8-8574-87c557*****	The ID of the customer master key (CMK) hosted in KMS. This parameter is valid only when the value of x-oss-server-side-encryption is set to KMS.
x-oss-object-acl	String	No	private	The ACL of the object when the object is created. Valid values: <i>public-read</i>, <i>private</i>, and <i>public-read-write</i>

Header	Type	Required	Example	Description
x-oss-security-token	String	No	None	The security token for temporary authorization. If an STS temporary security credential is used for this access, you must set this field to the SecurityToken value and set OSSAccessKeyId to the value of the paired temporary AccessKey ID. The method to calculate a signature based on a temporary AccessKey ID is the same as that based on a typical AccessKey ID. Default value: null
file	String	Yes	MyFilename.txt	The file or text content. Browsers automatically set the Content-Type header based on the file type and overwrite the user setting. Only one file can be uploaded to OSS at a time. Default value: null  Notice This header must be the last field in the form.

Response headers

Header	Type	Example	Description
x-oss-server-side-encryption	String	KMS	If x-oss-server-side-encryption is specified in the request, the response contains this header to indicate the encryption algorithm used to encrypt the object at the server side.

Response elements

Element	Type	Example	Description
PostResponse	Container	N/A	The container that stores the result of the Post request.
Bucket	String	examplebucket	The bucket name.
ETag	String	E64478CDF7F0BFAF602F8D676D5B04E1	The entity tag (ETag) that is created when an object is generated. If an object is created by calling the PostObject operation, the ETag value is the object UUID, which can be used to check whether the object content is changed.
Location	String	https://examplebucket.endpoint/user/a/objectName.txt	The URL of the new object that is created.

Examples

Sample requests

```
POST / HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 344606
Content-Type: multipart/form-data; boundary=9431149156168
--9431149156168
Content-Disposition: form-data; name="key"
/user/a/${filename}
--9431149156168
Content-Disposition: form-data; name="success_action_status"
200
--9431149156168
Content-Disposition: form-data; name="Content-Disposition"
content_disposition
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-uuid"
uuid
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-tag"
metadata
--9431149156168
Content-Disposition: form-data; name="OSSAccessKeyId"
44CF9590006BF252F707
--9431149156168
Content-Disposition: form-data; name="policy"
eyJleHBpcnF0aW9uIjoiMjAxMjY0aWwvVQxMjowMDowMjY0LWxlbmd0aClyYW5nZSIsIDAsIDEwNDg1NzYwXSw7ImJlY2tldCI6ImFoYWhhIn0sIHsiQSI6ICJhIn0seyJrZXkiOiAiQUJDInlkdQ==
--9431149156168
Content-Disposition: form-data; name="Signature"
k***0+d***k=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.txt"
Content-Type: text/plain
abcdefg
--9431149156168
Content-Disposition: form-data; name="submit"
Upload to OSS
--9431149156168--
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 6***e
Date: Fri, 24 Feb 2014 06:03:28 GMT
ETag: 5***E
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Post policy

The policy form field in a PostObject request is used to verify the validity of the request. The policy value is a JSON text encoded in UTF-8 and Base64. This value declares the conditions that a PostObject request must meet. The policy form field is optional for uploading objects to a bucket whose ACL is public read/write. However, to limit PostObject requests, we recommend that you use this field.

- Policy example

```
{ "expiration": "2014-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "johnsmith" },
    ["starts-with", "$key", "user/eric/"]
  ]
}
```

In the PostObject request, the policy must contain the expiration and conditions attributes.

- Expiration

Expiration specifies the expiration time of the policy, which is expressed in ISO 8601 GMT. For example, "2014-12-01T12:00:00.000Z" indicates that the PostObject request must be sent before 12:00:00 on December 1, 2014.

- Conditions

Conditions is a list that specifies the valid values of form fields in the PostObject request. The value of a form field is extended after OSS checks the policy. Therefore, the valid value of the form field set in policy is equivalent to the value of the form field before extension. For example, if the key form field is set to user/user1/\${filename} and the object name before the upload is a.txt, the key form field in policy of the PostObject request must be set to ["eq", "\$key", "user/user1/a.txt"] instead of ["eq", "\$key", "user/user1/a.txt"]. The following table lists the conditions supported by the policy.

Conditions

Condition	Description
content-length-range	The acceptable maximum and minimum sizes of the uploaded file. This condition supports the content-length-range match mode.
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	HTTP request headers. This condition supports the exact match and starts-with match modes.
key	The name of the object after the upload. This condition supports the exact match and starts-with match modes.
success_action_redirect	The URL to which the client is redirected after the object is uploaded. This condition supports the exact match and starts-with match modes.
success_action_status	The status code returned after the object is uploaded if success_action_redirect is not specified. This condition supports the exact match and starts-with match modes.
x-oss-meta-*	The metadata customized by the user. This condition supports the exact match and starts-with match modes.

If the PostObject request contains other form fields, these extra form fields can be added to conditions of the policy. OSS does not check validity of the form fields that are not included in conditions.

- Condition match modes

Match mode	Description
Exact match	The value of a form field must be exactly the same as the value that is declared in conditions. For example, if the value of the key form field must be a, the condition can be [{"key": "a"}] or ["eq", "\$key", "a"].
Starts With	The value of a form field must start with the specified value. For example, if the value of the key form field must start with /user/user1, the condition is ["starts-with", "\$key", "/user/user1"].
Specified object size	The maximum and minimum sizes of the object that can be uploaded. For example, if the acceptable object size is 1 to 10 bytes, the condition is ["content-length-range", 1, 10].

- Escape characters

In the policy of a PostObject request, a dollar sign (\$) is used to indicate a variable. Therefore, to describe the dollar sign (\$), you must use escape characters (\\$). In addition, some characters in JSON are escaped. The following table describes escape characters for the JSON string of the policy in the PostObject request.

Escape characters

Escape character	Description
\/	Slash
\\	Backslash
\"	Double quotation marks

Escape character	Description
\\$	Dollar sign
\b	Backspace
\f	Form feed
\n	Line feed
\r	Carriage return
\t	Horizontal tab
\uxxxx	Unicode character

Post Signature

For signature authentication, the HTML form must contain policy and Signature. policy specifies the values that are valid in the request. The procedure to calculate the Signature field value:

1. Create a UTF-8 encoded policy.
2. Encode the policy in Base64. The encoding result is the value of the policy form field, and this value is used as the string-to-sign.
3. Use the AccessKey secret to sign the string. The signing method is the same as the method to calculate the signature in the header, which replaces the string-to-sign with the policy form field value.

1.2.6.2. ACL

1.2.6.2.1. PutObjectACL

Modifies the access control list (ACL) of an object.

When you call the PutObjectACL operation, you can set the x-oss-object-acl header in the request to configure the ACL of an object. Only bucket owners have permissions to configure object ACLs. If the operation succeeds, 200 OK is returned. Otherwise, the corresponding error code and message are returned. Object Storage Service (OSS) supports the following object ACLs: private, public read, public read/write, and default.

Versioning

By default, the PutObjectACL operation is called to configure the ACL of the current version of an object. You can specify a version ID in the request to configure the ACL of the specified version of an object. If the specified version is a delete marker, OSS returns 404 Not Found.

ACL overview

When you call the PutObjectACL operation, you can set the `x-oss-object-acl` header in the request to configure the ACL of an object. The following table describes the ACLs that you can configure for an object.

ACL	Description
private	The ACL of the object is private. Only the owner of the object has permissions to read and write the object. Other users cannot access the object.
public-read	The ACL of the object is public read. Only the owner of the object has permissions to read and write this object. Other users can only read the object.
public-read-write	The ACL of the object is public read/write. All users have permissions to read and write the object.
default	The ACL of the object is the same as the ACL of the bucket in which the object is stored.

Note

- The ACL of an object takes precedence over the ACL of the bucket in which the object is stored. For example, if an object whose ACL is public read/write is stored in a bucket whose ACL is private, all users can read and write the object. By default, if you do not configure the ACL of an object, the ACL of the object is the same as the ACL of the bucket in which the object is stored.
- The following operations read objects when they are called: GetObject, HeadObject, CopyObject, and UploadPartCopy. CopyObject and UploadPartCopy read the source object. The following operations write objects when they are called: PutObject, PostObject, AppendObject, DeleteObject, DeleteMultipleObjects, CompleteMultipartUpload, and CopyObject. CopyObject writes the destination object.
- When you call operations to write an object, you can also include the x-oss-object-acl header in the request to configure the ACL of the object. For example, you can include the x-oss-object-acl header in a PutObject request to configure the ACL of the object that you upload.

Request syntax

```
PUT /ObjectName?acl HTTP/1.1
x-oss-object-acl: Permission
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

Header	Type	Required	Example	Description

Header	Type	Required	Example	Description
x-oss-object-acl	String	Yes	private	The ACL of the object. Valid values: - private: Only authorized users can access the object. - public-read: All users can read the object. Only authorized users can modify the object. - public-read-write: All users, including anonymous users, can read and write the object. - default: The ACL of an object is the same as the ACL of the bucket in which the object is stored.

For more information about the other common request headers involved in the PutObjectACL operation, see [Common request headers](#).

Response headers

All headers in the response to a PutObjectACL request are common response headers. For more information, see [Common response headers](#).

Examples

- Modify the ACL of an object in an unversioned bucket

Sample requests

```
PUT /test-object?acl HTTP/1.1
x-oss-object-acl: public-read
Host: ***.regionid.example.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZ****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- Modify the ACL of an object in a versioned bucket

Sample requests

```
PUT /example?acl&versionId=CAEQMhiBgIC3rpSD0BYiIDBjYTk5MmIzN2JlNjQxZTFiNGIzM2E3OTliODA0**** HTTP/1.1
x-oss-object-acl: public-read
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 06:30:11 GMT
Authorization: OSS qctg2ns3l8u5liu:UTsv3F7L34v+ECq52vURdCSv****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-version-id: CAEQMhiBgIC3rpSD0BYiIDBjYTk5MmIzN2JlNjQxZTFiNGIzM2E3OTliODA0****
x-oss-request-id: 5CAC3BF3B7AEADE017000624
Date: Tue, 09 Apr 2019 06:30:11 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

1.2.6.2.2. GetObjectACL

Queries the ACL of an object.

Versioning

By default, when you call the GetObjectACL operation to query the ACL of an object, only the ACL of the current version of the object is returned. You can specify the versionId parameter in the request to query the ACL of a specified version of an object. If the specified version is a delete marker, Object Storage Service (OSS) returns 404 Not Found.

Note If you call the GetObjectACL operation to query the ACL of an object for which no ACL is configured, OSS returns the default ACL of this object. In this case, the ACL of this object is the same as the ACL of the bucket in which the object is stored. For example, if the ACL of the bucket in which the object is stored is private, the ACL of the object is also private.

Request syntax

```
GET /ObjectName?acl HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

All headers in a GetObjectACL request are common request headers. For more information about common request headers, see [Common request headers](#).

Response elements

Element	Type	Example	Description
AccessControlList	Container	N/A	The container that contains the ACL information.
AccessControlPolicy	Container	N/A	The container used to store the results of the GetObjectACL request.
DisplayName	String	1746495857*****	The name of the bucket owner, which is the same as the user ID.
Grant	Enumerated string	private	The ACL of the object. Default value: default. Valid values: private, public-read, and public-read-write
ID	String	1746495857*****	The user ID of the bucket owner.
Owner	Container	N/A	The container that stores the information about the bucket owner.

Examples

- Query the ACL of an object in an unversioned bucket

Sample requests

```
GET /test-object?acl HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAI4XZ+WwIfNm0Fmgb****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC75F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 253
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" ?>
<AccessControlPolicy>
  <Owner>
    <ID>00220120222</ID>
    <DisplayName>00220120222</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

- Query the ACL of an object in a versioned bucket

Sample requests

```
GET /example?acl&versionId=CAEQMhiBgMClqpSD0BYiIGQ0ZmI5ZDEyYWVvKNTQwMjBiNTliY2NjNmY3ZTVk**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 06:30:10 GMT
Authorization: OSS qctg2ns3l8u5liu:w4DK66Kb/0M9GJKdsrpNs811****
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-version-id: CAEQMhiBgMClqpSD0BYiIGQ0ZmI5ZDEyYWVvKNTQwMjBiNTliY2NjNmY3ZTVk****
x-oss-request-id: 5CAC3BF2B7AEADE017000621
Date: Tue, 09 Apr 2019 06:30:10 GMT
Content-Length: 261
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy>
  <Owner>
    <ID>1234513715092****</ID>
    <DisplayName>1234513715092****</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read</Grant>
  </AccessControlList>
</AccessControlPolicy>
```

1.2.6.3. Tagging

1.2.6.3.1. PutObjectTagging

You can call this operation to configure object tagging.

Usage notes

The object tagging feature uses a key-value pair to tag an object. When you call this operation, take note of the following items:

- **Limits on tags**
 - You can add up to 10 tags to an object. The tags added to an object must have unique tag keys.
 - A tag key can be up to 128 bytes in length. A tag value can be up to 256 bytes in length.
 - Tag keys and values are case-sensitive.
 - The key and value of a tag can contain letters, digits, spaces, and the following special characters:
+ - = . _ : /

If you configure the tags in the HTTP header and the tags contain any of the preceding characters, you can use URL encoding to encode the keys and values of the tags.
- **Last-Modified**

The Last-Modified value of an object is not updated when the object tags are changed.

Request structure

```
PUT /objectname? tagging
Content-Length: 114
Host: bucketname.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
<Tagging>
<TagSet>
  <Tag>
    <Key>Key</Key>
    <Value>Value</Value>
  </Tag>
</TagSet>
</Tagging>
```

Request elements

Element	Type	Required	Example	Description
Tagging	Container	Yes	N/A	The collection of tags.
TagSet	Container	Yes	N/A	The collection of tags.
Tag	Container	No	N/A	The collection of tags.
Key	String	No	xx	The key of the tag.
Value	String	No	1	The value of the object tag.

Response headers

Response headers involved in the PutObjectTagging operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
PUT /objectname? tagging
Content-Length: 114
Host: bucketname.oss-cn-hangzhou.aliyuncs.com
Date: Mon, 18 Mar 2019 08:25:17 GMT
Authorization: OSS *****:*****
<Tagging>
<TagSet>
  <Tag>
    <Key>a</Key>
    <Value>1</Value>
  </Tag>
  <Tag>
    <Key>b</Key>
    <Value>2</Value>
  </Tag>
</TagSet>
</Tagging>
```

Sample responses

```
200 (OK)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5C8F55ED461FB4A64C000004
date: Mon, 18 Mar 2019 08:25:17 GMT
```

1.2.6.3.2. GetObjectTagging

You can call this operation to query the tag information of an object.

Request structure

```
GET /objectname? tagging
Host: bucketname.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

The request headers involved in the PutObjectTagging operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

The response headers involved in the PutObjectTagging operation contain only common response headers. For more information, see [Common response headers](#).

Response elements

Element	Type	Example	Description
Tagging	Container	N/A	The collection of tags.
TagSet	Container	N/A	The collection of tags.
Tag	Container	N/A	The collection of tags.
Key	String	a	The key of the tag.
Value	String	1	The value of the tag.

Examples

Sample requests

```
GET /objectname? tagging
Host: bucketname.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 20 Mar 2019 02:02:36 GMT
Authorization: OSS *****:*****
```

Sample responses

```
200 (OK)
content-length: 209
server: AliyunOSS
x-oss-request-id: 5C919F38461FB42826000002
date: Wed, 20 Mar 2019 02:02:36 GMT
content-type: application/xml
<? xml version="1.0" encoding="UTF-8"? >
<Tagging>
  <TagSet>
    <Tag>
      <Key>a</Key>
      <Value>1</Value>
    </Tag>
    <Tag>
      <Key>b</Key>
      <Value>2</Value>
    </Tag>
  </TagSet>
</Tagging>
```

1.2.6.3.3. DeleteObjectTagging

You can call this operation to delete tags of a specific object.

Request structure

```
DELETE /objectname? tagging
Host: bucketname.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request headers

The request headers involved in the DeleteObjectTagging operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

The response headers involved in the DeleteObjectTagging operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
DELETE /objectname? tagging
Host: bucketname.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 03:00:33 GMT
Authorization: OSS LTAIbsTkySSptaz****/Zr0o6BKgAl7iiBtHN2JMC****
```

Sample responses

```
204 (No Content)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5CAC0AD16D0232E2051B****
date: Tue, 09 Apr 2019 03:00:33 GMT
```

1.2.6.4. Multipart upload

1.2.6.4.1. Overview

OSS provides multipart upload in addition to operations such as PutObject and PostObject.

Multipart upload can be used in the following scenarios but is not limited to:

- Resumable upload of the object is required.
- The size of the object you want to upload is larger than 5 GB.
- The OSS server is frequently disconnected due to poor network conditions.
- The size of the object you want to upload is unknown.

1.2.6.4.2. InitiateMultipartUpload

Before you use multipart upload to upload data, you must call the InitiateMultipartUpload operation to notify OSS to initiate a multipart upload task.

The OSS server returns a globally unique upload ID for the InitiateMultipartUpload operation. The upload ID uniquely identifies the multipart upload task. You can initiate operations such as stopping or querying the multipart upload task based on this upload ID.

Usage notes

- When you perform this operation to calculate the signature for authentication, you must add `?uploads` to CanonicalizedResource.
- If you set the x-oss-server-side-encryption header in the InitiateMultipartUpload request, the response contains this header. When each part is uploaded, the server encrypts and stores them. OSS server-side encryption supports the AES-256 and SM4 algorithms. If other algorithms are specified, OSS returns HTTP status code 400 and InvalidEncryptionAlgorithmError. When you upload each part, you do not need to add the x-oss-server-side-encryption request header. If this header is specified, OSS returns HTTP status code 400 and InvalidArgument.

Request structure

```
POST /ObjectName? uploads HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT date
Authorization: SignatureValue
```

Request parameters

The encoding-type parameter can be specified in the InitiateMultipartUpload request. OSS uses the specified encoding type to encode the object name in the response.

Parameter	Type	Required	Example	Description
encoding-type	String	No	url	The encoding type of the object names in the response. The object name can contain characters that are encoded in UTF-8. However, the XML 1.0 standard cannot be used to parse certain control characters such as characters with an ASCII value from 0 to 10. You can set the encoding-type parameter to encode the object names in the response. Set the value to url. Default value: null Valid value: url

Request headers

Header	Type	Required	Example	Description
Cache-Control	String	No	no-cache	The web page caching behavior that is specified when the object is downloaded. For more information, visit RFC 2616 . Type: string. Default value: null
Content-Disposition	String	No	attachment;filename=oss_download.jpg	The presentational information of the object during the download. For more information, visit RFC 2616 . Type: string. Default value: null

Header	Type	Required	Example	Description
Content-Encoding	String	No	utf-8	The content encoding type of the object during the download. For more information, visit RFC 2616 . Type: string. Default value: null
Expires	Integer	No	None	The expiration time in milliseconds. For more information, visit RFC 2616 . Default value: null
x-oss-server-side-encryption	String	No	AES256	The encryption algorithm that OSS uses to encrypt each uploaded part of an object before storage. Valid values: AES256 and SM4

Response elements

Element	Type	Example	Description
Bucket	String	multipart_upload	The name of the bucket to which the object is uploaded by the multipart upload task.
InitiateMultipartUploadResult	Container	N/A	The container that is used to store the result of the InitiateMultipartUpload request.
Key	String	multipart.data	The name of the object that is uploaded by the multipart upload task.
UploadId	String	0***D	The unique ID of the multipart upload task.
EncodingType	String	url	The encoding type of the object name in the response. If the encoding-type parameter is specified in the request, the object name in the response is encoded.

Examples

Sample requests

```
POST /multipart.data?uploads HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS q***4=
```

Sample responses

```
HTTP/1.1 200 OK
Content-Length: 230
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 4***8
Date: Wed, 22 Feb 2012 08:32:21 GMT
Content-Type: application/xml
<? xml version="1.0" encoding="UTF-8"? >
<InitiateMultipartUploadResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Bucket>multipart_upload</Bucket>
  <Key>multipart.data</Key>
  <UploadId>0***D</UploadId>
</InitiateMultipartUploadResult>
```

1.2.6.4.3. UploadPart

After initializing a MultipartUpload task, you can call this operation to upload data in parts based on the specified object name and upload ID.

Usage notes

- Before you call UploadPart to upload parts, you must call InitiateMultipartUpload to obtain an upload ID issued by the OSS server.
- The part number ranges from 1 to 10000. If a part number is not within this range, OSS returns the InvalidArgument error code.
- In the multipart upload mode, the minimum size of each part except the last part is 100 KB. The UploadPart operation does not immediately verify the sizes of the uploaded parts because this operation cannot immediately determine which part is the last part. The size of the uploaded part is verified only when multipart upload is complete.
- OSS includes the MD5 hash for each part received by the server in the ETag header and returns the ETag header to the user.
- If the x-oss-server-side-encryption request header is specified when you call InitiateMultipartUpload, the uploaded parts are encoded. The x-oss-server-side-encryption header is included in the response to the UploadPart request, which indicates the server-side encryption method of the parts. For more information, see [InitiateMultipartUpload](#).

Request structure

```
PUT /ObjectName? partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
```

Request headers

The request headers involved in the UploadPart operation contain only common request headers. For more information, see [Common request headers](#).

Response headers

The response headers involved in the UploadPart operation contains only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
PUT /multipart.data? partNumber=1&uploadId=0***6 HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 6291456
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS g***I=
[6291456 bytes data]
```

Sample responses

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
ETag: 7***9
x-oss-request-id: 3***a
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

1.2.6.4.4. UploadPartCopy

Copies data from an existing object to upload a part.

Usage notes

To copy an object larger than 1 GB in size, use UploadPartCopy. For more information about how to upload an object smaller than 1 GB in size, see [CopyObject](#).

When you call UploadPartCopy, take note of the following items:

- Objects uploaded by calling AppendObject cannot be copied.
- When you call UploadPartCopy, the source and destination buckets must be located within the same region.
- Before you call UploadPartCopy to upload a part, you must call InitiateMultipartUpload to obtain an upload ID issued by the Object Storage Service (OSS) server.
- If the x-oss-server-side-encryption request header is specified when you call InitiateMultipartUpload, the uploaded part is encoded. The x-oss-server-side-encryption header is included in the response to the UploadPart request, which indicates the server-side encryption method of the part.
- In a multipart upload task, each part must be larger than 100 KB in size, except the last part. The size of each part is not verified when you call UploadPart because not all parts are uploaded and OSS does not know which part is the last part. The size of each part is verified only when you call CompleteMultipartUpload.

Versioning

By default, the UploadPartCopy operation uploads a part by copying data from the current version of an existing object. You can add versionId to the x-oss-copy-source request header as the condition to upload a part by copying data from the specified version of an existing object. Example: x-oss-copy-source: /SourceBucketName/SourceObjectName?versionId=111111.

 **Note** The value of SourceObjectName must be URL-encoded. x-oss-copy-source-version-id is returned in the response. The value of x-oss-copy-source-version-id indicates the version ID of the source object.

If versionId is not specified and the current version of the source object is a delete marker, OSS returns 404 Not Found. If versionId is specified in the request and the specified version of the source object is a delete marker, OSS returns 400 Bad Request.

Request syntax

```
PUT /ObjectName? partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
x-oss-copy-source: /SourceBucketName/SourceObjectName
x-oss-copy-source-range: bytes=first-last
```

Request headers

In addition to common request headers, you can use the following headers in an UploadPartCopy request to specify the address of the source object and the range of data that you want to copy.

Header	Type	Required	Example	Description

Header	Type	Required	Example	Description
x-oss-copy-source	String	Yes	oss-example/src-object	The address of the source object. You must have permissions to read the source object. This parameter is empty by default.
x-oss-copy-source-range	Integer	No	bytes=100-6291756	The range of bytes to copy from the source object. For example, if you set bytes to 0~9, OSS transfers a total of 10 bytes that range from byte 0 to byte 9. This parameter is empty by default. - If the x-oss-copy-source-range request header is not specified, the entire source object is copied. - If the x-oss-copy-source-range request header is specified, the response contains the length of the entire object and the range of bytes that are copied for this operation. For example, Content-Range: bytes 0~9/44 indicates that the length of the entire object is 44 bytes and the range of bytes copied is byte 0 to byte 9. - If the specified range does not conform to the range conventions, OSS copies the entire object and does not include Content-Range in the response.

The following table describes request headers that are used for the source object specified by x-oss-copy-source.

Header	Type	Required	Example	Description
x-oss-copy-source-if-match	String	No	5B3C1A2E053D763E1B002CC607C5****	The data copy operation condition. If the ETag value of the source object is the same as the ETag value provided by the user, OSS copies data from the object. Otherwise, OSS returns 412 Precondition Failed. This parameter is empty by default.
x-oss-copy-source-if-none-match	String	No	2019-04-09T07:01:56.000Z	The data copy operation condition. If the source object is not modified after the time specified in the request, OSS copies data from the object. Otherwise, OSS returns 412 Precondition Failed. This parameter is empty by default.
x-oss-copy-source-if-unmodified-since	String	No	2019-04-09T07:01:56.000Z	The object copy condition. If the specified time is later than or equal to the actual time when the object is modified, OSS transfers the object and returns the 200 OK message. Otherwise, OSS returns 412 Precondition Failed. This parameter is empty by default.
x-oss-copy-source-if-modified-since	String	No	2019-04-09T07:01:56.000Z	The data copy operation condition. If the source object is modified after the time specified in the request, OSS copies data from the object. Otherwise, OSS returns 412 Precondition Failed. This parameter is empty by default.

Response elements

Element	Type	Example	Description
CopyObjectResult	Container	N/A	The result of the CopyObject operation.
LastModified	String	2019-04-09T07:01:56.000Z	The time when the destination object was last modified.
ETag	String	5B3C1A2E053D763E1B002CC607C5****	The ETag value of the destination object.

Examples

- Upload a part by calling UploadPartCopy on an object in an unversioned bucket

Sample requests

```
PUT /multipart.data?partNumber=1&uploadId=0004B9895DBBB6EC98E36 HTTP/1.1
Host: *.regionid.example.com
Content-Length: 6291456
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****;J/1ICfXEvPmmSW86bBAfMmUm****
x-oss-copy-source: /oss-example/ src-object
x-oss-copy-source-range: bytes=100-6291756
```

Sample responses

```

HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0****
Date: Thu, 17 Jul 2014 06:27:54 GMT'
<?xml version="1.0" encoding="UTF-8"?>
<CopyPartResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <LastModified>2014-07-17T06:27:54.000Z </LastModified>
  <ETag>"5B3C1A2E053D763E1B002CC607C5*****"</ETag>
</CopyPartResult>

```

- Upload a part by calling UploadPartCopy on an object with the specified version ID in a versioned bucket

Sample requests

```

PUT /multipart.data?partNumber=2&uploadId=63C06A5CFF6F4AE4A6BB3AD7F01C**** HTTP/1.1
Host: ***.regionid.example.com
Date: Tue, 09 Apr 2019 07:01:56 GMT
Authorization: OSS 6jfttm6x6san0ewtgyk****:v/sJFtYvg7DTa4pJ2AMShZL/****
x-oss-copy-source: /oss-example/src-object?versionId=CAEQMxiBgMC0vs6D0BYiIGUjZWRjOTRjNTg0NzQ1MTRiN2Y1OTYxMTdkYjQ0****

```

Sample responses

```

HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
x-oss-copy-source-version-id: CAEQMxiBgMC0vs6D0BYiIGUjZWRjOTRjNTg0NzQ1MTRiN2Y1OTYxMTdkYjQ0****
x-oss-request-id: 5CAC4364B7AEADE017000660
Date: Tue, 09 Apr 2019 07:01:56 GMT
<?xml version="1.0" encoding="UTF-8"?>
<CopyPartResult>
  <LastModified>2019-04-09T07:01:56.000Z</LastModified>
  <ETag>"25A9F4ABFCC05743DF6E2C886C56*****"</ETag>
</CopyPartResult>

```

1.2.6.4.5. CompleteMultipartUpload

Completes multipart upload of an object after all parts of the object are uploaded.

Usage notes

When you call the CompleteMultipartUpload operation, you must provide a complete list of all valid parts, including the part number and entity tag (ETag) of each part. After Object Storage Service (OSS) receives the list of parts, OSS verifies the validity of each part one after another. After all parts are validated, OSS combines these parts into a complete object.

- Confirm the size of each part

When you call the CompleteMultipartUpload operation, OSS checks whether each part except the last part is larger than or equal to 100 KB in size and verifies the part number and ETag of each part provided in the part list. Therefore, when a part is uploaded, the client must record not only the part number of the part but also the ETag value of the part returned from the server.

- Handle requests

It may take a period of time for OSS to process a CompleteMultipartUpload request. If the client is disconnected from OSS during the period, OSS continues to process the request.

- PartNumber

OSS verifies the value of PartNumber when you call the CompleteMultipartUpload operation.

The value of PartNumber ranges from 1 to 10000. The part numbers listed in the request can be non-consecutive but must be sorted in ascending order. For example, if the part number of the first part is 1, the part number of the second part can be 5.

- UploadId

An object can be uploaded by multiple upload tasks that have independent upload IDs. When one upload task is completed, its upload ID becomes invalid but the upload IDs of other upload tasks are not affected.

- x-oss-server-side-encryption

If the x-oss-server-side-encryption header is specified in an InitiateMultipartUpload request, this header is returned in the response to the request. The value of the x-oss-server-side-encryption header in the response indicates the method used to encrypt the object on the OSS server.

Versioning

You can call the CompleteMultipartUpload operation to complete the multipart upload task of an object when versioning is enabled for the bucket to which the object is uploaded. In this case, OSS generates a unique version ID for the object, and returns the version ID as the x-oss-version-id header in the response.

Request syntax


```

POST /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Content-Length: Size
Authorization: Signature
<CompleteMultipartUpload>
<Part>
<PartNumber>PartNumber</PartNumber>
<ETag>ETag</ETag>
</Part>
...
</CompleteMultipartUpload>

```

Request headers

Header	Type	Required	Description
x-oss-complete-all:yes	String	No	Specifies whether to list all parts that are uploaded by using the current upload ID. - If x-oss-complete-all:yes is specified in the request, OSS lists all parts that are uploaded by using the current upload ID, sorts the parts by their part numbers, and then performs the CompleteMultipartUpload operation. OSS cannot detect the parts that are not uploaded or being uploaded when OSS performs the CompleteMultipartUpload operation. Before you call the CompleteMultipartUpload operation, make sure that all parts are uploaded. - If x-oss-complete-all:yes is specified in the request, the request body cannot be specified. Otherwise, an error occurs. - The format of the response remains unchanged if x-oss-complete-all:yes is specified in the request.

For more information about other common request headers such as Host and Date, see [Common request headers](#).

Request elements

Element	Type	Required	Example	Description
CompleteMultipartUpload	Container	Yes	N/A	The container that stores the content of the CompleteMultipartUpload request.
ETag	String	No	3***9	The ETag values that are returned by OSS after parts are uploaded.
Part	Container	Yes	N/A	The container that stores the uploaded parts.
PartNumber	Integer	Yes	1	The number of parts.

Response elements

Element	Type	Example	Description
Bucket	String	oss-example	The name of the bucket.
CompleteMultipartUploadResult	Container	N/A	The container that stores the response to the CompleteMultipartUpload request.
ETag	String	B***D-3	The UUID of the object content. The ETag is generated to identify the content of an object when the object is created by using the CompleteMultipartUpload request. You can use the ETag value of an object to check whether the object content is modified.
Location	String	http://oss-example.endpoint/multipart.data	The URL that is used to access the uploaded object.
Key	String	multipart.data	The name of the uploaded object.
EncodingType	String	utf-8	The encoding type of the object name in the response. If the encoding-type parameter is specified in the request, the object name in the response is encoded.

Examples

- Complete the multipart upload task of an object in an unversioned bucket
- Sample requests

```
POST /multipart.data?uploadId=0004B9B2D2F7815C432C9057C031****&encoding-type=url HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 1056
Date: Fri, 24 Feb 2012 10:19:18 GMT
Authorization: OSS qn6qrrqx02oawuk53otf****:8VwFhFUWmVecK6jQlHlXMK/z****
<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"3349DC700140D7F86A0784842780****"</ETag>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <ETag>"8EFDA8BE206636A695359836FE0A****"</ETag>
  </Part>
  <Part>
    <PartNumber>8</PartNumber>
    <ETag>"8C31506516713244177411FDA14****"</ETag>
  </Part>
</CompleteMultipartUpload>
```

Sample responses

```
HTTP/1.1 200 OK
Server: AliyunOSS
Content-Length: 329
Content-Type: Application/xml
Connection: keep-alive
x-oss-request-id: 594f0751-3b1e-168f-4501-4ac71d21****
Date: Fri, 24 Feb 2012 10:19:18 GMT
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <EncodingType>url</EncodingType>
  <Location>http://oss-example.oss-cn-hangzhou.aliyuncs.com /multipart.data</Location>
  <Bucket>oss-example</Bucket>
  <Key>multipart.data</Key>
  <ETag>"B864DB6A936D376F9F8D3ED3BBE540****"</ETag>
</CompleteMultipartUploadResult>
```

- Complete the multipart upload task of an object in a versioned bucket

Sample requests

```
POST /multipart.data?uploadId=63C06A5CFF6F4AE4A6BB3AD7F01C**** HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 223
Date: Tue, 09 Apr 2019 07:01:56 GMT
Authorization: OSS 6jfttm6x6s****:XljBrYBYxDnxKdFMj9WYI6qu****
<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"25A9F4ABFCC05743DF6E2C886C56****"</ETag>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <ETag>"25A9F4ABFCC05743DF6E2C886C56****"</ETag>
  </Part>
</CompleteMultipartUpload>
```

Sample responses

```
HTTP/1.1 200 OK
Server: AliyunOSS
Content-Length: 314
Content-Type: Application/xml
Connection: keep-alive
x-oss-version-id: CAEQMxiBgID6v86D0BYiIDc3ZDI0YTBjZGQzYjQ2Mjk4OWVjYWNiMDljYzhln****
x-oss-request-id: 5CAC4364B7AEADE017000662
Date: Tue, 09 Apr 2019 07:01:56 GMT
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult>
  <Location>http://oss-example.oss-cn-hangzhou.aliyuncs.com/multipart.data</Location>
  <Bucket>oss-example</Bucket>
  <Key>multipart.data</Key>
  <ETag>"097DE458AD02B5F89F9D0530231876****"</ETag>
</CompleteMultipartUploadResult>
```

1.2.6.4.6. AbortMultipartUpload

You can call this operation to cancel a multipart upload task based on the specified upload ID.

Usage notes

When you call AbortMultipartUpload, take note of the following items:

- You must provide the upload ID of the multipart upload task that you want to cancel.
- When you cancel a multipart upload task, parts that are being uploaded cannot be deleted.
- After a multipart upload task is canceled, the upload ID of the task cannot be used to perform any operations and the parts uploaded by the task are deleted.

- We recommend that you complete or cancel multipart upload tasks that are not complete in a timely manner because parts generated by these tasks consume storage space and incur storage fees.

Request structure

```
DELETE /ObjectName? uploadId=UploadId HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: Signature
```

Request elements

Element	Type	Required	Example	Description
uploadId	String	Yes	F18A92392DFD4B3FA897C26782*****	The ID of the multipart upload task.

For more information about other common request headers such as Host and Date, see [Common request headers](#).

Response headers

Response headers involved in the AbortMultipartUpload operation contain only common response headers. For more information, see [Common response headers](#).

Examples

Sample requests

```
Delete /multipart.data? &uploadId=0***E HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS q***I=
```

Sample responses

```
HTTP/1.1 204
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 0***d
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

1.2.6.4.7. ListMultipartUploads

You can call this operation to list all ongoing multipart upload tasks that have been initiated but not complete or canceled.

The list returned by OSS contains a maximum of 1,000 multipart upload tasks. To specify the number of tasks that are returned in the response from OSS, you can add the max-uploads parameter to the request. In the response returned by OSS, the IsTruncated element indicates whether the response contains all the ongoing multipart upload tasks.

Request structure

```
Get /? uploads HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: Signature
```

Request parameters

Parameter	Type	Required	Example	Description
delimiter	String	No	None	The delimiter used to group objects by name. If you specify the Delimiter parameter in the request, the response contains the CommonPrefixes element. Objects whose names contain the same string from the prefix to the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes.
max-uploads	String	No	1000	The maximum number of multipart upload tasks to return for one request. If this parameter is not specified, default value 1000 applies. The max-uploads value cannot exceed 1000. Type: string

Parameter	Type	Required	Example	Description
key-marker	String	No	None	The name of the object after which the list begins. This parameter is used together with the upload-id-marker parameter. - If the upload-id-marker parameter is not set, OSS returns multipart upload tasks in which object names are listed after the key-marker value. Multipart upload tasks are listed in alphabetical order of their object names. - If the upload-id-marker parameter is set, OSS returns the following tasks: Multipart upload tasks in which object names are listed after the key-marker value. Multipart upload tasks are listed in alphabetical order of their object names. Multipart upload tasks in which object names are the key-marker value but upload IDs are greater than the upload-id-marker value.
prefix	String	No	None	The prefix that the returned object names must contain. Note that if you use a prefix to make a query, the returned object names still contain the prefix.
upload-id-marker	String	No	None	The upload ID of the multipart upload task after which the list begins. This parameter is used together with the key-marker parameter. - If the key-marker parameter is not set, OSS ignores the upload-id-marker parameter. - If the key-marker is set, the response includes the following tasks: Multipart upload tasks in which object names are listed after the key-marker value. Multipart upload tasks are listed in alphabetical order of their object names. Multipart upload tasks in which object names are equal to the key-marker parameter value but upload IDs are greater than the upload-id-marker parameter value.
encoding-type	String	No	utf-8	The encoding type of the object names in the response. Values of Delimiter, KeyMarker, Prefix, NextKeyMarker, and Key can be encoded in UTF-8. However, the XML 1.0 standard cannot be used to parse control characters such as characters with an ASCII value 0 to 10. You can set the encoding-type parameter to encode values of Delimiter, KeyMarker, Prefix, NextKeyMarker, and Key in the response. Default value: null

Response elements

Element	Type	Example	Description
ListMultipartUploadsResult	Container	N/A	The container that stores the response of the ListMultipartUpload request.
Bucket	String	oss-example	The name of the bucket.
EncodingType	String	UTF-8	The encoding type of the object name in the response. If encoding-type is specified in the request, values of those elements including Delimiter, KeyMarker, Prefix, NextKeyMarker, and Key are encoded in the returned result.
KeyMarker	String	5	The name of the object after which the list begins.
UploadIdMarker	String	None	The upload ID of the multipart upload task after which the list begins.
NextKeyMarker	String	oss.avi	The object name marker in the response for the next request to return the remaining results.
NextUploadMarker	String	0004B99B8E707874FC2D692FA5D77D3F	The NextUploadMarker value that is used for the UploadMarker value in a subsequent request when the response does not contain all required results.
MaxUploads	Integer	1000	The maximum number of upload tasks returned by OSS.
IsTruncated	Enumerated string	false	Indicates whether the multipart upload task list returned in this response is truncated. A value of true indicates that the response does not contain all required results. A value of false indicates that the response contains all required results. Valid values: false and true Default value: false
Upload	Container	N/A	The container that stores the information about multipart upload tasks.
Key	String	multipart.data	The object name in an initiated multipart upload task.
UploadId	String	0004B999EF518A1FE585B0C9360DC4C8	The ID of the multipart upload task.
Initiated	Release date	2012-02-23T04:18:23.000Z	The time when the multipart upload task was initiated.

Detail analysis

- The maximum value of the max-uploads parameter is 1000.
- The results returned by OSS are listed based on the alphabetical order of object names. If multipart upload tasks involve the same object, the results are listed in

ascending order of time.

- You can use prefixes to group and manage objects in a bucket in the same way you manage a folder in a file system.
- ListMultipartUploads supports the following request parameters: prefix, marker, delimiter, upload-id-marker, and max-uploads. You can use one or more of the preceding parameters to configure rules to query multipart upload tasks so that results that meet conditions are returned.

Examples

Sample requests

```
Get /? uploads HTTP/1.1
Host:***.regionid.example.com
Date: Thu, 23 Feb 2012 06:14:27 GMT
Authorization: OSS q***Y=
```

Sample responses

```
HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1839
Content-type: application/xml
x-oss-request-id: 5***a
Date: Thu, 23 Feb 2012 06:14:27 GMT
<? xml version="1.0" encoding="UTF-8"? >
<ListMultipartUploadsResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Bucket>oss-example</Bucket>
  <KeyMarker></KeyMarker>
  <UploadIdMarker></UploadIdMarker>
  <NextKeyMarker>oss.avi</NextKeyMarker>
  <NextUploadIdMarker>0***F</NextUploadIdMarker>
  <Delimiter></Delimiter>
  <Prefix></Prefix>
  <MaxUploads>1000</MaxUploads>
  <IsTruncated>false</IsTruncated>
  <Upload>
    <Key>multipart.data</Key>
    <UploadId>0***8</UploadId>
    <Initiated>2012-02-23T04:18:23.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>multipart.data</Key>
    <UploadId>0***5</UploadId>
    <Initiated>2012-02-23T04:18:23.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>oss.avi</Key>
    <UploadId>0***F</UploadId>
    <Initiated>2012-02-23T06:14:27.000Z</Initiated>
  </Upload>
</ListMultipartUploadsResult>
```

1.2.6.4.8. ListParts

You can call this operation to list all uploaded parts that are mapped to a specific upload ID.

Usage notes

- ListParts supports two request parameters: max-parts and part-number-marker.
- The maximum value of the max-parts parameter is 1000. The default value is 1000.
- The results returned by OSS are listed based on the ascending order of part numbers.
- Errors may occur during network transmission. We recommend that you do not use the results (part numbers and ET ag values) of ListParts to generate the final part list of CompleteMultipart.

Request structure

```
Get /ObjectName? uploadId=UploadId HTTP/1.1
Host: BucketName.regionid.example.com
Date: GMT Date
Authorization: Signature
```

Request parameters

Parameter	Type	Required	Example	Description
uploadId	String	No	0004B9999EF5A239BB9138C6227D69F95	The ID of the multipart upload task. Default value: null

Parameter	Type	Required	Example	Description
max-parts	Integer	No	1000	The maximum number of parts that can be returned by OSS. Type: integer Default value: 1000
part-number-marker	Integer	No	5	The number of the part after which the list begins. All parts whose part numbers are greater than this parameter value are listed. Type: integer Default value: null
encoding-type	String	No	url	The encoding type of the object name in the response. The object name can contain any characters that are encoded in UTF-8. However, the XML 1.0 standard cannot be used to parse control characters such as characters with an ASCII value from 0 to 10. You can set the encoding-type parameter to encode the object names in the response. Set the value to url.

Response elements

Element	Type	Example	Description
ListPartsResult	Container	N/A	The container that stores the response to the ListParts request.
Bucket	String	multipart_upload	The name of the bucket.
EncodingType	String	url	The encoding type of the object name in the response. If the encoding-type parameter is specified in the request, the object name in the response is encoded.
Key	String	multipart.data	The name of the object.
UploadId	String	0004B999EF5A239BB9 138C6227D69F95	The ID of an upload task.
PartNumberMarker	Integer	5	The number of the part after which the list begins. All parts whose part numbers are greater than this parameter value are listed.
NextPartNumberMarker	Integer	5	The NextPartNumberMarker value that is used for the PartNumberMarker value in a subsequent request when the response does not contain all required results.
MaxParts	Integer	1000	The maximum number of parts in this response.
IsTruncated	Enumerated string	false	Indicates whether the part list returned in this response is truncated. A value of true indicates that the response does not contain all required results. A value of false indicates that the response contains all required results. Valid values: true and false
Part	Container	N/A	The container that stores part information.
PartNumber	Integer	1	The number that identifies a part.
LastModified	Release date	2012-02- 23T07:01:34.000Z	The time when the part was uploaded.
ETag	String	3349DC700140D7F86A 0784842780****	The ETag value of the content of the uploaded part. Type: Parent nodes: ListPartsResult and Part
Size	Integer	6291456	The size of the uploaded part.

Examples

Sample requests

```
Get /multipart.data? uploadId=0***5 HTTP/1.1
Host: ***.regionid.example.com
Date: Thu, 23 Feb 2012 07:13:28 GMT
Authorization: OSS q***8=
```

Sample responses

```

HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1221
Content-type: application/xml
x-oss-request-id: 1***c
Date: Thu, 23 Feb 2012 07:13:28 GMT
<? xml version="1.0" encoding="UTF-8"? >
<ListPartsResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Bucket>multipart_upload</Bucket>
  <Key>multipart.data</Key>
  <UploadId>0***5</UploadId>
  <NextPartNumberMarker>5</NextPartNumberMarker>
  <MaxParts>1000</MaxParts>
  <IsTruncated>false</IsTruncated>
  <Part>
    <PartNumber>1</PartNumber>
    <LastModified>2012-02-23T07:01:34.000Z</LastModified>
    <ETag>3***9</ETag>
    <Size>6291456</Size>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <LastModified>2012-02-23T07:01:12.000Z</LastModified>
    <ETag>3***9</ETag>
    <Size>6291456</Size>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <LastModified>2012-02-23T07:02:03.000Z</LastModified>
    <ETag>7***9</ETag>
    <Size>1024</Size>
  </Part>
</ListPartsResult>

```

1.2.7. Error responses

If an error occurs when you access OSS, OSS returns the error code and error message so that you can locate and rectify the error.

OSS error response format

If an error occurs when you access OSS, OSS returns HTTP status code 3xx, 4xx, or 5xx and a message body in the application/xml format.

Example of an error response body:

```

<? xml version="1.0" ? >
<Error xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
  <Code>
    AccessDenied
  </Code>
  <Message>
    Query-string authentication requires the Signature, Expires and OSSAccessKeyId parameters
  </Message>
  <RequestId>
    1D842BC5425544BB
  </RequestId>
  <HostId>
    regionid.example.com
  </HostId>
</Error>

```

All error message bodies include the following elements:

- Code: the error code that OSS returns to the user.
- Message: the detailed error message provided by OSS.
- RequestId: the UUID that uniquely identifies a request. If a problem persists, submit the request ID to OSS technical support personnel to seek help.
- HostId: the ID of the host in the accessed OSS cluster, which is the same as the host ID specified in the request.

For more information about the elements in error messages, see specific request descriptions.

Error codes

The following table lists the OSS error codes.

Error codes

Error code	Description	HTTP status code
AccessDenied	The error message returned because the access is denied.	403
BucketAlreadyExists	The error message returned because the bucket already exists.	409

Error code	Description	HTTP status code
BucketNotEmpty	The error message returned because the bucket is not empty.	409
EntityTooLarge	The error message returned because the entity is too large.	400
EntityTooSmall	The error message returned because the entity is too small.	400
FieldItemTooLong	The error message returned because the total length of the form fields in the POST request exceeds the limit.	400
FilePartIntegrity	The error message returned because the part has changed.	400
FilePartNotExist	The error message returned because the part does not exist.	400
FilePartStale	The error message returned because the part has expired.	400
IncorrectNumberOfFilesInPOSTRequest	The error message returned because the number of objects in the POST request is invalid.	400
InvalidArgument	The error message returned because the parameter format is invalid.	400
InvalidAccessKeyId	The error message returned because the AccessKey ID does not exist.	403
InvalidBucketName	The error message returned because the bucket name is invalid.	400
InvalidDigest	The error message returned because Content-MD5 you specified is invalid.	400
InvalidEncryptionAlgorithmError	The error message returned because the specified encryption algorithm based on entropy encoding is invalid.	400
InvalidObjectName	The error message returned because the object name is invalid.	400
InvalidPart	The error message returned because the part is invalid.	400
InvalidPartOrder	The error message returned because the part sequence is invalid.	400
InvalidPolicyDocument	The error message returned because the content of the form does not meet the conditions specified in the policy document.	400
InvalidTargetBucketForLogging	The error message returned because the destination bucket specified for logging is invalid.	400
InternalError	The error message returned because an internal OSS error occurs.	500
MalformedXML	The error message returned because the XML format is invalid.	400
MalformedPOSTRequest	The error message returned because the format of the POST request body is invalid.	400
MaxPOSTPreDataLengthExceededError	The error message returned because the POST request body, excluding the object content to upload, is too large.	400
MethodNotAllowed	The error message returned because the method is not supported.	405
MissingArgument	The error message returned because the required parameters are not set.	411
MissingContentLength	The error message returned because the content length is not specified.	411
NoSuchBucket	The error message returned because the specified bucket does not exist.	404
NoSuchKey	The error message returned because the specified object does not exist.	404
NoSuchUpload	The error message returned because the specified multipart upload ID does not exist.	404

Error code	Description	HTTP status code
NotImplemented	The error message returned because the method cannot be implemented.	400
PreconditionFailed	The error message returned because the precondition you specified does not hold.	412
RequestTimeTooSkewed	The error message returned because the difference between the time on the client when the request is sent and the time on the server when the request is received is greater than 15 minutes.	403
RequestTimeout	The error message returned because the request times out.	400
RequestIsNotMultiPartContent	The error message returned because the content-type value specified in the POST request is invalid.	400
SignatureDoesNotMatch	The error message returned because the request signature provided does not match the signature calculated by OSS.	403
TooManyBuckets	The error message returned because the number of the buckets in your account has exceeded the limit.	400

Operations not supported by OSS

If an operation not supported by OSS is performed to access a resource, OSS returns HTTP status code 405 and error code MethodNotAllowed.

Sample error requests

```
ABC /1.txt HTTP/1.1
Host: bucketname.regionid.example.com
Date: Thu, 11 Aug 2016 03:53:40 GMT
Authorization: signatureValue
```

Sample responses

```
HTTP/1.1 405 Method Not Allowed
Server: AliyunOSS
Date: Thu, 11 Aug 2016 03:53:44 GMT
Content-Type: application/xml
Content-Length: 338
Connection: keep-alive
x-oss-request-id: 57ABF6C8BC4D25D86CBA5ADE
Allow: GET DELETE HEAD PUT POST OPTIONS
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <Code>MethodNotAllowed</Code>
  <Message>The specified method is not allowed against this resource. </Message>
  <RequestId>57ABF6C8BC4D25D86CBA5ADE</RequestId>
  <HostId>bucketname.regionid.example.com</HostId>
  <Method>abc</Method>
  <ResourceType>Bucket</ResourceType>
</Error>
```

Note If the accessed resource is /bucket/, ResourceType must be set to Bucket. If the accessed resource is /bucket/object, ResourceType must be set to Object.

Operations supported by OSS but not supported by parameters

If parameters not supported by OSS are added to the request for the object supported by OSS (for example, the If-Modified-Since parameter is added to the PUT operation request), OSS returns 400 Bad Request.

Sample error requests

```
PUT /abc.zip HTTP/1.1
Host: bucketname.regionid.example.com
Accept: */*
Date: Thu, 11 Aug 2016 01:44:50 GMT
If-Modified-Since: Thu, 11 Aug 2016 01:43:51 GMT
Content-Length: 363
```

Sample responses

```

HTTP/1.1 400 Bad Request
Server: AliyunOSS
Date: Thu, 11 Aug 2016 01:44:54 GMT
Content-Type: application/xml
Content-Length: 322
Connection: keep-alive
x-oss-request-id: 57ABD896CCB80C366955187E
x-oss-server-time: 0
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <Code>NotImplemented</Code>
  <Message>A header you provided implies functionality that is not implemented. </Message>
  <RequestId>57ABD896CCB80C366955187E</RequestId>
  <HostId>bucketname.regionid.example.com</HostId>
  <Header>If-Modified-Since</Header>
</Error>

```

1.3. ASAPI Developer Guide

1.3.1. Preparations

1.3.1.1. Obtain API information

To call API operations by using an SDK, you must first obtain information about the service name, API operation name, and API version.

Context

The information on the API Directory page of the Apsara Uni-manager Management Console is for reference only. For more information about how to call API operations, see the developer guides of cloud services. If the information on the API Directory page is different from that in the developer guide of a cloud service, the information in the developer guide prevails.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Products** and choose **Others > API Tool**.
3. In the left-side navigation pane, click **API Directory**.
4. On the **API Directory** page, select the required cloud service from the Product Name drop-down list, enter a keyword of an operation name in the search box, and then click Search.
5. In the search result, you can view the **cloud service name**, **API operation name**, and **API version**.

1.3.1.2. Obtain an AccessKey pair

AccessKey pairs are classified into AccessKey pairs assigned by Resource Access Management (RAM) and those assigned by Security Token Service (STS). You can use either type of AccessKey pair for authentication when you call operations. This topic describes how to obtain an AccessKey pair that is assigned by RAM.

Context

An AccessKey pair is used to verify the identity of a request sender based on symmetric encryption. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string.

AccessKey credentials can be assigned by RAM or STS:

- Use RAM to assign an AccessKey ID and an AccessKey secret to a third party.
- Use STS to assign an AccessKey ID, an AccessKey secret, and a security token to a third party. For more information, see [Obtain an STS AccessKey pair](#).

The Apsara Uni-manager Management Console provides AccessKey pairs for personal accounts and organizations. Only operations administrators and the administrators of level-1 organizations can obtain the AccessKey pairs of organizations. We recommend that you use the AccessKey pairs of personal accounts to call the operations provided by the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, configure the request headers that are described in the following table for access control. Otherwise, an error that indicates insufficient permissions may occur.

 **Warning** The AccessKey pair of a personal account is managed and controlled by the permission system of the Apsara Uni-manager Management Console. If you need to obtain the AccessKey pair of an organization, make sure that your operation is allowed by an administrator. This ensures security.

Parameter	Description
x-acs-regionid	The region where the bucket resides. Example: cn-hangzhou-*
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain organization IDs, see the "GetOrganizationList" section of <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain resource set IDs, see the "ListResourceGroup" section of <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-instanceid	The name of the bucket that you want to operate. Example: examplebucket. You can find the OSS bucket name in the bucket list. For more information, see the "Obtain a bucket name" section of the Obtain the values of common header parameters topic.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, click the profile picture and then click **User Information**.
3. In the **AccessKey Pair** section, view the AccessKey pair of your account.

AccessKey Create AccessKey Pair

The AccessKey ID and AccessKey Secret are the keys used to access the cloud service resources. They have full permissions on the account. Please keep them properly.

AccessKey ID	AccessKey Secret	Status	Created At	Actions	
W1YRFvKH...			Enable	Sep 18, 2021, 11:05:57	Disable Delete
IsJsAUyX...			Enable	Sep 10, 2021, 12:28:35	Disable Delete

Note The AccessKey pair consists of an AccessKey ID and an AccessKey secret. These credentials provide you with full permissions on Apsara Stack resources. You must keep the AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization list, click the name of the level-1 organization whose AccessKey pair you want to obtain.
5. Click **Management Accesskey**.
6. In the dialog box that appears, view the AccessKey pair of the organization.

1.3.1.3. Obtain the values of common header parameters

You must provide multiple header parameters when you call API operations. This topic describes how to obtain the values of common header parameters. This topic also explains some response parameters.

Obtain a region ID

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Regions**.
4. Click the name of the organization whose region ID you want to view.
5. View the region ID in the **Regions** section.

Obtain an organization ID

You can call the `GetOrganizationList` operation to obtain the organization ID. For more information, see *GetOrganizationList* in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a resource set ID

You can call the `ListResourceGroup` operation to obtain the resource set ID. For more information, see *ListResourceGroup* in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a bucket name

You can obtain the name of an OSS bucket in the Apsara Uni-manager Management Console. A bucket name must be in the `x-acs-instanceid` format. To obtain the name of a bucket, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Storage > Object Storage Service**.
3. In the left-side navigation pane, click **Buckets**.
4. On the **Buckets** page, view the name in the **Bucket Name** column.

Response parameters

The following table describes the additional response parameters returned by default when you call operations to query the instances of cloud services.

Parameter	Description
ResourceGroupName	The name of the resource set.
ResourceGroup	The ID of the resource set.
DepartmentName	The name of the organization.

Parameter	Description
Department	The ID of the organization.

1.3.1.4. Obtain an STS AccessKey pair

This topic describes how to use Security Token Service (STS) to call API operations in Apsara Stack.

Background information

STS allows you to manage temporary credentials to your Apsara Stack resources. You can use STS to call API operations in Apsara Stack.

An STS AccessKey pair consists of an AccessKey ID, an AccessKey secret, and a security token. When you use STS to call API operations, you must use an STS AccessKey pair.

Procedure

1. Obtain the `RAM role`.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. Move the pointer over the profile picture in the upper-right corner of the page and click **Personal Information**.
 - iii. Click **View Current Role Policy**.

In the **View Policies of Current Role** dialog box, view the ARN of your `RAM role`.

2. Obtain the STS AccessKey pair.

Note

In the following code, replace the variables with actual values. Delete the angle brackets <> in the final code.

The following section describes the variables:

- <yourRegionID>: Replace this variable with the actual region ID.
- <yourAccessKeyID>: Replace this variable with your AccessKey ID.
- <yourAccessKeySecret>: Replace this variable with your AccessKey secret.
- <yourRoleSessionName>: Replace this variable with your current session name.
- <yourRoleArn>: Replace this variable with the ARN of your RAM role.
- "sts.aliyuns.com": Replace this variable with the endpoint of STS.

```

public static void main(String[] args) {
    // Create and initialize a DefaultAcsClient instance.
    // Specify the region ID, AccessKey ID, and AccessKey secret.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKeyID>", "<yourAccessKeySecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    String assumeRole = callAssumeRole(client);
    if (StringUtils.isEmpty(assumeRole)) return;
    JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
    JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
    // Obtain the STS AccessKey pair.
    String stsAK = credentials.getString("AccessKeyId");
    String stsSK = credentials.getString("AccessKeySecret");
    String stsToken = credentials.getString("SecurityToken");
}

private static String callAssumeRole(IAcsClient client) {
    CommonRequest request = new CommonRequest();
    // Replace sts.aliyuns.com with the endpoint of STS. The endpoint of STS is in the sts-vpc.${global:region}.${global:internet-domain} format
    request.setSysDomain("sts.aliyuns.com");
    request.setSysProduct("Sts");
    request.setSysAction("AssumeRole");
    request.setSysVersion("2015-04-01");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace <yourRoleSessionName> with the name of the current session.
    request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
    // Replace "<yourRoleArn>" with the actual ARN of the RAM role that you obtained in the preceding step.
    request.putQueryParameter("RoleArn", "<yourRoleArn>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        return response.getData();
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
    return null;
}

```

3. Use the STS AccessKey pair to call an operation of a service. The following sample code provides an example on how to use the STS AccessKey pair to call the `PutBucket` operation of Object Storage Service (OSS).

Note

In the following code, replace the variables with actual values. Delete the angle brackets <> in the final code.

```

import com.alibaba.fastjson.JSONObject;
import com.alibaba.fastjson.serializer.SerializerFeature;
import com.aliyun.asapi.ASClient;

import java.util.HashMap;
import java.util.Map;
import java.util.UUID;

public class asapitestdemo {
    public static void main(String[] args) {
        ASClient client = new ASClient();
        // Specify the identifier of the caller. The identifier is used only to record logs.
        client.setSdkSource("asapi@asapi-inc.com");

        // Prepare for initialization and specify the request parameters.
        Map<String, Object> parameters = new HashMap<String, Object>();
        parameters.put(ASClient.PRODUCT, "OneRouter");
        parameters.put(ASClient.ACTION, "DoOpenApi");
        parameters.put(ASClient.VERSION, "2018-12-12");
        String uuid = UUID.randomUUID().toString();
        // Enter your AccessKey ID and AccessKey secret.
        parameters.put(ASClient.ACCESSKEY_ID, "xxx");
        parameters.put(ASClient.ACCESSKEY_SECRET, "xxx");
        // When you use an STS AccessKey pair to call the rpc operation, you must add the SecurityToken field to the request parameters.
        parameters.put("SecurityToken", "<your-securitytoken>");
        // When you use an STS AccessKey pair to call the roa operation, you must add the x-acs-security-token field to the headers.
        // headers.put("x-acs-security-token", "<your-securitytoken>");
        // Enter your actual region ID.
        parameters.put(ASClient.REGIONID, "xxx");
        // Enter the service name.
        parameters.put("ProductName", "oss");
        // Enter the name of the API operation. It must be the same as the name of the API operation in API Reference.
        parameters.put("OpenApiAction", "PutBucket");
        // Enter the user account information.
        parameters.put("AccountInfo", uuid);

        JSONObject params = new JSONObject();
        // Enter the edition type of the service. Set this parameter to enterprise for the enterprise edition.
        params.put("asVersion", "enterprise");
        // Enter the architecture type of CPU.
        params.put("asArchitecture", "arm");
        params.put("SecurityToken", "<yourStsToken>");

        // Enter the parameters in API Reference. The parameters are case-sensitive.

        params.put("BucketName", "asapibucket");
        parameters.put("Params", params.toJSONString());

        Map<String, String> headers = new HashMap<String, String>();
        // Enter the ID of the organization in the Apsara Uni-manager Management Console.
        headers.put(ASClient.X_ACS_ORGANIZATION_ID, "3");
        // Enter the ID of the resource set in the Apsara Uni-manager Management Console.
        headers.put(ASClient.X_ACS_RESOURCEGROUP_ID, "<your-resourcegroupid>");

        // Enter the endpoint of Apsara Stack API (ASAPI).
        String endpoint = "https://asapi.xxx.xxx/asapi/v3";

        // Initiate a request.
        String result = client.doPost(endpoint, headers, parameters);

        // Display the results.
        System.out.println(JSONObject.toJSONString(JSONObject.parseObject(result), SerializerFeature.PrettyFormat));
    }
}

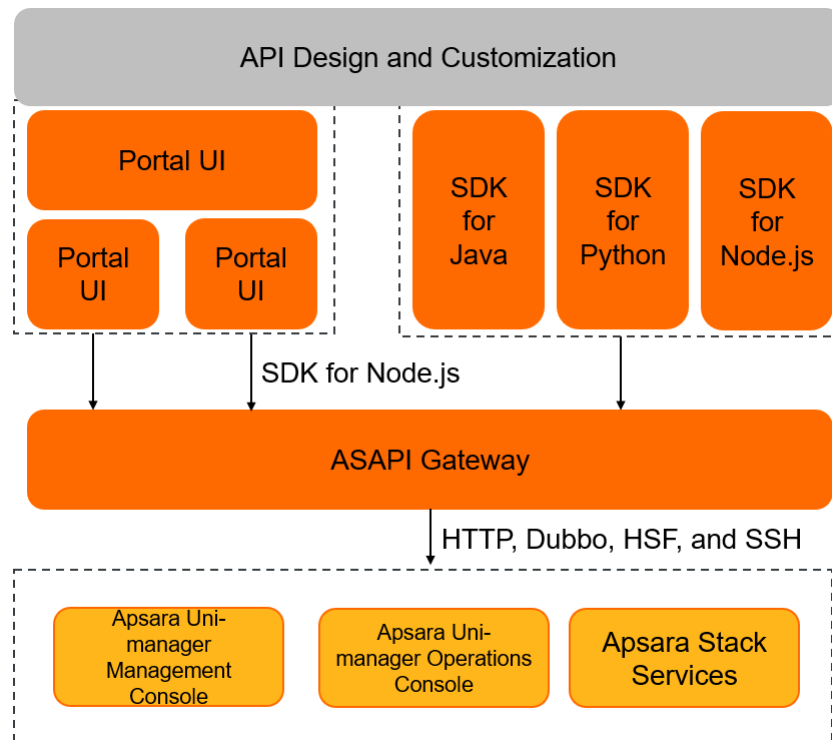
```

1.3.2. Use ASAPI to make API calls

1.3.2.1. ASAPI overview

This topic describes the architecture and features of the Apsara Stack API (ASAPI).

The ASAPI is a unified ingress gateway for Apsara Stack services and a main component of the Apsara Uni-manager Management Console. The ASAPI provides a unified method for you to manage and make API calls. You can use Apsara Stack SDK to centrally make API calls to Apsara Stack services, including Elastic Compute Service (ECS) and Virtual Private Cloud (VPC).



The ASAPI allows you to make API calls to manage and maintain Apsara Stack services.

- Make API calls to implement enterprise-grade management. For example, you can manage organizations, users, roles, resource sets, and logon policies. These operations are provided by the Apsara Uni-manager Management Console.
- Make API calls to manage operations. For example, you can manage quotas, usage statistics, specifications, and password policies. These operations are provided by the Apsara Uni-manager Management Console.
- Make API calls to manage resources. For example, you can create, delete, query, or modify cloud resources. These operations are provided by Apsara Stack services. If you call an API operation to create an ECS instance or a VPC, the API operation is provided by ECS or VPC.
- Make API calls to query monitoring metrics of Apsara Stack services. These operations are provided by the Apsara Uni-manager Management Console.
- Make API calls to perform O&M. For example, you can manage monitoring and alerting, inventories, tasks, and physical servers. These operations are provided by the Apsara Uni-manager Operations Console.

For more information, see the developer guide of the corresponding Apsara Stack service. For information about API operations used to manage organization users, see *Apsara Uni-manager Management Console Developer Guide*. For information about the API operation used to create an ECS instance, see *ECS Developer Guide*.

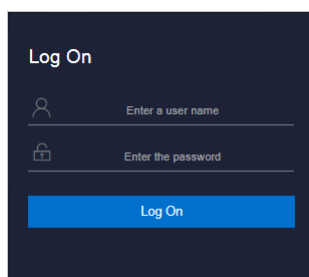
Note If the "Use Apsara Stack SDK to make API calls" section is included in the developer guide of an Apsara Stack service, the ASAPI is supported by the Apsara Stack service. In this case, you can make API calls based on the developer guide.

1.3.2.2. Obtain the endpoint of ASAPI

This topic describes how to obtain the endpoint of Apsara Stack API (ASAPI) in the Apsara Uni-manager Operations Console.

Procedure

1. In the address bar, enter the URL `region-id.aso.intranet-domain-id.com` and press the Enter key.



Note You can select a language from the drop-down list in the upper-right corner of the page.

2. Enter your username and password.

Note Obtain the username and password for logging on to the ASO console from the deployment personnel or an administrator.

When you log on to the ASO console for the first time, you must change the password of your username as prompted.

To enhance security, a password must meet the following requirements:

- It must contain uppercase and lowercase letters.
- It must contain digits.
- It must contain special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).
- It must be 10 to 20 characters in length.

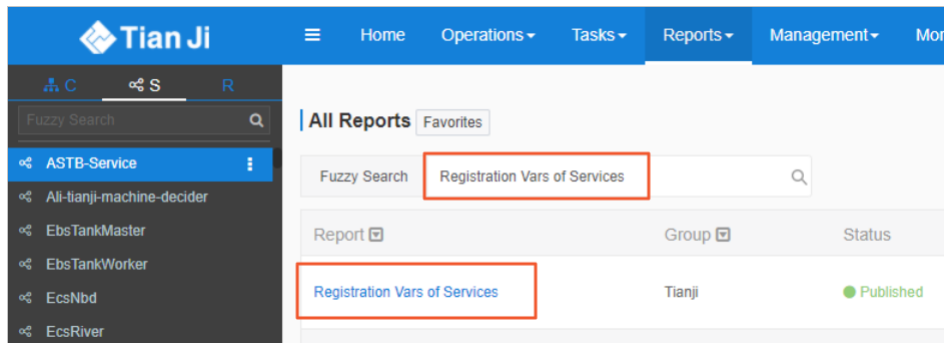
3. Click **Log On** to go to the ASO console.

4. In the left-

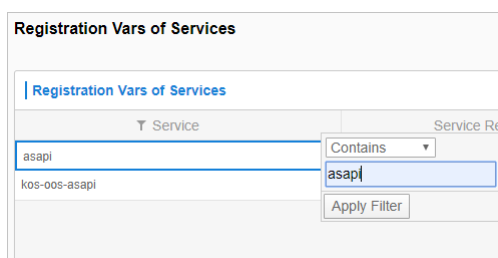
side navigation pane, choose **Product Management > Product List > Apsara Infrastructure Management Framework** to go to the Apsara Infrastructure Manage

5. In the left-side navigation pane, choose **Reports**.

6. On the **All Reports** page, search for **Registration Vars of Services**. Click **Registration Vars of Services**.



7. On the **Registration Vars of Services** page, click the  icon next to **Service**. In the dialog box that appears, search for **asapi**.



8. Right-click the **Service Registration** column that corresponds to **asapi** and select **Show More** from the shortcut menu.

9. In the **Details** dialog box, view the value of **asapi.gateway.endpoint**. The value is the endpoint of ASAPI.



1.3.2.3. Obtain Apsara Stack SDKs

Apsara Stack API (ASAPI) is a unified portal provided by Apsara Stack. You can use the ASAPI to connect to all Apsara Stack services. Apsara Stack provides SDKs for API calls.

You can go to [Apsara Stack documentation](#) and download the Apsara Stack SDKs in the **Other Resources** section at the lower part of the page.

1.3.2.4. Common parameters

Common request parameters must be included in all API requests.

Common request parameters

The following table describes the common request parameters that are used when you call API operations supported by Apsara Stack API (ASAPI).

Parameter	Type	Required	Description

Parameter	Type	Required	Description
Action	String	Yes	The operation mode. Set this parameter to DoOpenApi.
Product	String	Yes	The method to access the service. Set this parameter to OneRouter.
Version	String	Yes	The API version of the service registered on ASAPI. The current API version is 2018-12-12.
ProductName	String	Yes	The name of the service. Set this parameter to oss.
OpenApiAction	String	Yes	The name of the API operation. It must be the same as the name of the API operation in API Reference.
RegionId	String	Yes	The ID of the region. For more information, see Obtain a region ID .
AccessKeyId	String	Yes	The AccessKey ID that is used to access the service. For information about how to obtain an AccessKey ID, see Obtain an AccessKey pair .
Signature	String	Yes	The signature string. For more information about how signatures are calculated, see Request signatures.
SignatureMethod	String	Yes	The encryption algorithm of the signature string. Valid values: HMAC-SHA1 and HMAC-SHA256.
SignatureVersion	String	Yes	The version of the signature encryption algorithm. Valid values: 1.0 and 2.0.
SignatureNonce	String	Yes	A unique, random number used to prevent replay attacks. You must use different numbers for different requests.
Timestamp	String	Yes	The timestamp of the request. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC. Example: 2018-01-01T12:00:00Z.
Format	String	Yes	The format of the response parameters. Set the value to JSON.

Note

The `Signature`, `SignatureMethod`, `SignatureVersion`, `SignatureNonce`, `Timestamp`, and `Format` parameters are encapsulated into Apsara Stack SDK. You do not need to specify these parameters.

1.3.2.5. SDK description

1.3.2.5.1. ASAPI SDK for Java

This topic describes how to obtain and use ASAPI SDK for Java to call API operations.

Prerequisites

- An authorized account is created, and an AccessKey pair is obtained. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. You can obtain and view your AccessKey pair in the Apsara Uni-manager Management Console. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of ASAPI is obtained. For more information, see [Obtain the endpoint of ASAPI](#).
- OSS SDK for Java 1.8 or later is installed.

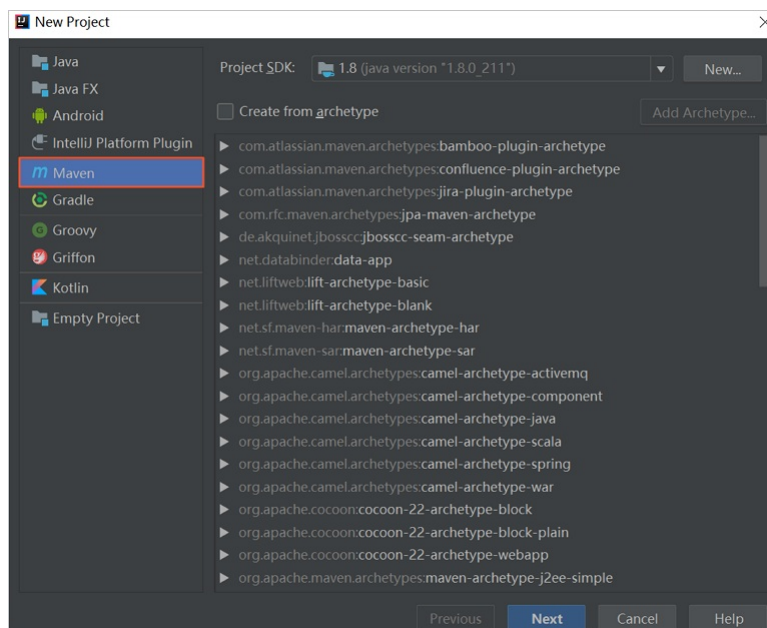
Background information

The help center of Apsara Stack provides Maven dependencies and JAR files for Apsara Stack SDK for Java. You can write code to use Apsara Stack SDK for Java to call API operations.

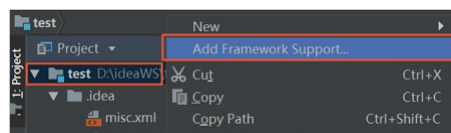
In this example, the Windows 10 64-bit OS and IntelliJ IDEA are used.

Procedure

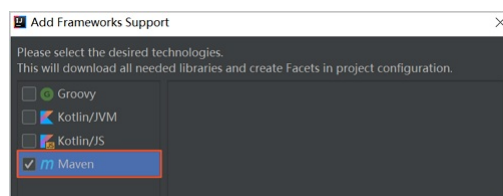
- Use one of the following methods to configure Maven in IntelliJ IDEA:
 - Use Maven that is integrated in IntelliJ IDEA.
 - Download Maven based on your OS from the official website and configure Maven. For more information, see [Downloading Apache Maven 3.8.6](#).
- Obtain the Maven dependencies for Apsara Stack SDK for Java from the official website. For more information about how to obtain the dependencies, see [Obtain the core package for ASAPI](#).
- Use one of the following methods to create a Maven project:
 - Method 1: Create a Maven project in IntelliJ IDEA.



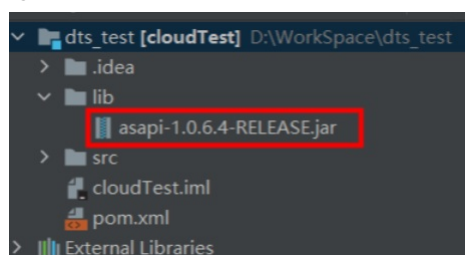
- Method 2: Convert an existing project to a Maven project.
 - Right-click the project that you want to convert and select **Add Framework Support**



- Select **Maven** and click **OK**.



- Create a folder named **libs** in the root directory of the project. Then, add the JAR file of Apsara Stack SDK for Java to the **libs** folder, as shown in the following figure.



- Add the dependencies of Apsara Stack SDK for Java to the **pom.xml** file in the project directory.

```

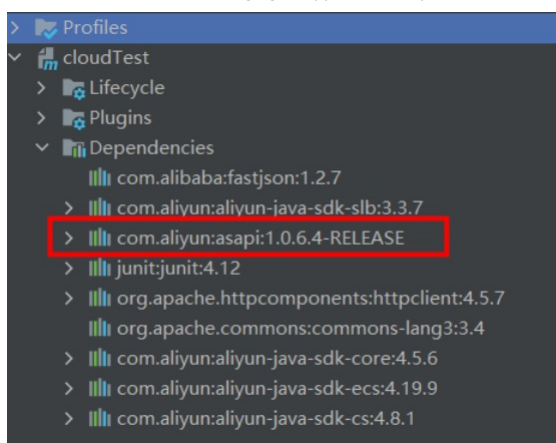
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>asapi</artifactId>
    <version>1.0.6.4-RELEASE</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/libs/asapi-1.0.6.4-RELEASE.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>alibaba-crypto</artifactId>
    <version>1.0.4</version>
    <systemPath>${project.basedir}/libs/alibaba-crypto-1.0.4.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>gmsse</artifactId>
    <version>1.1.0</version>
    <systemPath>${project.basedir}/libs/gmsse-1.1.0.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.7</version>
    <systemPath>${project.basedir}/libs/httpclient-4.5.7.jar</systemPath>
  </dependency>
</dependencies>

```

 **Note** Replace `1.0.6.4-RELEASE` with the actual version of Apsara Stack SDK for Java.

6. On the right side of IntelliJ IDEA, view the information in the Maven section.

If the information in the following figure appears, the dependencies are added.



Use OSS SDK for Java

Currently, you can use ASAPI to call OSS APIs by using OneRouter. The following sample code provides an example on how to call OSS API operations:

```
// 1. Create a connection to an ASClient object.
ASClient client = new ASClient();
// 2. Configure the identifier of the requester. This parameter is required and used only for identification. You can set this parameter to a random
value.
client.setSdkSource("<The identifier of the requester>");
Map<String, Object> parameters = new HashMap<String, Object>();
// 3. Configure the parameters for authorization. The parameter values vary based on your environment.
parameters.put(ASClient.PRODUCT, "OneRouter");
parameters.put(ASClient.ACTION, "DoOpenApi");
parameters.put(ASClient.VERSION, "2018-12-12");
String uuid = UUID.randomUUID().toString();
// Specify the AccessKey ID of the authorized account.
parameters.put("AccessKeyId", "<AccessKey ID>");
// Specify the AccessKey secret of the authorized account.
parameters.put("AccessKeySecret", "<AccessKey Secret>");
// Specify the region ID.
parameters.put("RegionId", "<your RegionID>");
// 4. Specify the information about the API operation that you want to call.
// Specify the name of the service.
parameters.put("ProductName", "oss");
// Specify the name of the API operation. It must be the same as the name of the API operation in API Reference.
parameters.put("OpenApiAction", "PutBucket");
// Specify the user account information.
parameters.put("AccountInfo", uuid);
// When you use an STS AccessKey pair to call the rpc operation, you must add the SecurityToken field to the request parameters.
// parameters.put("SecurityToken", "<your-securitytoken>");
// When you use an STS AccessKey pair to call the roa operation, you must add the x-acis-security-token field to the headers.
// headers.put("x-acis-security-token", "<your-securitytoken>");
// 5. Configure operation-specific parameters.
JSONObject params = new JSONObject();
parameters.put("<The name of the operation-specific parameter>", "<The value of the operation-specific parameter>");
parameters.put("Params", params.toJSONString());
// 6. Configure request headers and the encryption algorithm for the signature string. Valid values of the SignatureMethod parameter: HMAC-SHA1 and H
MAC-SHA256. By default, HMAC-SHA256 is used.
Map<String, String> headers = new HashMap<String, String>();
//headers.put("SignatureMethod", "HMAC-SHA1");
headers.put("x-acis-regionid", "<your-regionid>");
headers.put("x-acis-resourcegroupid", "<your-resourcegroupid>");
headers.put("x-acis-organizationid", "<your-organizationid>");
//headers.put("x-acis-instanceid", "<your-instanceid>");
// 7. Call the API operation. We recommend that you use the doPost() method.
// Configure the connection timeout period.
//client.setConnectTimeout(10000);
// Configure the request timeout period.
//client.setSocketTimeout(10000);
// Enter the endpoint of ASAPI.
String endpoint = "https://public.asapi.xxx.xxx/asapi/v3";
// Initiate a request.
String result = client.doPost(endpoint, headers, parameters);
// Display the results.
System.out.println(result);
```

Configure a proxy

If you need to use a proxy to call API operations, use the following sample code to configure the proxy:

```
Cloud cloud = new Cloud();
// Configure the IP address.
cloud.setConfig("proxy_host", "<IP>");
// Configure the port.
cloud.setConfig("proxy_port", "<Port>");
// Create a connection to an ASClient object.
ASClient client = new ASClient(cloud);
```

Headers

Header	Description
x-acis-regionid	The ID of the region. For more information, see Obtain a region ID .
x-acis-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information, see Query an organization ID .
x-acis-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information, see Query a resource set ID .
x-acis-instanceid	The name of the bucket on which the operation is performed. For more information, see Query a bucket name .

1.3.2.5.2. ASAPI SDK for Python

This topic describes how to obtain and use Apsara Stack SDK for Python to call API operations.

Prerequisites

- An authorized account is created, and an AccessKey pair is obtained. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. You can obtain and view your manager Management Console. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of the Apsara Stack API (ASAPI) is obtained. For more information, see [Generate an endpoint for ASAPI](#).

Install Apsara Stack SDK for Python

1. Download the required version of Apsara Stack SDK for Python. For more information, see [Obtain Apsara Stack SDKs](#).
2. Upload the compressed file of Apsara Stack SDK for Python to the host from which you want to send API requests and extract the file. Then, run the following code

```
# Install Apsara Stack SDK for Python 2.x.
pip install aliyun-python-sdk-asapi-2.4.7-py2.tar.gz
# Install Apsara Stack SDK for Python 3.x.
pip3 install aliyun-python-sdk-asapi-2.4.7-py3.tar.gz
```

Note

Replace `2.4.7` with the actual version of Apsara Stack SDK for Python.

Call examples

You can use the GET or POST method to call API operations provided by ASAPI.

```
# 1. Add dependencies.
from aliyunsdkasapi.ASClient import ASClient
from aliyunsdkasapi.AsapiRequest import AsapiRequest

if __name__ == '__main__':

    #2. Create a request.
    # Set the access mode to OneRouter, the ASAPI version to 2018-12-12, and the operation mode to DoOpenApi, and enter the ASAPI endpoint based on your business requirements.
    req = AsapiRequest("OneRouter", "2018-12-12", "DoOpenApi", "<asapi-endpoint>")

    #3. Specify business information.
    # Set the service name to oss.
    req.add_body_params("ProductName", "oss")
    # Specify the name of the API operation. It must be the same as the name of the API operation in API Reference.
    req.add_body_params("OpenApiAction", "PutBucket")
    req.add_body_params("AccountInfo", "xxx")
    # Specify the API-related parameters. In this example, set BucketName to examplebucket. Modify the parameters based on your business requirements.
    req.add_body_params("Params", "{\"BucketName\":\"examplebucket\"}")

    # When you use STS AccessKey credentials to call the rpc operation, you must add the SecurityToken field to the request parameters.
    # req.add_body_params("SecurityToken", "<your-securitytoken>");
    # When you use STS AccessKey credentials to call the roa operation, you must add the x-acs-security-token field to the headers.
    # req.add_header("x-acs-security-token", "<your-securitytoken>");

    #4. Specify the header information.
    req.add_header("x-acs-resourcegroupid", "<your-resourcegroupid>");
    req.add_header("x-acs-organizationid", "<your-organizationid>");
    req.add_header("x-acs-regionid", "<your-regionid>");
    req.add_header("x-acs-instanceid", "<your-instanceid>");

    #5. Specify the request mode.
    req.set_method("POST")

    # 6. Use the environment information to initialize an ASClient object.
    # The timeout parameter specifies the timeout period for a request. The cert_file parameter specifies the certificate. The verify parameter specifies whether to verify the certificate.
    as_client = ASClient("<accesskeyid>", "<accesskeysecret>", "<regionid>", timeout=1000,
                        cert_file=None,
                        verify=False)
    # Specify the identifier of the requester. This parameter is required and used only for identification. You can set this parameter to a random value.
    as_client.setSdkSource("<The identifier of the requester>")
    response = as_client.do_request(req)
    print(response)
```

Headers

Header	Description
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information, see Query a resource set ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information, see Query an organization ID .

x-acs-regionid	The ID of the region. For more information about how to obtain a region ID, see Obtain a region ID .
x-acs-instanceid	The name of the bucket on which the operation is performed. For more information, see Query a bucket name .

Configure a proxy

Perform the following steps to configure a proxy:

1. Import OS dependencies.

```
import os
```

2. Configure one or more proxies.

```
# Before you call an API operation, configure one or more proxies.
os.environ['http_proxy'] = "http://x.x.x.x:xxxx"
os.environ['https_proxy'] = "https://x.x.x.x:xxxx"
```

1.3.2.5.3. ASAPI SDK for Node.js

This topic describes how to obtain and use ASAPI SDK for Node.js to call API operations.

Prerequisites

- An authorized account is created, and an AccessKey pair is obtained. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. You can obtain and view your AccessKey pair in the Apsara Uni-manager Management Console. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of ASAPI is generated. For more information, see [Generate an endpoint for ASAPI](#).
- tnpm is installed. If tnpm is not installed, run the following command to install tnpm:

```
npm i -g npminstall --registry=http://registry.npm.alibaba-inc.com
npm install -g tnpm --registry=http://registry.npm.alibaba-inc.com
```

Install ASAPI SDK for Node.js

Go to the directory in which you want to install ASAPI SDK for Node.js and run the following command to install ASAPI SDK for Node.js:

```
tnpm i @ali/asapi-nodejs-sdk@0.0.33 --save
```

Use ASAPI SDK for Node.js

The following code provides an example on how to use ASAPI SDK for Node.js to call API operations:

```
// 1. Import the dependencies of ASAPI SDK for Node.js.
const Controller = require('egg').Controller;
const ASAPIClient = require('@ali/asapi-nodejs-sdk');
// 2. Configure the ASAPI client.
const config = {
  API_GATEWAY: '<The endpoint of ASAPI>',
  regionId: '<The region in which the Apsara Stack service is located>',
  accessKeyId: '<The AccessKey ID of the authorized Apsara Stack tenant account>',
  accessKeySecret: '<The AccessKey secret of the authorized Apsara Stack tenant account>'
};
// 3. Initiate the ASAPI client.
const AsapiClient = new ASAPIClient(config, this.ctx);
// 4. Specify the request parameters.
const params = {
  // Set the access mode to OneRouter, the ASAPI version to 2018-12-12, and the operation mode to DoOpenApi, and enter the ID of the region in which the Apsara Stack cloud service is located based on your business requirements.
  "Action": "DoOpenAPI",
  "Product": "OneRouter",
  "Version": "2018-12-12",
  regionId: '<The region in which the Apsara Stack service is located>',
  // When you use an STS AccessKey pair to call the rpc operation, you must specify the SecurityToken parameter.
  // "SecurityToken": "<your-securitytoken>",
  // Specify business parameters in the following format: "<Business parameters required by the parameter>": "<Parameter value>".
  // Set the Apsara Stack service name to oss.
  "ProductName": "oss",
  // Specify the name of the API operation. It must be the same as the name of the API operation in API Reference. Example: PutBucket.
  "OpenApiAction": "PutBucket",
  // Specify the API-related parameters. In this example, BucketName is set to asapibucket. Specify the parameters based on your business requirements.
  "Params": "{ \"BucketName\": \"asapibucket\" }"
}
// 5. Add headers.
// When you use an STS AccessKey pair to call the roa operation, you must specify the x-acis-security-token header.
//AsapiClient.addHeader("x-acis-security-token", "<your-securitytoken>")
AsapiClient.addHeader('x-acis-caller-sdk-source', '<The identifier of the requester>')
AsapiClient.addHeader('x-acis-roleid', '<The ID of the role>')
AsapiClient.addHeader('x-acis-userid', '<The ID of the user>')
AsapiClient.addHeader('x-acis-organizationid', '<your-organizationid>')
AsapiClient.addHeader('x-acis-regionid', '<your-regionid>');
AsapiClient.addHeader("x-acis-resourcegroupid", "<your-resourcegroupid>");
AsapiClient.addHeader("x-acis-instanceid", "<your-instanceid>");
// 6. Initiate a call and obtain the response.
try {
  const res = await AsapiClient.doRequest(config.API_GATEWAY, params, {
    method: 'POST', // Specify the request method.
    formatParams: false, // Specify whether to format parameters and capitalize the first letter of each parameter. If this parameter is set to false, parameters are not formatted.
    enableProxy: true, // Specify whether to configure a proxy. If this parameter is set to true, a proxy is configured and you must specify the proxy parameters.
    proxy: 'http://x.x.x.x:xxxx',
    timeout: 10000, // Specify the timeout period.
    // ca: , // Specify the CA certificate. For more information, see https://nodejs.org/api/https.html#https_https_request_options_callback.
  })
  console.log(res)
} catch (e) {
  console.log(e)
}
```

Disable certificate validation

The following code provides an example on how to disable certificate validation:

```
// Specify environment variables.
NODE_TLS_REJECT_UNAUTHORIZED=0
```

Headers

Header	Description
x-acis-regionid	The ID of the region. For more information, see Obtain a region ID .
x-acis-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information, see Obtain an organization ID .
x-acis-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information, see Obtain a resource set ID .
x-acis-instanceid	The name of the bucket on which the operation is performed. For more information, see Obtain a bucket name .
x-acis-caller-sdk-source	The custom name of the requester or the system. This parameter is used to identify the source of the API request.
x-acis-roleid	The ID of the role.

Header	Description
x-acs-userid	The ID of the user.

1.3.3. HTTP requests

1.3.3.1. Overview

This topic describes how to call Apsara Stack APIs registered with ASAPI by sending HTTP requests.

Note

We recommend that you use SDKs to call API operations instead of using HTTP requests. Currently, ASAPI provides SDKs for Node.js, Python, and Java, which save you from the signing process. If you use an SDK to initiate requests, you can skip this topic.

The request URL consists of different parameters but has a fixed request structure. The URL contains common request parameters, your signature, and operation-specific parameters. Sample request URLs are provided for each API operation. These URLs are not encoded, which makes them easy to read. Before you initiate a request, you must encode the request URL. After the request is verified, the results are returned. Response parameters are returned if the call is successful, whereas an error message is returned if the call fails. You can troubleshoot issues based on the error message.

1.3.3.2. Request syntax

ASAPI allows you to initiate HTTP GET requests based on URLs. Each URL must contain request parameters. This topic describes the syntax of HTTP GET requests and provides the endpoint of ASAPI.

Sample request

The following example shows an unencoded URL that is used to call the PutBucket operation:

```
https://public.asapi.aliyuns.com/asapi/v3?Action=DoOpenApi&Product=OneRouter&Version=2018-12-12&ProductName=oss&OpenApiAction=PutBucket&Id=<Organization ID>&<Common request parameters>
```

- `https` specifies the protocol that is used to send the request.
- `public.asapi.aliyuns.com/asapi/v3` specifies the endpoint of ASAPI.
- `Action=DoOpenApi&Product=OneRouter&Version=2018-12-12&ProductName=oss&OpenApiAction=PutBucket` specifies information about the API operation that you want to call. `OpenApiAction` specifies the name of the OSS API operation.
- `Id` specifies the ID of the organization.
- `<Common request parameters>` specifies the common parameters specified in the system.

Protocols

ASAPI supports only HTTPS.

Endpoints

For more information about how to generate the endpoint of ASAPI, see [Generate an endpoint for ASAPI](#).

You must specify parameters, such as the Action, Product, Version, ProductName, and OpenApiAction parameters, to specify the API operation that you want to call, such as `Action=DoOpenApi&Product=OneRouter&Version=2018-12-12&ProductName=oss&OpenApiAction=PutBucket`. You must also specify other operation-specific request parameters and common request parameters. For more information, see [Common request parameters](#).

1.3.3.3. Request signatures

This topic describes how to sign HTTP API requests. Alibaba Cloud uses the request signature to verify the identity of the requester. Specifically, Alibaba Cloud uses an AccessKey pair to perform symmetric encryption and verify the identity of the requester.

Note

- An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is issued to each user or organization in the Apsara Uni-manager Management Console.
- The AccessKey ID is used to verify the identities of users, and the AccessKey secret is used to encrypt and verify signature strings. You must store the AccessKey secret in a secure location.
- ASAPI provides SDKs for Node.js, Python, and Java to facilitate complex signature calculation. For more information, see [Obtain Apsara Stack SDKs](#).

Step 1: Construct a canonicalized query string

- Sort the request parameters in alphabetical order, including all common and operation-specific parameters except for `Signature`.

Note

If you use the GET method to send a request, the request parameters are included as part of the request URL. The parameters follow the question mark (?) and are connected by ampersands (&) in the URL.

2. Encode the names and values of the request parameters in UTF-8 based on RFC 3986.

- Letters, digits, and special characters such as hyphens (-), underscores (_), and periods (.) do not need to be encoded.
- Other characters must be percent-encoded in the %XY format. XY represents the ASCII code of the characters in hexadecimal notation. For example, double quotation marks (") are encoded as %22 .
- Extended UTF-8 characters are encoded in the %XY%ZA... format.
- Spaces must be encoded as %20 . Do not encode spaces as plus signs (+).

The preceding encoding method is similar to but slightly different from the application/x-www-form-urlencoded MIME encoding algorithm.

If you use java.net.URLEncoder in the Java standard library, use percentEncode to encode request parameters and their values. In the encoded query string, replace plus signs (+) with %20 , asterisks (*) with %2A , and %7E with tildes (~). This way, you can obtain an encoded string that matches the preceding encoding rules.

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedEncodingException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("***", "%2A").replace("%7E", "~") : null;
}
```

3. Use equal signs (=) to connect the name and value of an encoded parameter.

4. Use ampersands (&) to connect the encoded request parameters. These parameters must be arranged in the same order as those in Step 1.

Then, a canonicalized query string that follows the request structure is obtained. For more information, see [Request structure](#).

Step 2: Create a string-to-sign

- Create a string-to-sign (StringToSign). You can also use percentEncode to encode the canonicalized query string that is generated in the Step 1. When you create a string-to-sign, take note of the following rules:

```
StringToSign=
    HTTPMethod + "&" + //HTTPMethod: The HTTP method that is used to send the request, such as GET.
    percentEncode("/") + "&" + //percentEncode("/"): Encode backslashes (/) as %2F by using UTF-8.
    percentEncode(CanonicalizedQueryString) // Encode the canonicalized query string that is created in Step 1.
```

- Calculate the HMAC-SHA1 or HMAC-256 value of StringToSign based on RFC 2104. In this example, the Java Base64 encoding method is used.

```
String signature = Base64.getEncoder().encodeToString(signData);
```

Note

When you calculate the signature, the key value that is specified by RFC 2104 is your AccessKey secret plus an ampersand (&), which has an ASCII value of 38. For more information, see [Obtain an AccessKey pair](#).

- Encode the Signature parameter based on RFC 3986 and add the parameter to the canonicalized query string.

Example 1: Concatenate parameters

The following example shows how to call the PutBucket operation to create a bucket. In this example, the value of AccessKeyID is testid and the value of AccessKeySecret is testsecret .

- Construct a canonicalized query string.

```
https://public.asapi.aliyuns.com/asapi/v3?AccessKeyId=testid&Action=DoOpenApi&BucketName=examplebucket&&Format=JSON&Id=3&OpenApiAction=PutBucket&Product=OneRouter&ProductName=oss&RegionId=cn-neimeng-env141-d01&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2022-01-11T08%3A50%3A40Z&Version=2018-12-12
```

- Create a string-to-sign (StringToSign).

```
POST%2F&AccessKeyId%3Dtestid%26Action%3DDoOpenApi%26BucketName%3Dexamplebucket%26%26Format%3DJJSON%26Id%3D3%26OpenApiAction%3DPutBucket%26Product%3DOneRouter%26ProductName%3Doss%26RegionId%3Dcn-neimeng-env141-d01%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D43967c6e-e3ad-44fa-b06b-fcf3db5f39d2%26SignatureVersion%3D1.0%26Timestamp%3D2022-01-11T08%25A50%25A40Z%26Version%3D2018-12-12
```

- Calculate the signature.

Calculate the signature. As AccessKeySecret is set to testsecret , testsecret& is used as the key value to calculate the signature. The calculated signature is OLeaidSlJvxuMvnyH0wuJ+uX5qY= . In this example, the Java Base64 encoding method is used.

```
String signature = Base64.getEncoder().encodeToString(signData); Encode Signature=KankeY6w8gp/yWQfoeaD+LZZ25Q= based on RFC 3986 and add it to the canonicalized query string.
```

```
https://public.asapi.aliyuns.com/asapi/v3?AccessKeyId=testid&Action=DoOpenApi&BucketName=examplebucket&Format=JSON&Id=3&OpenApiAction=PutBucket&Product=OneRouter&ProductName=oss&RegionId=cn-neimeng-env141-d01&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2022-01-11T08%3A50%3A40Z&Version=2018-12-12&Signature=KankeY6w8gp/yWQfoeaD+LZZ25Q=
```

You can use the new URL in a browser, in cURL, or in Wget to initiate an HTTP request that calls the PutBucket operation to create a bucket named examplebucket.

Example 2: Use the standard library of the programming language

The following example shows how to call the `PutBucket` operation to create a bucket. In this example, `AccessKeyId` is set to `testid`, `AccessKeySecret` is set to `testsecret`, and all request parameters are stored in a Java `Map<String, String>` object.

1. Predefine the encoding method.

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedEncodingException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("'", "%2A").replace("%7E", "-") : null;
}
```

2. Predefine the time format for the `Timestamp` parameter. Specify the `Timestamp` parameter in the ISO 8601 format. The time must be in UTC+0.

```
private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static String formatIso8601Date(Date date) {
    SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
    df.setTimeZone(new SimpleTimeZone(0, "GMT"));
    return df.format(date);
}
```

3. Create a query string.

```
final String HTTP_METHOD = "POST";
Map parameters = new HashMap();
// Specify request parameters.
parameters.put("Product", "OneRouter");
parameters.put("Action", "DoOpenApi");
parameters.put("Version", "2018-12-12");
parameters.put("AccessKeyId", "testid");
parameters.put("RegionId", "cn-region-test");
parameters.put("Timestamp", formatIso8601Date(new Date()));
parameters.put("SignatureMethod", "HMAC-SHA1");
parameters.put("SignatureVersion", "1.0");
parameters.put("SignatureNonce", UUID.randomUUID().toString());
parameters.put("Format", "JSON");
parameters.put("Id", "1");
JSONObject params = new JSONObject();
// Specify the following parameters for this API call. The parameters are case-sensitive.

params.put("BucketName", "asapibucket");
parameters.put("Params", params.toJSONString());
// Sort the request parameters.
String[] sortedKeys = parameters.keySet().toArray(new String[]{});
Arrays.sort(sortedKeys);
final String SEPARATOR = "&";
// Create a string-to-sign.
StringBuilder stringToSign = new StringBuilder();
stringToSign.append(HTTP_METHOD).append(SEPARATOR);
stringToSign.append(percentEncode("/")).append(SEPARATOR);
StringBuilder canonicalizedQueryString = new StringBuilder();
for(String key : sortedKeys) {
    // Encode the keys and values of the parameters.
    canonicalizedQueryString.append("&")
        .append(percentEncode(key)).append("=")
        .append(percentEncode(parameters.get(key)));
}
// Encode the canonicalized query string.
stringToSign.append(percentEncode(
    canonicalizedQueryString.toString().substring(1)));
```

4. Calculate the signature. As `AccessKeySecret` is set to `testsecret`, `testsecret&` is used as the key value to calculate the signature. The calculated signature is

```
OLeaidSlJvxuMvnyH0wuJ%2BuX5qY%3D .
```

```
// The following code provides an example on how to calculate the signature.
final String ALGORITHM = "HmacSHA1";
final String ENCODING = "UTF-8";
key = "testsecret&";
Mac mac = Mac.getInstance(ALGORITHM);
mac.init(new SecretKeySpec(key.getBytes(ENCODING), ALGORITHM));
byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));
String signature = Base64.getEncoder().encodeToString(signData);
```

Encode the Signature parameter based on [RFC 3986](#) and add the parameter to the URL. Then, you can obtain the following URL:

5. Use a browser or a tool such as `cURL` or `Wget` to send the HTTP request.

1.3.3.4. Responses

Responses are returned only in the JSON format. If a return code is greater than or equal to 200 and less than 300, a success response is returned. Otherwise, an error response is returned.

Sample success responses

If a request is successful, the request ID and operation-specific parameter values are returned. For more information, see the developer guide of the corresponding Apsara Stack service.

```
{
  "successResponse": true,
  "eagleEyeTraceId": "xxxxxxxxxxxxxx",
  "asapiSuccess": true,
  "code": "200",
  "cost": 12,
  "asapiRequestId": "xxxxxxxxxxxxxx",
  "data": [
    {
      "muserId": "aliyuntest",
      "internal": false,
      "level": "0",
      "supportRegions": "cn-*",
      "personNum": 100,
      "mtime": 1553247962000,
      "uuid": "1",
      "parentId": 0,
      "supportRegionList": [
        "cn-*"
      ],
      "name": "root",
      "alias": "root",
      "ctime": 1553247962000,
      "id": 1,
      "resourceGroupNum": 100,
      "cuserId": "aliyuntest"
    }
  ],
  "requestId": "xxxxxxxxxxxxxx",
  "success": true,
  "message": "success"
}
```

Sample error responses

If a request fails, the following information is returned. For more information, see the developer guide of the corresponding Apsara Stack service.

```
{
  "code": "asapi.server.api.notfound", // The error code.
  "eagleEyeTraceId": "xxx-xxx-xxx-xxx", // The trace ID provided by EagleEye.
  "asapiSuccess": false, // Indicates whether the request is successful.
  "hostId": "xxxx", // The ID of the host.
  "message": "Failed to ...", // The message returned.
  "asapiErrorMessage": "Failed to ...", // The error message provided by ASAPI.
  "asapiErrorCode": "asapi.server.api.notfound" // The error code provided by ASAPI.
}
```

1.4. SDK Reference

1.4.1. Java SDK

1.4.1.1. Preface

This topic is written for OSS SDK for Java version 3.8.0.

Compatibility

- For SDKs version 3. x. x:
 - API operation: compatible
 - Namespace: compatible
- For SDKs version 2. x. x:
 - API operation: compatible
 - Namespace: compatible
- For SDKs version 1.0. x:
 - API operation: compatible
 - Namespace: incompatible. The Tablestore code in version 1.0.x is removed from version 2.0.0. Package names `com.aliyun.openservices. *` and `com.aliyun.openservices.oss. *` are replaced with `com.aliyun.oss. *`.

SDK source code and API documents

For the SDK source code, visit [GitHub](#). For more information, visit [Aliyun OSS SDK for Java V3.8.0 API](#).

Sample code

OSS SDK for Java provides a variety of sample code for you to reference or use. The following table describes the sample code.

Sample file	Description
GetStartedSample.java	Shows you how to perform basic upload and download operations.
SimpleGetObjectSample.java	Shows you how to download objects.
ListObjectsSample.java	Shows you how to list objects.
DeleteObjectsSample.java	Shows you how to delete multiple objects at a time.
AppendObjectSample.java	Shows you how to use append upload.
ObjectMetaSample.java	Shows you how to use object metadata.
CreateFolderSample.java	Shows you how to create a folder. For more information about OSS folders, see the topic about folders.
UploadSample.java	Shows you how to use resumable upload.
DownloadSample.java	Shows you how to use resumable download.
ImageSample.java	Shows you how to use IMG.
PostObjectSample.java	Shows you how to use PostObject. The implementation is independent of OSS SDK for Java.
GetProgressSample.java	Shows you how to use progress bars in object upload and download.
CallbackSample.java	Shows you how to perform upload callback.
CRCSample.java	Shows you how to perform CRC for object upload and download.
BucketOperationsSample.java	Shows you how to configure a bucket, including configuring ACL, lifecycle, logging, hotlink protection, and CORS.
MultipartUploadSample.java	We recommend that you use the uploadFile operation to perform resumable upload for concurrent uploads implemented by the multipart upload API operation.
ConcurrentGetObjectSample.java	We recommend that you use the downloadFile operation to perform resumable download for concurrent downloads implemented by range download.
UploadPartCopySample.java	Shows you how to perform multipart copy for large objects.

1.4.1.2. Installation

This topic describes how to install OSS SDK for Java.

Prerequisites

- **Environment**
Use Java 1.7 or later.
- **Compatibility**
Run the `java -version` command to check the Java version.

Download the SDK

- [Download directly](#)
- [Download from GitHub](#)
- [Download previous versions](#)

Install the SDK

- **Method 1: (Recommended) Add dependencies to your Maven project**

To use OSS SDK for Java in Maven, you need only to add the corresponding dependencies to the pom.xml file. For example, when you use OSS SDK for Java version 3.8.0, add the following content to <dependencies>:

```
<dependency>
  <groupId>com.aliyun.oss</groupId>
  <artifactId>aliyun-sdk-oss</artifactId>
  <version>3.8.0</version>
</dependency>
```

- **Method 2: Import the JAR package to your Eclipse project**

For example, when you use OSS SDK for Java version 3.8.0, take the following steps to install the SDK:

- Download the [OSS SDK for Java package](#).
- Decompress the package.
- Copy the `aliyun-sdk-oss-3.8.0.jar` file and all files in the lib directory to your project.
- Right-click your project name in Eclipse. Choose **Properties > Java Build Path > Add JARs**.
- Select all JAR files that you have copied in Step 3 and import them to Libraries.

- **Method 3: Import the JAR package to your IntelliJ IDEA project**

For example, when you use OSS SDK for Java version 3.8.0, take the following steps to install the SDK:

- Download the [OSS SDK for Java package](#).

- ii. Decompress the package.
- iii. Copy the *aliyun-sdk-oss-3.8.0.jar* file and all JAR files in the lib directory to your project.
- iv. In IntelliJ IDEA, select your project and choose **File > Project Structure > Modules > Dependencies > + > JARs or directories**.
- v. Select all JAR files that you have copied in Step 3 and import them to External Libraries.

1.4.1.3. Initialization

OSSClient serves as the Java client to manage OSS resources such as buckets and objects. To use OSS SDK for Java to initiate a request, you must initialize an OSSClient instance and modify the default configuration items of ClientConfiguration.

Create an OSSClient instance

To create an OSSClient instance, you must specify an endpoint. For more information about endpoints, see [Obtain an endpoint](#).

To create an OSSClient instance, you must specify an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

- Create an OSSClient instance based on an OSS endpoint

The following code provides an example on how to create an OSSClient instance based on an OSS endpoint:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

You can use one or more OSSClient instances for your project. In other words, you can use multiple OSSClient instances simultaneously.

- Create an OSSClient instance based on Apsara Stack or a private region

The following code provides an example on how to create an OSSClient instance based on Apsara Stack or a private region:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create a ClientConfiguration instance and modify the default parameters based on your requirements.
ClientBuilderConfiguration conf = new ClientBuilderConfiguration();
// Disable CNAME.
conf.setSupportCname(false);
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, conf);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Create an OSSClient instance based on an IP address

The following code provides an example on how to create an OSSClient instance based on an IP address:

```
// In some special cases such as a private region, the IP address is used as the endpoint. In such cases, specify the actual IP address based on your requirements.
String endpoint = "http://10.10.10.10";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create a ClientConfiguration instance.
ClientBuilderConfiguration conf = new ClientBuilderConfiguration();
// By default, access from a second-level domain is disabled. Enable OSS access from a second-level domain. You must configure this parameter for OSS SDK for Java version 2.1.2 or earlier. The versions later than OSS SDK for Java version 2.1.2 automatically detect IP addresses.
conf.setSLDEnabled(true);
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, conf);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Create an OSSClient instance based on STS

The following code provides an example on how to create an OSSClient instance based on Security Token Service (STS):

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String securityToken = "<yourSecurityToken>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToken);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Configure an OSSClient instance

ClientConfiguration is a configuration class of OSSClient. You can use ClientConfiguration to configure parameters such as host proxies, connection timeouts, and the maximum number of connections. The following table describes the parameters of this operation.

Parameter	Description	Configuration method
MaxConnections	The maximum number of HTTP connections that are allowed. Default value: 1024.	ClientConfiguration.setMaxConnections
SocketTimeout	The timeout period for data transmission at the socket layer. Unit: milliseconds. Default value: 50000.	ClientConfiguration.setSocketTimeout
ConnectionTimeout	The timeout period for establishing a connection. Unit: milliseconds. Default value: 50000.	ClientConfiguration.setConnectionTimeout
ConnectionRequestTimeout	The timeout period to obtain a connection from the connection pool. Unit: milliseconds. By default, this parameter is not configured.	ClientConfiguration.setConnectionRequestTimeout
IdleConnectionTime	The idle connection timeout period to close the connection. Unit: milliseconds. Default value: 60000.	ClientConfiguration.setIdleConnectionTime
MaxErrorRetry	The maximum number of allowed retry attempts in the case of a request error. Default value: 3.	ClientConfiguration.setMaxErrorRetry
SupportCname	Specifies whether CNAME can be used as an endpoint. By default, CNAME can be used as an endpoint.	ClientConfiguration.setSupportCname
SLDEnabled	Specifies whether access over second-level domains is enabled. By default, access over second-level domains is disabled.	ClientConfiguration.setSLDEnabled
Protocol	The protocol used to connect to OSS. Default value: HTTP. Valid values: HTTP and HTTPS.	ClientConfiguration.setProtocol
UserAgent	The user agent in the HTTP header. Default value: aliyun-sdk-java.	ClientConfiguration.setUserAgent
ProxyHost	The IP address to access the proxy host.	ClientConfiguration.setProxyHost
ProxyPort	The port for the proxy host.	ClientConfiguration.setProxyPort
ProxyUsername	The username verified by the proxy host.	ClientConfiguration.setProxyUsername
ProxyPassword	The password verified by the proxy host.	ClientConfiguration.setProxyPassword

The following code provides an example on how to configure parameters for an OSSClient instance with ClientConfiguration:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create a ClientConfiguration instance. ClientConfiguration is a configuration class of OSSClient. You can use ClientConfiguration to configure parameters such as host proxies, connection timeouts, and the maximum number of connections.
ClientBuilderConfiguration conf = new ClientBuilderConfiguration();
// Configure the maximum number of HTTP connections allowed by an OSSClient instance. Default value: 1024.
conf.setMaxConnections(200);
// Configure the timeout period for data transmission at the socket layer. Unit: milliseconds. Default value: 50000.
conf.setSocketTimeout(10000);
// Configure the timeout period for establishing a connection. Unit: milliseconds. Default value: 50000.
conf.setConnectionTimeout(10000);
// Configure the timeout period for obtaining a connection from the connections pool. Unit: milliseconds. By default, this parameter is not configured.
conf.setConnectionRequestTimeout(1000);
// Configure the timeout period for idle connections. The connection is closed when the timeout period expires. Unit: milliseconds. Default value: 60000.
conf.setIdleConnectionTime(10000);
// Configure the maximum number of allowed retry attempts in the case of a request error. Default value: 3.
conf.setMaxErrorRetry(5);
// Configure whether CNAME can be used as an endpoint. By default, CNAME can be used as an endpoint.
conf.setSupportCname(true);
// Configure whether access from second-level domains is enabled. By default, access from second-level domains is disabled.
conf.setSLDEnabled(true);
// Configure the protocol used to connect to OSS. Default value: HTTP. Valid values: HTTP and HTTPS.
conf.setProtocol(Protocol.HTTP);
// Configure the user agent in the HTTP header. Default value: aliyun-sdk-java.
conf.setUserAgent("aliyun-sdk-java");
// Configure the port for the proxy host.
conf.setProxyHost("<yourProxyHost>");
// Configure the username verified by the proxy host.
conf.setProxyUsername("<yourProxyUserName>");
// Configure the password verified by the proxy host.
conf.setProxyPassword("<yourProxyPassword>");
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, conf);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.4. Quick start

This topic describes how to use OSS SDK for Java to complete routine operations such as create buckets, upload objects, and download objects.

Sample projects

OSS SDK for Java provides Maven- and Ant-based sample projects. You can compile and run the sample projects on local devices or develop your own applications based on the sample projects. For more information about how to compile and run a project, see [README.md](#) in the project directory.

- Maven-based sample project: [aliyun-oss-java-sdk-demo-mvn.zip](#)
- Ant-based sample project: [aliyun-oss-java-sdk-demo-ant.zip](#)

Create a bucket

A bucket is a global namespace in OSS. A bucket is a container for objects stored in OSS. The following code provides an example on how to create a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a bucket.
ossClient.createBucket(bucketName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about how to obtain an AccessKey ID and AccessKey secret, see [Obtain an AccessKey pair](#).

For more information about how to obtain an endpoint, see [Obtain an endpoint](#).

Upload objects

The following code provides an example on how to upload an object to OSS:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Upload content to a specified bucket and save it by using the specified object name.
String content = "Hello OSS";
ossClient.putObject(bucketName, objectName, new ByteArrayInputStream(content.getBytes()));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Download an object

The following code provides an example on how to query the content of an object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Call ossClient.getObject to query an OSSObject instance. The OSSObject instance contains the content and metadata of the object.
OSSObject ossObject = ossClient.getObject(bucketName, objectName);
// Call ossObject.getObjectContent to query an input stream of the object. Read the input stream to query the content of the object.
InputStream content = ossObject.getObjectContent();
if (content != null) {
    BufferedReader reader = new BufferedReader(new InputStreamReader(content));
    while (true) {
        String line = reader.readLine();
        if (line == null) break;
        System.out.println("\n" + line);
    }
    // You must close the connection after each use to prevent connection leaks. Otherwise, requests may be unable to establish connections and the program cannot run properly.
    content.close();
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List objects

The following code provides an example on how to list objects in a specified bucket. By default, a maximum of 100 objects are listed.

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Call ossClient.listObjects to obtain an ObjectListing instance. The ObjectListing instance contains the response to the listObject request.
ObjectListing objectListing = ossClient.listObjects(bucketName);
// Call objectListing.getObjectSummaries to query the descriptions of all objects.
for (OSSObjectSummary objectSummary : objectListing.getObjectSummaries()) {
    System.out.println(" - " + objectSummary.getKey() + "  " +
        "(size = " + objectSummary.getSize() + ")");
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete objects

For the complete code used to delete objects, visit [GitHub](#).

The following code provides an example on how to delete an object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Delete the object.
ossClient.deleteObject(bucketName, objectName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5. Buckets

1.4.1.5.1. Create buckets

A bucket is a container for objects stored in OSS. Every object is contained in a bucket. This topic describes how to create a bucket.

The following code provides an example on how to create a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a bucket.
ossClient.createBucket(bucketName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.2. List buckets

A bucket is a container for objects stored in OSS. Every object is contained in a bucket. This topic describes how to list buckets.

Buckets are listed in alphabetical order. You can list all buckets or only buckets that meet specified conditions. To list buckets of the IA or Archive storage class, use OSS SDK for Java version 2.6.0 or later.

List all buckets

The following code provides an example on how to list all buckets:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List all buckets.
List<Bucket> buckets = ossClient.listBuckets();
for (Bucket bucket : buckets) {
    System.out.println(" - " + bucket.getName());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```


List buckets whose names contain a specified prefix

The following code provides an example on how to list buckets whose names contain a specified prefix:

```
ListBucketsRequest listBucketsRequest = new ListBucketsRequest();
// List buckets whose names contain a specified prefix.
listBucketsRequest.setPrefix("<yourBucketPrefix>");
BucketList bucketList = ossClient.listBuckets(listBucketsRequest);
for (Bucket bucket : bucketList.getBucketList()) {
    System.out.println(" - " + bucket.getName());
}
```

List buckets whose names come after a specified marker

The marker parameter specifies the name of the bucket after which the list begins. The following code provides an example on how to list buckets whose names come after a specified marker:

```
ListBucketsRequest listBucketsRequest = new ListBucketsRequest();
// List buckets whose names come after the specified marker.
listBucketsRequest.setMarker("<yourBucketMarker>");
BucketList bucketList = ossClient.listBuckets(listBucketsRequest);
for (Bucket bucket : bucketList.getBucketList()) {
    System.out.println(" - " + bucket.getName());
}
```

List a specified number of buckets

The following code provides an example on how to specify the maxKeys parameter to list a specified number of buckets:

```
ListBucketsRequest listBucketsRequest = new ListBucketsRequest();
// Set the maximum number of buckets to list each time to 500. The default value is 100. The maximum value is 1000.
listBucketsRequest.setMaxKeys(500);
BucketList bucketList = ossClient.listBuckets(listBucketsRequest);
for (Bucket bucket : bucketList.getBucketList()) {
    System.out.println(" - " + bucket.getName());
}
```

1.4.1.5.3. Determine whether buckets exist

A bucket is a container for objects stored in OSS. Every object is contained in a bucket. This topic describes how to determine whether a bucket exists.

The following code provides an example on how to determine whether a specific bucket exists:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS because the account has permissions on all API
operations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on to
https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
boolean exists = ossClient.doesBucketExist("<yourBucketName>");
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.4. Manage bucket ACLs

A bucket is a container for objects stored in OSS. Every object is contained in a bucket. This topic describes how to configure and obtain the access control list (ACL) of a bucket.

Configure ACL for a bucket

The following table describes the ACLs for buckets.

ACL	Description	Value
Private	Only the owner or authorized users of the bucket can read and write the object.	CannedAccessControlList.Private
Public read	Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object. Exercise caution when you use this ACL.	CannedAccessControlList.PublicRead
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	CannedAccessControlList.PublicReadWrite

The following code provides an example on how to configure ACL for a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Set the ACL of the bucket to private.
ossClient.setBucketAcl("<yourBucketName>", CannedAccessControlList.Private);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query the ACL of a bucket

The following code provides an example on how to query the ACL of a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Query the ACL of the bucket.
AccessControlList acl = ossClient.getBucketAcl("<yourBucketName>");
System.out.println(acl.toString());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.5. Bucket tagging

You can configure bucket tagging to classify and manage buckets in Object Storage Service (OSS). For example, you can configure that only buckets with specific tags are listed in ListBucket operations.

Note

- Only the bucket owner or authorized Resource Access Management (RAM) users can configure tagging for the bucket. Otherwise, 403 Forbidden is returned with the error code AccessDenied.
- You can configure up to 20 tags for a bucket. A tag is a key-value pair.
- The key of a tag can be up to 64 bytes in length and cannot start with `http://`, `https://`, or `Aliyun`. The key of a tag cannot be empty.
- The tag value is optional. The tag value can be up to 128 bytes in length. You can leave the parameter empty.
- The key and value of a tag must be UTF-8-encoded.
- When you use PutBucketTags to configure a tag for a bucket, the new tag overwrites the existing tag.

Configure tags for a bucket

The following code provides an example on how to configure tags for a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Configure tags for the bucket.
SetBucketTaggingRequest request = new SetBucketTaggingRequest("<yourBucketName>");
request.setTag("<yourTagkey1>", "<yourTagValue1>");
request.setTag("<yourTagkey2>", "<yourTagValue2>");
ossClient.setBucketTagging(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query the tags of a bucket

The following code provides an example on how to query the tags of a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Query the tags of the bucket.
TagSet tagSet = ossClient.getBucketTagging(new GenericRequest("<yourBucketName>"));
Map<String, String> tags = tagSet.getAllTags();
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List buckets that have specified tags

The following code provides an example on how to list buckets that have specified tags:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List buckets that have a specified tag.
ListBucketsRequest listBucketsRequest = new ListBucketsRequest();
listBucketsRequest.setTag("<yourTagKey>", "<yourTagValue>");
BucketList bucketList = ossClient.listBuckets(listBucketsRequest);
for (Bucket o : bucketList.getBucketList()) {
    System.out.println("list result bucket: " + o.getName());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete the tags of a bucket

The following code provides an example on how to delete the tags of a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete the tags of the bucket.
ossClient.deleteBucketTagging(new GenericRequest("<yourBucketName>"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.6. Delete buckets

A bucket is a container for objects stored in OSS. Every object is contained in a bucket. This topic describes how to delete a bucket.

 **Note** Before you delete a bucket, you must delete all objects and parts generated by multipart upload in the bucket.

The following code provides an example on how to delete a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete the bucket.
ossClient.deleteBucket("<yourBucketName>");
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.7. Manage lifecycle rules

You can configure lifecycle rules for an OSS bucket to automatically delete expired objects to save storage space. This topic describes how to manage lifecycle rules.

Background

Each lifecycle rule contains the following information:

- Policy: specifies the mode to match objects and parts.
 - Match by prefix: matches objects and parts by prefix. You can create multiple rules to configure different prefixes. Each prefix must be unique.
 - Match by bucket: matches all objects and parts contained in the bucket. You can create only one rule if you select this method.
- Object lifecycle policy: specifies the validity period or expiration date and the operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired objects are deleted.
 - Expiration date: specifies a date. Objects that are last modified before this date expire and are deleted.
- Part lifecycle policy: specifies the validity period or expiration time and the delete operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted.
 - Expiration date: specifies a date. Parts that are last modified before this date expire and are deleted.

Lifecycle rules also apply to the parts uploaded by using `uploadPart`. In this case, the last modified time of an object is the time the multipart upload task is initiated.

Configure lifecycle rules

The following code provides an example on how to configure lifecycle rules:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
SetBucketLifecycleRequest request = new SetBucketLifecycleRequest(bucketName);
// Set the rule ID and the prefix of object names to match the rule.
String ruleId0 = "rule0";
String matchPrefix0 = "A0/";
String ruleId1 = "rule1";
String matchPrefix1 = "A1/";
String ruleId2 = "rule2";
String matchPrefix2 = "A2/";
String ruleId3 = "rule3";
String matchPrefix3 = "A3/";
// Specify that objects expire three days after they are last modified.
request.AddLifecycleRule(new LifecycleRule(ruleId0, matchPrefix0, RuleStatus.Enabled, 3));
// Specify that the objects that are last modified before the specified date expire.
LifecycleRule rule = new LifecycleRule(ruleId1, matchPrefix1, RuleStatus.Enabled);
rule.setCreatedBeforeDate(DateUtil.parseISO8601Date("2022-10-12T00:00:00.000Z"));
request.AddLifecycleRule(rule);
// Specify that parts expire three days after they are last modified.
rule = new LifecycleRule(ruleId2, matchPrefix2, RuleStatus.Enabled);
LifecycleRule.AbortMultipartUpload abortMultipartUpload = new LifecycleRule.AbortMultipartUpload();
abortMultipartUpload.setExpirationDays(3);
rule.setAbortMultipartUpload(abortMultipartUpload);
request.AddLifecycleRule(rule);
// Specify that the parts that are last modified before the specified date expire.
rule = new LifecycleRule(ruleId3, matchPrefix3, RuleStatus.Enabled);
abortMultipartUpload = new LifecycleRule.AbortMultipartUpload();
abortMultipartUpload.setCreatedBeforeDate(DateUtil.parseISO8601Date("2022-10-12T00:00:00.000Z"));
rule.setAbortMultipartUpload(abortMultipartUpload);
request.AddLifecycleRule(rule);
ossClient.setBucketLifecycle(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

View lifecycle rules

The following code provides an example on how to view lifecycle rules:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
List<LifecycleRule> rules = ossClient.getBucketLifecycle(bucketName);
for (LifecycleRule rule : rules) {
    System.out.println(rule.getId());
    System.out.println(rule.getPrefix());
    System.out.println(rule.getExpirationDays());
    System.out.println(rule.getCreatedBeforeDate());
    if (rule.hasAbortMultipartUpload()) {
        System.out.println(rule.getAbortMultipartUpload().getExpirationDays());
        System.out.println(rule.getAbortMultipartUpload().getCreatedBeforeDate());
    }
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete lifecycle rules

The following code provides an example on how to delete the lifecycle rules configured for a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
ossClient.deleteBucketLifecycle(bucketName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.8. Configure hotlink protection

OSS provides hotlink protection to prevent unauthorized domain names from accessing your OSS data. You can configure the Referer field in the HTTP header.

To configure hotlink protection, you must configure the following parameters:

- **Referer Whitelist:** Only specified domain names are allowed to access OSS resources.
- **Allow Empty Referer:** If you do not allow empty Refer fields, a request is allowed to access OSS resources only if the request includes the Referer field configured in the HTTP or HTTPS header.

Configure a Referer whitelist

The following code provides an example on how to configure the Referer whitelist for a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
List<String> refererList = new ArrayList<String>();
// Add Referer values to the whitelist. You can use an asterisk (*) or a question mark (?) as a wildcard to set the Referer parameter.
refererList.add("http://www.aliyun.com");
refererList.add("http://www. *.com");
refererList.add("http://www.?.aliyuncs.com");
// Configure BucketReferer. A value of true indicates that an empty Referer field is allowed.
BucketReferer br = new BucketReferer(true, refererList);
ossClient.setBucketReferer(bucketName, br);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query the hotlink configurations of a bucket

The following code provides an example on how to query the hotlink configurations of a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Query the Referer whitelist of the bucket.
BucketReferer br = ossClient.getBucketReferer(bucketName);
List<String> refererList = br.getRefererList();
for (String referer : refererList) {
    System.out.println(referer);
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete the hotlink configurations of a bucket

The following code provides an example on how to delete the hotlink configurations of a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// You cannot delete a Referer whitelist directly. To delete a Referer whitelist, you must create a rule that allows an empty Referer field to replace the existing rule.
BucketReferer br = new BucketReferer();
ossClient.setBucketReferer(bucketName, br);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.9. Configure CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to different regions. OSS provides CORS operations to facilitate cross-origin access control.

Configure CORS rules

The following code provides an example on how to configure CORS rules for a specific bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
SetBucketCORSRequest request = new SetBucketCORSRequest(bucketName);
// Create a container for the CORS rules. A maximum of 10 CORS rules can be configured for each bucket.
ArrayList<CORSRule> putCorsRules = new ArrayList<CORSRule>();
CORSRule corRule = new CORSRule();
ArrayList<String> allowedOrigin = new ArrayList<String>();
// Specify the allowed origins from which CORS requests are sent.
allowedOrigin.add("http://www.b.com");
ArrayList<String> allowedMethod = new ArrayList<String>();
// Specify the HTTP methods that CORS requests are allowed to use, including GET, PUT, DELETE, POST, and HEAD.
allowedMethod.add("GET");
ArrayList<String> allowedHeader = new ArrayList<String>();
// Determine whether to allow the header specified in Access-Control-Request-Headers of the preflight (OPTIONS) request.
allowedHeader.add("x-oss-test");
ArrayList<String> exposedHeader = new ArrayList<String>();
// Specify the response headers that you are allowed to access from applications.
exposedHeader.add("x-oss-test1");
// You can use only one asterisk (*) as the wildcard for AllowedOrigins and AllowedMethods in a CORS rule. Asterisks (*) indicate to allow all origin
s or operations.
corRule.setAllowedMethods(allowedMethod);
corRule.setAllowedOrigins(allowedOrigin);
// AllowedHeaders and ExposeHeaders do not support wildcards.
corRule.setAllowedHeaders(allowedHeader);
corRule.setExposeHeaders(exposedHeader);
// Specifies the time the browser can cache the response to a preflight (OPTIONS) request to a specific resource. Unit: seconds.
corRule.setMaxAgeSeconds(10);
// You can configure up to 10 CORS rules for the bucket.
putCorsRules.add(corRule);
// The existing rules are replaced.
request.setCorsRules(putCorsRules);
ossClient.setBucketCORS(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query CORS rules

The following code provides an example on how to query CORS rules configured for a specific bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
ArrayList<CORSRule> corsRules;
// Query the CORS rule list.
corsRules = (ArrayList<CORSRule>) ossClient.getBucketCORSRules(bucketName);
for (CORSRule rule : corsRules) {
    for (String allowedOrigin1 : rule.getAllowedOrigins()) {
        // Query the allowed origins from which CORS requests are sent.
        System.out.println(allowedOrigin1);
    }
    for (String allowedMethod1 : rule.getAllowedMethods()) {
        // Query the allowed cross-origin request methods.
        System.out.println(allowedMethod1);
    }
    if (rule.getAllowedHeaders().size() > 0) {
        for (String allowedHeader1 : rule.getAllowedHeaders()) {
            // Query the list of headers allowed in cross-origin requests.
            System.out.println(allowedHeader1);
        }
    }
    if (rule.getExposeHeaders().size() > 0) {
        for (String exposeHeader : rule.getExposeHeaders()) {
            // Query the list of response headers that can be accessed from an application.
            System.out.println(exposeHeader);
        }
    }
    if (null != rule.getMaxAgeSeconds()) {
        System.out.println(rule.getMaxAgeSeconds());
    }
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete CORS rules

The following code provides an example on how to delete the CORS rules configured for a specific bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete the CORS rules configured for the bucket.
ossClient.deleteBucketCORSRules(bucketName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.10. Configure logging

You can enable logging to record access to a bucket in log objects, which are stored in a specified bucket.

The log objects are in the following format: `<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString`.

Enable logging

The following code provides an example on how to enable logging for a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
SetBucketLoggingRequest request = new SetBucketLoggingRequest("<yourSourceBucketName>");
// Specify the bucket used to store log objects.
request.setTargetBucket("<yourTargetBucketName>");
// Specify the folder where log objects are stored.
request.setTargetPrefix("<yourTargetPrefix>");
ossClient.setBucketLogging(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

View logging configurations

The following code provides an example on how to view the logging configurations of a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
BucketLoggingResult result = ossClient.getBucketLogging("<yourSourceBucketName>");
System.out.println(result.getTargetBucket());
System.out.println(result.getTargetPrefix());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Disable logging

The following code provides an example on how to disable logging for a bucket:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
SetBucketLoggingRequest request = new SetBucketLoggingRequest("<yourSourceBucketName>");
request.setTargetBucket(null);
request.setTargetPrefix(null);
ossClient.setBucketLogging(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.5.11. Configure static website hosting

This topic describes how to set your bucket to the static website hosting mode. After the configurations take effect, you can use the bucket domain name to access this static website and be redirected to a specified index page or error page.

Configure static website hosting

The following code provides an example on how to configure static website hosting:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
SetBucketWebsiteRequest request = new SetBucketWebsiteRequest("<yourBucketName>");
request.setIndexDocument("index.html");
request.setErrorDocument("error.html");
ossClient.setBucketWebsite(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

View static website hosting configurations

The following code provides an example on how to view static website hosting configurations:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
BucketWebsiteResult result = ossClient.getBucketWebsite("<yourBucketName>");
System.out.println(result.getIndexDocument());
System.out.println(result.getErrorDocument());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete static website hosting configurations

The following code provides an example on how to delete static website hosting configurations:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
ossClient.deleteBucketWebsite("<yourBucketName>");
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.6. Upload objects

1.4.1.6.1. Overview

This topic describes a variety of methods OSS SDK for Java provides to upload objects.

Objects are the basic unit for data operations in OSS. OSS SDK for Java provides the following object upload methods:

- **Simple upload:** includes streaming upload and local file upload. You can use this method to upload an object up to 5 GB in size.
- **Form upload:** uploads an object up to 5 GB in size.
- **Append upload:** uploads an object up to 5 GB in size.
- **Resumable upload:** supports concurrent upload and resumable upload in which you can define the size of each part. This method is suitable for the upload of large objects. You can use this method to upload an object up to 48.8 TB in size.
- **Multipart upload:** supports the upload of an object up to 48.8 TB in size. This method is suitable for the upload of large objects.

During object upload, you can configure object metadata and view upload progress by using a progress bar. After the object is uploaded, you can perform upload callback.

1.4.1.6.2. Simple upload

Simple upload allows you to use the PutObject operation to upload a single object to OSS. This topic describes how to perform simple upload by using OSS SDK for Java.

Simple upload includes stream upload and file upload. Stream upload uses InputStream as the source object. File upload uses a local file as the source object.

 **Note** For the complete code used to perform simple upload, visit [GitHub](#).

Stream upload

Use `ossClient.putObject` to upload a data stream to OSS.

- Upload a string

The following code provides an example on how to upload a string:


```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a PutObjectRequest object.
String content = "Hello OSS";
// <yourObjectName> indicates the complete path of the object you want to upload to OSS. The path must include the extension of the object name. Example: abc/efg/123.jpg.
PutObjectRequest putObjectRequest = new PutObjectRequest("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content.getBytes()));
// Upload the string.
ossClient.putObject(putObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Upload a byte array

The following code provides an example on how to upload a byte array:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Upload the byte array.
byte[] content = "Hello OSS".getBytes();
ossClient.putObject("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Upload a network stream

The following code provides an example on how to upload a network stream:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Upload the network stream.
InputStream inputStream = new URL("https://www.aliyun.com/").openStream();
ossClient.putObject("<yourBucketName>", "<yourObjectName>", inputStream);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Upload a file stream

The following code provides an example on how to upload a file stream:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Upload the file stream.
InputStream inputStream = new FileInputStream("<yourLocalFile>");
ossClient.putObject("<yourBucketName>", "<yourObjectName>", inputStream);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Local file upload

The following code provides an example on how to upload a local file to OSS:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a PutObjectRequest object.
PutObjectRequest putObjectRequest = new PutObjectRequest("<yourBucketName>", "<yourObjectName>", new File("<yourLocalFile>"));
// Upload the local file.
ossClient.putObject(putObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.6.3. Form upload

Form upload uses an HTML form to upload an object to a specified bucket. The maximum size of an object to upload is 5 GB.

The following code provides an example on how to implement form upload:

```
public class PostObjectSample {
```

```

// Upload an object.
private String localFilePath = "<yourLocalFile>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
private String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you log on with the AccessKey pair of an Alibaba Cloud account because the account has permissions on all API operations. We recommend that you use a RAM user's credentials to call API operations or perform routine operations and maintenance. To create a RAM user, log on to https://ram.console.aliyun.com.
private String accessKeyId = "<yourAccessKeyId>";
private String accessKeySecret = "<yourAccessKeySecret>";
// Specify the bucket name.
private String bucketName = "<yourBucketName>";
// Specify the object name.
private String objectName = "<yourObjectName>";
/**
 * Start form upload.
 * @throws Exception
 */
private void PostObject() throws Exception {
    // Add the bucket name to the URL. The URL format is http://yourBucketName.oss-cn-hangzhou.aliyuncs.com.
    String urlStr = endpoint.replace("http://", "http://" + bucketName + ".");
    // Create a map for the form.
    Map<String, String> formFields = new LinkedHashMap<String, String>();
    // Specify the object name.
    formFields.put("key", this.objectName);
    // Configure Content-Disposition.
    formFields.put("Content-Disposition", "attachment;filename="
        + localFilePath);
    // Configure callback parameters.
    Callback callback = new Callback();
    // Configure the URL of the server to which you want to send the callback request. Example: http://oss-demo.aliyuncs.com:23450 or http://127.
    0.0.1:9090.
    callback.setCallbackUrl("<yourCallbackServerUrl>");
    // Configure the Host header in the callback request. Example: oss-cn-hangzhou.aliyuncs.com.
    callback.setCallbackHost("<yourCallbackServerHost>");
    // Configure the body field for the callback request.
    callback.setCallbackBody("{\"mime\\\": \"<yourMimeType>\", \"size\\\": \"<yourSize>\"}");
    // Configure Content-Type for the callback request.
    callback.setCallbackBodyType(CallbackBodyType.JSON);
    // Configure the custom parameters used to send the callback request. Each custom parameter consists of a key-value pair. The key must start
    with x: and be in lower case.
    callback.addCallbackVar("x:var1", "value1");
    callback.addCallbackVar("x:var2", "value2");
    // Configure the callback parameter in Map for the form.
    setCallBack(formFields, callback);
    // Set OSSAccessKeyId.
    formFields.put("OSSAccessKeyId", accessKeyId);
    String policy = "{\"expiration\\\": \"2120-01-01T12:00:00.000Z\\\", \"conditions\\\": [[\"content-length-range\\\", 0, 104857600]]}\"";
    String encodePolicy = new String(Base64.encodeBase64(policy.getBytes()));
    // Set the policy.
    formFields.put("policy", encodePolicy);
    // Set a signature.
    String signaturecom = com.aliyun.oss.common.auth.ServiceSignature.create().computeSignature(accessKeySecret, encodePolicy);
    // Set the signature.
    formFields.put("Signature", signaturecom);
    String ret = formUpload(urlStr, formFields, localFilePath);
    System.out.println("Post Object [" + this.objectName + "] to bucket [" + bucketName + "]");
    System.out.println("post reponse:" + ret);
}

private static String formUpload(String urlStr, Map<String, String> formFields, String localFile)
    throws Exception {
    String res = "";
    HttpURLConnection conn = null;
    String boundary = "9431149156168";
    try {
        URL url = new URL(urlStr);
        conn = (HttpURLConnection) url.openConnection();
        conn.setConnectTimeout(5000);
        conn.setReadTimeout(30000);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("User-Agent",
            "Mozilla/5.0 (Windows; U; Windows NT 6.1; zh-CN; rv:1.9.2.6)");
        // Set the MD5 hash. The MD5 hash is calculated based on the entire body.
        conn.setRequestProperty("Content-MD5", "<yourContentMD5>");
        conn.setRequestProperty("Content-Type",
            "multipart/form-data; boundary=" + boundary);
        OutputStream out = new DataOutputStream(conn.getOutputStream());
        // Recursively read all Map data in the form and write the data to the output stream.
        if (formFields != null) {
            StringBuffer strBuf = new StringBuffer();
            Iterator<Entry<String, String>> iter = formFields.entrySet().iterator();
            int i = 0;
            while (iter.hasNext()) {
                Entry<String, String> entry = iter.next();
                String inputName = entry.getKey();
                String inputValue = entry.getValue();

```

```

        if (inputValue == null) {
            continue;
        }
        if (i == 0) {
            strBuf.append("--").append(boundary).append("\r\n");
            strBuf.append("Content-Disposition: form-data; name=\""
                + inputName + "\"\r\n\r\n");
            strBuf.append(inputValue);
        } else {
            strBuf.append("\r\n").append("--").append(boundary).append("\r\n");
            strBuf.append("Content-Disposition: form-data; name=\""
                + inputName + "\"\r\n\r\n");
            strBuf.append(inputValue);
        }
        i++;
    }
    out.write(strBuf.toString().getBytes());
}

// Read the object and write it to the output stream.
File file = new File(localFile);
String filename = file.getName();
String contentType = new MimetypesFileTypeMap().getContentType(file);
if (contentType == null || contentType.equals("")) {
    contentType = "application/octet-stream";
}
StringBuffer strBuf = new StringBuffer();
strBuf.append("\r\n").append("--").append(boundary)
    .append("\r\n");
strBuf.append("Content-Disposition: form-data; name=\"file\"; "
    + "filename=\"" + filename + "\"\r\n");
strBuf.append("Content-Type: " + contentType + "\r\n\r\n");
out.write(strBuf.toString().getBytes());
DataInputStream in = new DataInputStream(new FileInputStream(file));
int bytes = 0;
byte[] bufferOut = new byte[1024];
while ((bytes = in.read(bufferOut)) != -1) {
    out.write(bufferOut, 0, bytes);
}
in.close();
byte[] endData = ("\r\n--" + boundary + "--\r\n").getBytes();
out.write(endData);
out.flush();
out.close();

// Read the returned data.
strBuf = new StringBuffer();
BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
String line = null;
while ((line = reader.readLine()) != null) {
    strBuf.append(line).append("\n");
}
res = strBuf.toString();
reader.close();
reader = null;
} catch (Exception e) {
    System.err.println("Send post request exception: " + e);
    throw e;
} finally {
    if (conn != null) {
        conn.disconnect();
        conn = null;
    }
}
return res;
}

private static void setCallBack(Map<String, String> formFields, Callback callback) {
    if (callback != null) {
        String jsonCb = OSSUtils.jsonizeCallback(callback);
        String base64Cb = BinaryUtil.toBase64String(jsonCb.getBytes());
        formFields.put("callback", base64Cb);
        if (callback.hasCallbackVar()) {
            Map<String, String> varMap = callback.getCallbackVar();
            for (Entry<String, String> entry : varMap.entrySet()) {
                formFields.put(entry.getKey(), entry.getValue());
            }
        }
    }
}

public static void main(String[] args) throws Exception {
    PostObjectSample ossPostObject = new PostObjectSample();
    ossPostObject.PostObject();
}
}

```

1.4.1.6.4. Append upload

OSS allows you to use append object to append content directly to the end of an object. This method applies to append objects.

You cannot perform copyObject for append objects.

The following code provides an example on how to perform append upload:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you log on with the AccessKey pair of an Alibaba Cloud account because the account has permissions on all API operations. We recommend that you use a RAM user's credentials to call API operations or perform routine operations and maintenance. To create a RAM user, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String content1 = "Hello OSS A \n";
String content2 = "Hello OSS B \n";
String content3 = "Hello OSS C \n";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
ObjectMetadata meta = new ObjectMetadata();
// Specify the type of content you want to upload.
meta.setContentType("text/plain");
// Configure parameters by using AppendObjectRequest.
AppendObjectRequest appendObjectRequest = new AppendObjectRequest("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content1.getBytes()), meta);
// Configure a single parameter by using AppendObjectRequest.
// Specify the bucket name.
//appendObjectRequest.setBucketName("<yourBucketName>");
// Specify the object name.
//appendObjectRequest.setKey("<yourObjectName>");
// Specify the content you want to append. Two types of content are available: InputStream and File. Select InputStream.
//appendObjectRequest.setInputStream(new ByteArrayInputStream(content1.getBytes()));
// Specify the content you want to append. Two types of content are available: InputStream and File. Select File.
//appendObjectRequest.setFile(new File("<yourLocalFile>"));
// Specify object metadata. The specified object metadata takes effect from the first append.
//appendObjectRequest.setMetadata(meta);
// Start the first append.
// Configure the position where the object is appended based on the previous object length.
appendObjectRequest.setPosition(0L);
AppendObjectResult appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Calculate the 64-bit CRC value. The value is calculated based on the ECMA-182 standard.
System.out.println(appendObjectResult.getObjectCRC());
// Start the second append.
// nextPosition indicates the position included in the next request, which is equal to the length of the current object.
appendObjectRequest.setPosition(appendObjectResult.getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayInputStream(content2.getBytes()));
appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Start the third append.
appendObjectRequest.setPosition(appendObjectResult.getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayInputStream(content3.getBytes()));
appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.6.5. Resumable upload

If the system stops responding during multipart upload, you can resume the upload by using the ListMultipartUploads and ListParts operations to retrieve all multipart upload tasks on an object and list the uploaded parts in each task. This method allows an upload task to be resumed from the last uploaded part.

The following code provides an example on how to perform resumable upload:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
ObjectMetadata meta = new ObjectMetadata();
// Specify the type of content you want to upload.
meta.setContentType("text/plain");
// Configure parameters by using UploadFileRequest.
UploadFileRequest uploadFileRequest = new UploadFileRequest("<yourBucketName>", "<yourObjectName>");
// Configure a single parameter by using UploadFileRequest.
// Specify the bucket name.
//uploadFileRequest.setBucketName("<yourBucketName>");
// Specify the object name.
//uploadFileRequest.setKey("<yourObjectName>");
// Specify the local file to upload.
uploadFileRequest.setUploadFile("<yourLocalFile>");
// Specify the number of threads for simultaneous upload. Default value: 1.
uploadFileRequest.setTaskNum(5);
// Specify the size of a part you want to upload. Valid values: 100 KB to 5 GB. The default size of each part is calculated based on the formula: Default size of each part = Object size/10000.
uploadFileRequest.setPartSize(1 * 1024 * 1024);
// Enable resumable upload. By default, resumable upload is disabled.
uploadFileRequest.setEnableCheckpoint(true);
// Specify the checkpoint file that records the upload results of each part. You need to configure this parameter when you enable resumable upload. This file stores progress information. If a part fails to be uploaded, you can continue the upload based on the progress information recorded in the checkpoint file. After the object is uploaded, the checkpoint file is deleted. By default, this file is stored in uploadFile.ucp, which is the same directory as that of the local file you want to upload.
uploadFileRequest.setCheckpointFile("<yourCheckpointFile>");
// Configure object metadata.
uploadFileRequest.setObjectMetadata(meta);
// Configure upload callback. The parameter type is Callback.
uploadFileRequest.setCallback("<yourCallbackEvent>");
// Start resumable upload.
ossClient.uploadFile(uploadFileRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.6.6. Multipart upload

OSS allows you to use multipart upload to split an object into several parts and upload them separately. After all parts are uploaded, you can combine the uploaded parts into a complete object to finish the upload.

To implement multipart upload, perform the following operations:

1. Initiate a multipart upload task.

Call the `ossClient.initiateMultipartUpload` method to obtain a unique upload ID in OSS.

2. Upload the parts.

Call the `ossClient.uploadPart` method to upload the parts.

Note

- o Part numbers identify the relative positions of parts in an object that share the same upload ID. If you have uploaded a part and used its part number again to upload another part, the latter part overwrites the former part.
- o OSS includes the MD5 hash of part data in the ETag header and returns the MD5 hash to the user.
- o OSS calculates the MD5 hash of uploaded data and compares it with the MD5 hash calculated by the SDK. If the two hashes are different, the `InvalidDigest` error code is returned.

3. Complete the multipart upload task.

After all parts are uploaded, call the `ossClient.completeMultipartUpload` method to combine these parts into a complete object.

The following code provides an example on how to perform multipart upload:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// <yourObjectName> indicates the complete path of the object you want to upload to OSS. The path must include the extension of the object name. Exam
ple: abc/efg/123.jpg.
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create an InitiateMultipartUploadRequest object.
InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest("<yourBucketName>", "<yourObjectName>");
// Optional. Specify the storage class.
// ObjectMetadata metadata = new ObjectMetadata();
// metadata.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard.toString());
// request.setObjectMetadata(metadata);
// Initiate a multipart upload task.
InitiateMultipartUploadResult upresult = ossClient.initiateMultipartUpload(request);
// Obtain an upload ID. The upload ID is the unique identification of a multipart upload task. You can initiate related requests based on the upload
ID, such as canceling a multipart upload and querying a multipart upload.
String uploadId = upresult.getUploadId();
// partETags is a set of PartETags. A PartETag consists of an ETag and a part number.
List<PartETag> partETags = new ArrayList<PartETag>();
// Calculate the total number of parts to upload.
final long partSize = 1 * 1024 * 1024L; // 1MB
final File sampleFile = new File("<localFile>");
long fileLength = sampleFile.length();
int partCount = (int) (fileLength / partSize);
if (fileLength % partSize != 0) {
    partCount++;
}
// Upload each part sequentially until all parts are uploaded.
for (int i = 0; i < partCount; i++) {
    long startPos = i * partSize;
    long curPartSize = (i + 1 == partCount) ? (fileLength - startPos) : partSize;
    InputStream instream = new FileInputStream(sampleFile);
    // Skip the uploaded parts.
    instream.skip(startPos);
    UploadPartRequest uploadPartRequest = new UploadPartRequest();
    uploadPartRequest.setBucketName(bucketName);
    uploadPartRequest.setKey(objectName);
    uploadPartRequest.setUploadId(uploadId);
    uploadPartRequest.setInputStream(instream);
    // Configure the size of each part. Each part except for the last part must be larger than 100 KB.
    uploadPartRequest.setPartSize(curPartSize);
    // Configure part numbers. Each part is configured with a part number. The number ranges from 1 to 10000. If you configure a number beyond the ra
nge, OSS returns an InvalidArgument error code.
    uploadPartRequest.setPartNumber(i + 1);
    // Parts are not necessarily uploaded in order. They may be uploaded from different OSS clients. OSS sorts the parts based on their part numbers
and combines them to obtain a complete object.
    UploadPartResult uploadPartResult = ossClient.uploadPart(uploadPartRequest);
    // After you upload a part, OSS returns a result that contains a PartETag. The PartETag is stored in partETags.
    partETags.add(uploadPartResult.getPartETag());
}
// Create a CompleteMultipartUploadRequest object.
// When the multipart upload task is complete, you must provide all valid partETags. After OSS receives the partETags, OSS verifies the validity of a
ll parts one by one. After all parts are validated, OSS combines these parts into a complete object.
CompleteMultipartUploadRequest completeMultipartUploadRequest =
    new CompleteMultipartUploadRequest(bucketName, objectName, uploadId, partETags);
// Optional. Set the ACL.
// completeMultipartUploadRequest.setObjectACL(CannedAccessControlList.PublicRead);
// Complete the multipart upload task.
CompleteMultipartUploadResult completeMultipartUploadResult = ossClient.completeMultipartUpload(completeMultipartUploadRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Cancel a multipart upload task

You can call the `ossClient.abortMultipartUpload` method to cancel a multipart upload task. If a multipart upload task is canceled, the upload ID can no longer be used to upload any parts. The parts uploaded by using that upload ID are also deleted.

The following code provides an example on how to cancel a multipart upload task:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Cancel the multipart upload task. The upload ID is returned by InitiateMultipartUpload.
AbortMultipartUploadRequest abortMultipartUploadRequest =
    new AbortMultipartUploadRequest("<yourBucketName>", "<yourObjectName>", "<uploadId>");
ossClient.abortMultipartUpload(abortMultipartUploadRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List uploaded parts

You can call the `ossClient.listParts` method to list all parts that are uploaded by using the specified upload ID.

- Simple list

The following code provides an example for simple list:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List uploaded parts. The upload ID is returned by InitiateMultipartUpload.
ListPartsRequest listPartsRequest = new ListPartsRequest("<yourBucketName>", "<yourObjectName>", "<uploadId>");
// Configure an upload ID.
// listPartsRequest.setUploadId(uploadId);
// Set the maximum number of parts to list on each page to 100. By default, a maximum of 1,000 parts can be listed.
listPartsRequest.setMaxParts(100);
// Specify the start point of the list. Only the parts whose part numbers are greater than the value of this parameter are listed.
listPartsRequest.setPartNumberMarker(2);
PartListing partListing = ossClient.listParts(listPartsRequest);
for (PartSummary part : partListing.getParts()) {
    // Query part numbers.
    part.getPartNumber();
    // Query the part size.
    part.getSize();
    // Query the ETag.
    part.getETag();
    // Query the last modified time.
    part.getLastModified();
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List all uploaded parts

By default, `listParts` can list up to 1,000 parts at a time. The following code provides an example on how to list more than 1,000 parts:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List all uploaded parts.
PartListing partListing;
ListPartsRequest listPartsRequest = new ListPartsRequest("<yourBucketName>", "<yourObjectName>", "<uploadId>");
do {
    partListing = ossClient.listParts(listPartsRequest);
    for (PartSummary part : partListing.getParts()) {
        // Query part numbers.
        part.getPartNumber();
        // Query the part size.
        part.getSize();
        // Query the ETag.
        part.getETag();
        // Query the last modified time.
        part.getLastModified();
    }
    // Specify the start point of the list. Only the parts whose part numbers are greater than the value of this parameter are listed.
    listPartsRequest.setPartNumberMarker(partListing.getNextPartNumberMarker());
} while (partListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List all uploaded parts by page

The following code provides an example on how to specify the maximum number of parts to list on each page and list all uploaded parts by page:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List all uploaded parts by page.
PartListing partListing;
ListPartsRequest listPartsRequest = new ListPartsRequest("<yourBucketName>", "<yourObjectName>", "<uploadId>");
// Set the maximum number of parts to list on each page to 100.
listPartsRequest.setMaxParts(100);
do {
    partListing = ossClient.listParts(listPartsRequest);
    for (PartSummary part : partListing.getParts()) {
        // Query part numbers.
        part.getPartNumber();
        // Query the part size.
        part.getSize();
        // Query the ETag.
        part.getETag();
        // Query the last modified time.
        part.getLastModified();
    }
    listPartsRequest.setPartNumberMarker(partListing.getNextPartNumberMarker());
} while (partListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List multipart upload tasks

You can call the `ossClient.listMultipartUploads` method to list all ongoing multipart upload tasks. Ongoing multipart upload tasks are tasks that are initiated but incomplete or tasks that are canceled. The following table describes the parameters.

Parameter	Description	Configuration method
prefix	Specifies that the returned object names must contain a specified prefix. Note that if you use a prefix for a query, the returned object names contain the prefix.	<code>ListMultipartUploadsRequest.setPrefix(String prefix)</code>
delimiter	Groups objects by name. Objects whose names contain the same string from the prefix and the next occurrence of the delimiter are grouped as a single result element in <code>CommonPrefixes</code> .	<code>ListMultipartUploadsRequest.setDelimiter(String delimiter)</code>
maxUploads	Specifies the maximum number of multipart upload tasks to return. The default value is the same as the maximum value, which is 1000.	<code>ListMultipartUploadsRequest.setMaxUploads(Integer maxUploads)</code>
keyMarker	Specifies the key-marker value from which to start the list. Multipart upload tasks are listed in alphabetical order of their object names. You can use this parameter with the <code>uploadIdMarker</code> parameter to specify the start point to list the returned results.	<code>ListMultipartUploadsRequest.setKeyMarker(String keyMarker)</code>
uploadIdMarker	<code>UploadIdMarker</code> is used with the <code>keyMarker</code> parameter to specify the start point to list the returned results. If the <code>keyMarker</code> parameter is not configured, the <code>uploadIdMarker</code> parameter is invalid. If the <code>keyMarker</code> parameter is configured, the query result includes: - All objects whose names come after the value of <code>keyMarker</code>. - All multipart upload tasks in which the object names come after the <code>keyMarker</code> value and the upload IDs are greater than the <code>uploadIdMarker</code> value.	<code>ListMultipartUploadsRequest.setUploadIdMarker(String uploadIdMarker)</code>

- Simple list

The following code provides an example for simple list:


```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List multipart upload tasks. By default, a maximum of 1,000 parts can be listed.
ListMultipartUploadsRequest listMultipartUploadsRequest = new ListMultipartUploadsRequest(bucketName);
MultipartUploadListing multipartUploadListing = ossClient.listMultipartUploads(listMultipartUploadsRequest);
for (MultipartUpload multipartUpload : multipartUploadListing.getMultipartUploads()) {
    // Query upload IDs.
    multipartUpload.getUploadId();
    // Query object names.
    multipartUpload.getKey();
    // Query the time the multipart upload tasks are initiated.
    multipartUpload.getInitiated();
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

If the value of the `isTruncated` field in the returned result is false, values of `nextKeyMarker` and `nextUploadIdMarker` are returned and used as the start point for the next object reading. If all multipart upload tasks cannot be returned at a time, list them by page.

- List all multipart upload tasks

By default, list `MultipartUploads` can list up to 1,000 multipart upload tasks at a time. The following code provides an example on how to list more than 1,000 multipart upload tasks:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List multipart upload tasks.
MultipartUploadListing multipartUploadListing;
ListMultipartUploadsRequest listMultipartUploadsRequest = new ListMultipartUploadsRequest(bucketName);
do {
    multipartUploadListing = ossClient.listMultipartUploads(listMultipartUploadsRequest);
    for (MultipartUpload multipartUpload : multipartUploadListing.getMultipartUploads()) {
        // Query upload IDs.
        multipartUpload.getUploadId();
        // Query object names.
        multipartUpload.getKey();
        // Query the time the multipart upload tasks are initiated.
        multipartUpload.getInitiated();
    }
    listMultipartUploadsRequest.setKeyMarker(multipartUploadListing.getNextKeyMarker());
    listMultipartUploadsRequest.setUploadIdMarker(multipartUploadListing.getNextUploadIdMarker());
} while (multipartUploadListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List all multipart upload tasks by page

The following code provides an example on how to list all multipart upload tasks by page:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List multipart upload tasks.
MultipartUploadListing multipartUploadListing;
ListMultipartUploadsRequest listMultipartUploadsRequest = new ListMultipartUploadsRequest(bucketName);
// Specify the number of multipart upload tasks to list on each page.
listMultipartUploadsRequest.setMaxUploads(50);
do {
    multipartUploadListing = ossClient.listMultipartUploads(listMultipartUploadsRequest);
    for (MultipartUpload multipartUpload : multipartUploadListing.getMultipartUploads()) {
        // Query upload IDs.
        multipartUpload.getUploadId();
        // Query object names.
        multipartUpload.getKey();
        // Query the time the multipart upload tasks are initiated.
        multipartUpload.getInitiated();
    }
    listMultipartUploadsRequest.setKeyMarker(multipartUploadListing.getNextKeyMarker());
    listMultipartUploadsRequest.setUploadIdMarker(multipartUploadListing.getNextUploadIdMarker());
} while (multipartUploadListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.6.7. Use a progress bar

Progress bars are used to indicate the progress of an upload or download.

The following code provides an example on how to use upload progress bars by using the `ossClient.putObject` method:

```

public class PutObjectProgressListener implements ProgressListener {
    private long bytesWritten = 0;
    private long totalBytes = -1;
    private boolean succeed = false;
    @Override
    public void progressChanged(ProgressEvent progressEvent) {
        long bytes = progressEvent.getBytes();
        ProgressEventType eventType = progressEvent.getEventType();
        switch (eventType) {
            case TRANSFER_STARTED_EVENT:
                System.out.println("Start to upload.....");
                break;
            case REQUEST_CONTENT_LENGTH_EVENT:
                this.totalBytes = bytes;
                System.out.println(this.totalBytes + " bytes in total will be uploaded to OSS");
                break;
            case REQUEST_BYTE_TRANSFER_EVENT:
                this.bytesWritten += bytes;
                if (this.totalBytes != -1) {
                    int percent = (int) (this.bytesWritten * 100.0 / this.totalBytes);
                    System.out.println(bytes + " bytes have been written at this time, upload progress: " + percent + "%(" + this.bytesWritten + "/" + this.totalBytes + ")");
                } else {
                    System.out.println(bytes + " bytes have been written at this time, upload ratio: unknown" + "(" + this.bytesWritten + "/" + "...")");
                }
                break;
            case TRANSFER_COMPLETED_EVENT:
                this.succeed = true;
                System.out.println("Succeed to upload, " + this.bytesWritten + " bytes have been transferred in total");
                break;
            case TRANSFER_FAILED_EVENT:
                System.out.println("Failed to upload, " + this.bytesWritten + " bytes have been transferred");
                break;
            default:
                break;
        }
    }
    public boolean isSucceed() {
        return succeed;
    }
    public static void main(String[] args) {
        // The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
        String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on to https://ram.console.aliyun.com.
        String accessKeyId = "<yourAccessKeyId>";
        String accessKeySecret = "<yourAccessKeySecret>";
        String bucketName = "<yourBucketName>";
        String objectName = "<yourObjectName>";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // View the upload progress information while you upload an object.
            ossClient.putObject(new PutObjectRequest(bucketName, objectName, new File("<yourLocalFile>")).
                <PutObjectRequest>withProgressListener(new PutObjectProgressListener()));
        } catch (Exception e) {
            e.printStackTrace();
        }
        // Shut down the OSSClient instance.
        ossClient.shutdown();
    }
}

```

You can view progress information when you use `ossClient.putObject`, `ossClient.getObject`, or `ossClient.uploadPart`. However, you cannot view progress information when you use `ossClient.uploadFile` or `ossClient.downloadFile`.

1.4.1.7. Download objects

1.4.1.7.1. Overview

OSS provides a variety of methods for you to download objects from OSS.

OSS SDK for Java provides the following object download methods:

- [Streaming download](#)
- [Download to local files](#)
- [Range download](#)
- [Resumable download](#)
- [Conditional download](#)

You can view the download progress by using a progress bar. For more information, see [Use a progress bar](#).

1.4.1.7.2. Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use streaming download. Streaming download enables you to download part of the object content each time until you download the entire object.

If you do not close the `ossObject` object after it is used, connection leaks may occur. As a result, no connection is available and an exception occurs. The following code provides an example on how to close an `ossObject` object:

```
OSSObject ossObject = ossClient.getObject(bucketName, objectName);
ossObject.close();
```

The following code provides an example on how to download an object by using streaming download:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// <yourObjectName> indicates the complete path of the object you want to download from OSS. The path must include the file extension of the object.
// For example, set <yourObjectName> to abc/efg/123.jpg.
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// ossObject includes the bucket name, object name, object metadata, and an input stream.
OSSObject ossObject = ossClient.getObject(bucketName, objectName);
// Read the object content.
System.out.println("Object content:");
BufferedReader reader = new BufferedReader(new InputStreamReader(ossObject.getObjectContent()));
while (true) {
    String line = reader.readLine();
    if (line == null) break;
    System.out.println("\n" + line);
}
// You must close the stream that is obtained by using ossClient.getObject after the object is read. Otherwise, connection leaks may occur. As a result, no connections are available and an exception occurs.
reader.close();
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.7.3. Download to local files

This topic describes how to download an object to a local file.

The following code provides an example on how to download a specified object to a local file:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Download the object to a local file. If the name of the object is the same as that of the local file, the object replaces the local file. If the name of the object is different from that of the local file, the object is downloaded.
ossClient.getObject(new GetObjectRequest(bucketName, objectName), new File("<yourLocalFile>"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.7.4. Range download

If you need only part of the data in an object, you can use range download to download data within the specified range.

Specify a valid range

The following code provides an example on how to specify a valid range to download an object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// <yourObjectName> indicates the complete path of the object you want to download from OSS. The path must include the file extension of the object.
// Example: abc/efg/123.jpg.
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName, objectName);
// The valid range for an object whose size is 1,000 bytes is from byte 0 to byte 999.
// Query the data that is within the range from byte 0 to byte 999, which includes a total of 1,000 bytes. If the specified range is invalid, for example, the specified range includes a negative number, or the specified value is larger than the object size, the entire object is downloaded.
getObjectRequest.setRange(0, 999);
// Start range download.
OSSObject ossObject = ossClient.getObject(getObjectRequest);
// Read the data.
byte[] buf = new byte[1024];
InputStream in = ossObject.getObjectContent();
for (int n = 0; n != -1; ) {
    n = in.read(buf, 0, buf.length);
}
// You must close the stream that is obtained by using ossClient.getObject after the object is read. Otherwise, connection leaks may occur. As a result, no connections are available and an exception occurs.
in.close();
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Streaming download may not read all data at a time. To read 64 KB of data from OSS in streaming mode, you can use the following method to read the data multiple times until 64 KB of data or the entire object is read. For more information, see [InputStream.read](#).

```
byte[] buf = new byte[1024];
InputStream in = ossObject.getObjectContent();
for (int n = 0; n != -1; ) {
    n = in.read(buf, 0, buf.length);
}
in.close();
```

Specify an invalid range

The valid range for an object whose size is 1,000 bytes is from byte 0 to byte 999. If the specified range is not within the range, the range does not take effect. Then, OSS returns the 200 HTTP status code and the data of the entire object. The following list provides examples of invalid requests and the returned results:

- If you set Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns the 200 HTTP status code and the data of the entire object.
- If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns the 200 HTTP status code and the data of the entire object.

Compatible download

If you add x-oss-range-behavior:standard to the request header, the download behavior is modified when the specified range is not within the valid range. Example: Use range download to download an object whose size is 1,000 bytes.

- If you set Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns the 206 HTTP status code and the data that is within the range from byte 500 to byte 999.
- If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns the 416 HTTP status code and the error code InvalidRange.

The following code provides an example on how to perform compatible download:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Specify an object whose size is 1,000 bytes.
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Specify the compatible behavior.
// When the value at the end of the range is invalid, OSS returns the data that is within the range from byte 500 to byte 999. The 206 HTTP status code is also returned.
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
request.setRange(500, 2000);
request.addHeader("x-oss-range-behavior", "standard");
OSSObject ossObject = ossClient.getObject(request);
ossObject.close();
System.out.println("standard get " + "500~2000 " + "statusCode:" + ossObject.getResponse().getStatusCode());
System.out.println("standard get " + "500~2000 " + "contentLength:" + ossObject.getResponse().getContentLength());
try {
    // When the value at the start of the range is invalid, exceptions are returned for the following code. The returned HTTP status code is 416 and the returned error code is InvalidRange.
    request.setRange(1000, 2000);
    request.addHeader("x-oss-range-behavior", "standard");
    ossClient.getObject(request);
} catch (OSSException e) {
    System.out.println("standard get " + "1000~2000:" + "error code:" + e.getErrorCode());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.7.5. Resumable download

You may fail to download a large object if the network is unstable or if the program stops responding. In some cases, you may still fail to download the object even after multiple attempts. To solve this problem, OSS SDK for Java provides the resumable download feature.

To enable resumable download, you need to split the object you want to download into multiple parts and download them separately. After all parts are downloaded, these parts are combined into a complete object.

You can use `ossClient.downloadFile` to enable resumable download. The following table describes the parameters you can configure for `DownloadFileRequest`.

Parameter	Description	Required	Default value	Configuration method
bucket	The name of the bucket.	Yes	None	Constructor
key	The name of the object.	Yes	None	Constructor
downloadFile	The path to the local file. The OSS object is downloaded to this file.	No	Name of the object	Constructor or <code>setUploadFile</code>
partSize	The size of each part. Valid values: 1 byte to 5 GB.	No	Object size/10000	<code>setPartSize</code>
taskNum	The number of parts you need to download simultaneously.	No	1	<code>setTaskNum</code>
enableCheckpoint	Specifies whether to enable resumable upload.	No	false	<code>setEnableCheckpoint</code>
checkpointFile	The file that records the results of multipart download. This parameter must be configured to enable resumable upload. This file stores information about download progress. If you fail to download a part, the next download continues based on the recorded progress. After the object is downloaded, the checkpoint file is deleted.	No	<code>downloadFile.ucp</code> (share the same directory with <code>DownloadFile</code>)	<code>setCheckpointFile</code>

The following code provides an example on how to perform resumable download:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Send a request to download 10 tasks simultaneously and start resumable download.
DownloadFileRequest downloadFileRequest = new DownloadFileRequest(bucketName, objectName);
downloadFileRequest.setDownloadFile("<yourDownloadFile>");
downloadFileRequest.setPartSize(1 * 1024 * 1024);
downloadFileRequest.setTaskNum(10);
downloadFileRequest.setEnableCheckpoint(true);
downloadFileRequest.setCheckpointFile("<yourCheckpointFile>");
// Download the object.
DownloadFileResult downloadRes = ossClient.downloadFile(downloadFileRequest);
// After the object is downloaded, the object metadata is returned.
downloadRes.getObjectMetadata();
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.7.6. Conditional download

This topic describes how to download objects that meet the specified conditions.

OSS allows you to configure one or more conditions for object downloads. If the specified conditions are met, the object can be downloaded. If the specified conditions are not met, an error code is returned and the object cannot be downloaded. The following table describes the download conditions.

Parameter	Description
If-Modified-Since	If the specified time is earlier than the time the object is modified, the object can be downloaded. Otherwise, 304 Not modified is returned.
If-Unmodified-Since	If the specified time is later than or equal to the time the object is modified, the object can be downloaded. Otherwise, 412 Precondition failed is returned.
If-Match	If the specified ETag matches that of an object, the object can be downloaded. Otherwise, 412 Precondition failed is returned.
If-None-Match	If the specified ETag does not match that of an object, the object can be downloaded. Otherwise, 304 Not modified is returned.

You can use If-Modified-Since and If-Unmodified-Since as object download conditions at the same time. You can also configure both If-Match and If-None-Match for object downloads.

You can obtain the ETag by using `ossClient.getObjectMeta`.

The following code provides an example on how to perform conditional download:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
// Configure the download conditions.
request.setModifiedSinceConstraint(new Date());
// Download the object to a local file.
ossClient.getObject(request, new File("<yourLocalFile>"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Conditional download supports `ossClient.getObject` and `ossClient.downloadFile`.

1.4.1.7.7. Use a progress bar

Progress bars are used to indicate the progress of an upload or download.

For the complete code of download progress bars, visit [GitHub](#).

The following code provides an example on how to use upload progress bars in `ossClient.getObject`:

```

static class GetObjectProgressListener implements ProgressListener {
    private long bytesRead = 0;
    private long totalBytes = -1;
    private boolean succeed = false;
    @Override
    public void progressChanged(ProgressEvent progressEvent) {
        long bytes = progressEvent.getBytes();
        ProgressEventType eventType = progressEvent.getEventType();
        switch (eventType) {
            case TRANSFER_STARTED_EVENT:
                System.out.println("Start to download.....");
                break;
            case RESPONSE_CONTENT_LENGTH_EVENT:
                this.totalBytes = bytes;
                System.out.println(this.totalBytes + " bytes in total will be downloaded to a local file");
                break;
            case RESPONSE_BYTE_TRANSFER_EVENT:
                this.bytesRead += bytes;
                if (this.totalBytes != -1) {
                    int percent = (int) (this.bytesRead * 100.0 / this.totalBytes);
                    System.out.println(bytes + " bytes have been read at this time, download progress: " +
                        percent + "%(" + this.bytesRead + "/" + this.totalBytes + ")");
                } else {
                    System.out.println(bytes + " bytes have been read at this time, download ratio: unknown" +
                        "(" + this.bytesRead + "/" + "...)"");
                }
                break;
            case TRANSFER_COMPLETED_EVENT:
                this.succeed = true;
                System.out.println("Succeed to download, " + this.bytesRead + " bytes have been transferred in total");
                break;
            case TRANSFER_FAILED_EVENT:
                System.out.println("Failed to download, " + this.bytesRead + " bytes have been transferred");
                break;
            default:
                break;
        }
    }
    public boolean isSucceed() {
        return succeed;
    }
}

public static void main(String[] args) {
    // The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
    String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
    String accessKeyId = "<yourAccessKeyId>";
    String accessKeySecret = "<yourAccessKeySecret>";
    String bucketName = "<yourBucketName>";
    String objectName = "<yourObjectName>";
    OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
    try {
        // View the download progress while you download an object.
        ossClient.getObject(new GetObjectRequest(bucketName, objectName).
            <GetObjectRequest>withProgressListener(new GetObjectProgressListener()),
            new File("<yourLocalFile>"));
    } catch (Exception e) {
        e.printStackTrace();
    }
    // Shut down the OSSClient instance.
    ossClient.shutdown();
}

```

You can view progress information when you use `ossClient.putObject`, `ossClient.getObject`, or `ossClient.uploadPart`. However, you cannot view progress information when you use `ossClient.uploadFile` or `ossClient.downloadFile`.

1.4.1.8. Manage objects

1.4.1.8.1. Overview

You can manage objects in a bucket by using a series of API operations such as `SetObjectAcl`, `GetObjectAcl`, `ListObjects`, `DeleteObject`, `CopyObject`, and `DoesObjectExist`.

By using the preceding operations, you can perform the following operations:

- Determine whether an object exists
- Manage object ACLs
- Manage object metadata
- List objects
- Query objects
- Delete objects
- Copy objects

1.4.1.8.2. Determine whether an object exists

This topic describes how to determine whether an object exists.

The following code provides an example on how to determine whether an object exists:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Determine whether the object exists. doesObjectExist includes the isOnlyInOSS parameter. If the value of isOnlyInOSS is true, OSS does not perform
redirection back-to-origin or mirroring back-to-origin when the object does not exist.
If the value of isOnlyInOSS is false, OSS performs redirection back-to-origin or mirroring back-to-origin when the object does not exist.
boolean found = ossClient.doesObjectExist("<yourBucketName>", "<yourObjectName>");
System.out.println(found);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.8.3. Manage object ACLs

This topic describes how to manage the access control list (ACL) of an object.

The following table describes the ACLs that can be configured for objects.

ACL	Description	Value
Inherited from bucket	The ACL of an object is the same as that of the bucket in which the object is stored.	CannedAccessControlList.Default
Private	Only the owner or authorized users of the bucket can read and write the object.	CannedAccessControlList.Private
Public read	Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object. Exercise caution when you use this ACL.	CannedAccessControlList.PublicRead
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	CannedAccessControlList.PublicReadWrite

The ACL of an object takes precedence over the ACL of the bucket in which the object is stored. For example, the ACL of a bucket is private, while the ACL of the object that is stored in this bucket is public. All users can read and write the object. If an object does not have a configured ACL, the object will inherit the ACL of the bucket in which the object is stored.

Configure ACL for an object

The following code provides an example on how to configure ACL for an object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Configure the ACL of the specified object to public read.
ossClient.setObjectAcl("<yourBucketName>", "<yourObjectName>", CannedAccessControlList.PublicRead);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query the ACL of an object

The following code provides an example on how to query the ACL of a specified object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Query the ACL of an object.
ObjectAcl objectAcl = ossClient.getObjectAcl("<yourBucketName>", "<yourObjectName>");
System.out.println(objectAcl.getPermission().toString());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.8.4. Manage object metadata

Object metadata contains HTTP headers and user metadata.

Configure object metadata

- Configure HTTP headers

The following code provides an example on how to configure HTTP headers:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String content = "Hello OSS";
// Create object metadata. You can configure HTTP headers by using object metadata.
ObjectMetadata meta = new ObjectMetadata();
String md5 = BinaryUtil.toBase64String(BinaryUtil.calculateMd5(content.getBytes()));
// Enable MD5 verification. After MD5 verification is enabled, OSS calculates the actual MD5 hash and compares this MD5 hash with that provided in the request. If the two values are different, an error occurs.
meta.setContentMD5(md5);
// Specify the type of content you want to upload. The browser determines the format and encoding type used to read the object based on the content type of the object. If the content type is not specified, a content type is generated based on the object name extension. If no extension is available, default value application/octet-stream is used as the content type.
meta.setContentType("text/plain");
// Specify the name for the object when the content is downloaded.
meta.setContentDisposition("attachment; filename=\"DownloadFilename\"");
// Specify the length of the object content to upload. If the actual object length is greater than the specified length, the object is uploaded based on the specified length. If the object is smaller than the specified length, the object is uploaded based on the actual length.
meta.setContentLength(content.length());
// Specify the caching behavior of the website when the content is downloaded.
meta.setCacheControl("Download Action");
// Specify the expiration time of the cache in the GMT.
meta.setExpirationTime(DateUtil.parseIso8601Date("2022-10-12T00:00:00.000Z"));
// Specify the content encoding format when the content is downloaded.
meta.setContentEncoding("utf-8");
// Configure the header.
meta.setHeader("<yourHeader>", "<yourHeaderValue>");
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Upload the object.
ossClient.putObject("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content.getBytes()), meta);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about HTTP headers, visit [RFC 2616](#).

• Configure user metadata

You can configure the user metadata of an object to describe the object.

The following code provides an example on how to configure the user metadata of an object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String content = "Hello OSS";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Create object metadata.
ObjectMetadata meta = new ObjectMetadata();
// Set the value of the property parameter for the user metadata to property-value. We recommend that you use the Base64 encoding method.
meta.addUserMetadata("property", "property-value");
// Upload the object.
ossClient.putObject("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content.getBytes()), meta);
// Query the object metadata.
ossClient.getObjectMetadata("<yourBucketName>", "<yourObjectName>");
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Object metadata is downloaded with the object. An object may contain multiple pieces of object metadata. The total size of the object metadata cannot exceed 8 KB.

Modify object metadata

The following code provides an example on how to modify object metadata:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String sourceBucketName = "<yourSourceBucketName>";
String sourceObjectName = "<yourSourceObjectName>";
String destinationBucketName = "<yourDestinationBucketName>";
String destinationObjectName = "<yourDestinationObjectName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Replace the source object with the target object. Call ossClient.copyObject to modify object metadata.
CopyObjectRequest request = new CopyObjectRequest(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);
ObjectMetadata meta = new ObjectMetadata();
// Specify the type of content you want to upload. The browser determines the format and encoding type used to read the object based on the content type of the object. If the content type is not specified, a content type is generated based on the object name extension. If no extension is available, the default value application/octet-stream is used as the content type.
meta.setContentType("text/plain");
// Specify the name for the object when the content is downloaded.
meta.setContentDisposition("Download File Name");
// Specify the caching behavior of the website when the content is downloaded.
meta.setCacheControl("Download Action");
// Specify the expiration time of the cache in the GMT.
meta.setExpirationTime(DateUtil.parseIso8601Date("2022-10-12T00:00:00.000Z"));
// Specify the content encoding format when the content is downloaded.
meta.setContentEncoding("utf-8");
// Configure the header.
meta.setHeader("<yourHeader>", "<yourHeaderValue>");
// Set the value of the property parameter for the user metadata to property-value.
meta.addUserMetadata("property", "property-value");
request.setNewObjectMetadata(meta);
// Modified the object metadata.
ossClient.copyObject(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query object metadata

The following table lists the methods you can use to query object metadata.

Configuration method	Description	Benefits
ossClient.getSimplifiedObjectMeta	Queries part of object metadata such as the ETag, Size, and LastModified (the last modified time) values of the object.	More lightweight and faster
ossClient.getObjectMetadata	Queries all object metadata	None

The following code provides an example on how to query the object metadata:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Query parts of the object metadata.
SimplifiedObjectMeta objectMeta = ossClient.getSimplifiedObjectMeta("<yourBucketName>", "<yourObjectName>");
System.out.println(objectMeta.getSize());
System.out.println(objectMeta.getETag());
System.out.println(objectMeta.getLastModified());
// Query all object metadata.
ObjectMetadata metadata = ossClient.getObjectMetadata("<yourBucketName>", "<yourObjectName>");
System.out.println(metadata.getContentType());
System.out.println(metadata.getLastModified());
System.out.println(metadata.getExpirationTime());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.8.5. List objects

Objects in a bucket are sorted alphabetically in OSS. You can use ossClient.listObjects to list objects in a bucket.

listObjects supports the following types of parameter formats:

- ObjectListing listObjects(String bucketName): lists up to 100 objects in a bucket.
- ObjectListing listObjects(String bucketName, String prefix): lists objects whose names contain a specified prefix in a bucket. A maximum of 100 objects can be listed.
- ObjectListing listObjects(ListObjectRequest listObjectRequest): lists objects that meet a specified condition. You can use this format to list objects in a flexible manner.

The following table describes parameters you can configure for ObjectListing.

Parameter	Description	Method
objectSummaries	The metadata of the objects that you want to list.	List<OSSObjectSummary> getObjectSummaries()
prefix	The prefix in the names of the objects that you want to list.	String getPrefix()
delimiter	The character used to group object names.	String getDelimiter()
marker	The position from which the list operation starts.	String getMarker()
maxKeys	The maximum number of objects that can be listed each time.	int getMaxKeys()
nextMarker	The position from which the next list operation starts.	String getNextMarker()
isTruncated	Indicates whether the listed objects are truncated. Valid values: - false: All objects are listed without truncation. - true: Only a part of the objects are listed.	boolean isTruncated()
commonPrefixes	A set of substrings of object names. Objects whose names end with the delimiter and contain the same prefix are grouped as a single result element in commonPrefixes.	List<String> getCommonPrefixes()
encodingType	The encoding type of the object names in the response.	String getEncodingType()

List objects by using a simple list request

The following code provides an example on how to list objects in a specified bucket. By default, a maximum of 100 objects can be listed.

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String KeyPrefix = "<yourKeyPrefix>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List objects. If you do not set KeyPrefix, all objects in the bucket are listed. If you set KeyPrefix, objects whose names contain the specified
// prefix in the bucket are listed.
ObjectListing objectListing = ossClient.listObjects(bucketName, KeyPrefix);
List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
for (OSSObjectSummary s : sums) {
    System.out.println("\t" + s.getKey());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List objects by using ListObjectsRequest

You can configure parameters of ListObjectsRequest to implement flexible queries. The following table describes the parameters you can configure for ListObjectsRequest.

Parameter	Description	Method
prefix	The prefix that must be included in the names of objects you want to list.	setPrefix(String prefix)
delimiter	The character used to group object names. Objects whose names contain the same string that stretches from the specified prefix to the first delimiter after the prefix are grouped as a CommonPrefixes element.	setDelimiter(String delimiter)
marker	The position from which the list operation starts. All objects whose names are alphabetically after the value of marker are listed.	setMarker(String marker)
maxKeys	The maximum number of objects that can be listed at a time. Maximum value: 1000. Default value: 100.	setMaxKeys(Integer maxKeys)
encodingType	The encoding type of the object names in the response. Only URL encoding is supported.	setEncodingType(String encodingType)

- List a specified number of objects in a bucket

The following code provides an example on how to list a specified number of objects:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Specify the maximum number of objects that this operation can list.
final int maxKeys = 200;
// List objects.
ObjectListing objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withMaxKeys(maxKeys));
List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
for (OSSObjectSummary s : sums) {
    System.out.println("\t" + s.getKey());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List objects whose names contain a specified prefix in a bucket

The following code provides an example on how to list objects whose names contain a specified prefix in a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Specify the prefix that is contained in the names of the objects that you want to list.
final String keyPrefix = "<yourkeyPrefix>";
// List objects whose names contain the specified prefix. By default, 100 objects are listed.
ObjectListing objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withPrefix(keyPrefix));
List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
for (OSSObjectSummary s : sums) {
    System.out.println("\t" + s.getKey());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List objects whose names are alphabetically after the value of marker

You can configure the marker parameter to specify the name of the object after which the list operation starts. The following code provides an example on how to list objects whose names are alphabetically after the specified marker:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
final String marker = "<yourMarker>";
// List objects whose names are alphabetically after the specified marker. By default, 100 objects are listed.
ObjectListing objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withMarker(marker));
List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
for (OSSObjectSummary s : sums) {
    System.out.println("\t" + s.getKey());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List all objects in a bucket by page

The following code provides an example on how to list all objects in a specified bucket by page. You can set maxKeys to specify the maximum number of objects that can be listed on each page.

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
final int maxKeys = 200;
String nextMarker = null;
ObjectListing objectListing;
do {
    objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withMarker(nextMarker).withMaxKeys(maxKeys));
    List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
    for (OSSObjectSummary s : sums) {
        System.out.println("\t" + s.getKey());
    }
    nextMarker = objectListing.getNextMarker();
} while (objectListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List objects whose names contain a specified prefix by page

The following code provides an example on how to list objects whose names contain a specified prefix by page:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List up to 200 objects with the specified prefix in their names on each page.
final int maxKeys = 200;
final String keyPrefix = "<yourkeyPrefix>";
String nextMarker = "<yourNextMarker>";
ObjectListing objectListing;
do {
    objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).
        withPrefix(keyPrefix).withMarker(nextMarker).withMaxKeys(maxKeys));
    List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
    for (OSSObjectSummary s : sums) {
        System.out.println("\t" + s.getKey());
    }
    nextMarker = objectListing.getNextMarker();
} while (objectListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- List objects whose names are encoded by a specific encoding type

Currently, OSS supports only URL encoding. If the name of an object contains one or more of the following special characters, the name must be encoded before the object is transmitted:

- Single quotation marks (')
- Double quotation marks (")
- Ampersands (&)
- Angle brackets (<>)
- Pause markers (, .)
- Chinese characters

The following code provides an example on how to list objects whose names are encoded by a specific encoding type:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
final int maxKeys = 200;
final String keyPrefix = "<yourkeyPrefix>";
String nextMarker = "<yourNextMarker>";
ObjectListing objectListing;
do {
    ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);
    listObjectsRequest.setPrefix(keyPrefix);
    listObjectsRequest.setMaxKeys(maxKeys);
    listObjectsRequest.setMarker(nextMarker);
    // Specify the encoding type of the object names.
    listObjectsRequest.setEncodingType("url");
    objectListing = ossClient.listObjects(listObjectsRequest);
    // Decode the value of Key in the response.
    for (OSSObjectSummary objectSummary: objectListing.getObjectSummaries()) {
        System.out.println("Key:" + URLDecoder.decode(objectSummary.getKey(), "UTF-8"));
    }
    // Decode the value of commonPrefixes in the response.
    for (String commonPrefixes: objectListing.getCommonPrefixes()) {
        System.out.println("CommonPrefixes:" + URLDecoder.decode(commonPrefixes, "UTF-8"));
    }
    // Decode the value of nextMarker in the response.
    if (objectListing.getNextMarker() != null) {
        nextMarker = URLDecoder.decode(objectListing.getNextMarker(), "UTF-8");
    }
} while (objectListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List objects in a bucket by directory

OSS uses a flat structure for objects instead of a hierarchical structure to store objects. A directory is a zero-byte object whose name ends with a forward slash (/). You can upload and download this object. By default, objects whose names end with a forward slash (/) are displayed as directories in the OSS console.

You can specify the delimiter and prefix parameters to list objects by directory.

- If prefix is set to the name of a directory, objects and subdirectories whose names contain the prefix are listed.
- If you also set delimiter to a forward slash (/) in the request, the objects and subdirectories whose names start with the specified prefix in the directory are listed. Each subdirectory is listed as a single result element in commonPrefixes. The objects and directories in these subdirectories are not listed.

For the complete sample code that is used to create a directory, visit [GitHub](#).

Consider a bucket that contains the following objects: oss.jpg, fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. The forward slash (/) is used as the directory delimiter. The following examples show how to list objects in the bucket.

- List all objects in the bucket

The following code provides an example on how to list all objects in the bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a ListObjectsRequest.
ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);
// List all objects in the bucket.
ObjectListing listing = ossClient.listObjects(listObjectsRequest);
// Traverse all objects.
System.out.println("Objects:");
for (OSSObjectSummary objectSummary : listing.getObjectSummaries()) {
    System.out.println(objectSummary.getKey());
}
// Traverse all commonPrefix elements.
System.out.println("CommonPrefixes:");
for (String commonPrefix : listing.getCommonPrefixes()) {
    System.out.println(commonPrefix);
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

The following objects are returned:

```
Objects:
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
oss.jpg
CommonPrefixes:
```

- List all objects in the specified directory

The following code provides an example on how to list all objects in the specified directory of the bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a ListObjectsRequest.
ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);
// Set Prefix to fun/ to list all objects in the fun/ directory.
listObjectsRequest.setPrefix("fun/");
// Lists all objects in the fun/ directory.
ObjectListing listing = ossClient.listObjects(listObjectsRequest);
// Traverse all objects.
System.out.println("Objects:");
for (OSSObjectSummary objectSummary : listing.getObjectSummaries()) {
    System.out.println(objectSummary.getKey());
}
// Traverse all commonPrefix elements.
System.out.println("\nCommonPrefixes:");
for (String commonPrefix : listing.getCommonPrefixes()) {
    System.out.println(commonPrefix);
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

The following objects are returned:

```
Objects:
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
CommonPrefixes:
```

- List objects and subdirectories in a specified directory

The following code provides an example on how to list the objects and subdirectories in a specified directory:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a ListObjectsRequest.
ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);
// Set the delimiter to a forward slash (/).
listObjectsRequest.setDelimiter("/");
// List all objects and subdirectories in the fun/ directory.
listObjectsRequest.setPrefix("fun/");
ObjectListing listing = ossClient.listObjects(listObjectsRequest);
// Traverse all objects.
System.out.println("Objects:");
// In the response, objectSummaries lists the objects in the fun/ directory.
for (OSSObjectSummary objectSummary : listing.getObjectSummaries()) {
    System.out.println(objectSummary.getKey());
}
// Traverse all commonPrefix elements.
System.out.println("\nCommonPrefixes:");
// commonPrefixes lists the subdirectories in the fun/ directory. The fun/movie/001.avi and fun/movie/007.avi objects are not listed because they are in the movie/ subdirectory of the fun/ directory.
for (String commonPrefix : listing.getCommonPrefixes()) {
    System.out.println(commonPrefix);
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

The following objects are returned:


```
Objects:
fun/test.jpg
CommonPrefixes:
fun/movie/
```

- List the sizes of objects in the specified directory

The following code provides an example on how to list the sizes of objects in a specified directory:

```
// Query the total size of objects in the specified directory.
private static long calculateFolderLength(OSS ossClient, String bucketName, String folder) {
    long size = 0L;
    ObjectListing objectListing = null;
    do {
        // The default value of MaxKeys is 100. The maximum value of MaxKeys is 1000.
        ListObjectsRequest request = new ListObjectsRequest(bucketName).withPrefix(folder).withMaxKeys(1000);
        if (objectListing != null) {
            request.setMarker(objectListing.getNextMarker());
        }
        objectListing = ossClient.listObjects(request);
        List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
        for (OSSObjectSummary s : sums) {
            size += s.getSize();
        }
    } while (objectListing.isTruncated());
    return size;
}

public void Calculate() {
    // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
    String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
    String accessKeyId = "<yourAccessKeyId>";
    String accessKeySecret = "<yourAccessKeySecret>";
    String bucketName = "<yourBucketName>";
    // Create an OSSClient instance.
    OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
    // Specify the prefix. If you want to traverse the sizes of objects in the root directory, leave this parameter empty.
    final String keyPrefix = "<YourPrefix>";
    ObjectListing objectListing = null;
    do {
        // By default, 100 objects are listed each time.
        ListObjectsRequest request = new ListObjectsRequest(bucketName).withDelimiter("/").withPrefix(keyPrefix);
        if (objectListing != null) {
            request.setMarker(objectListing.getNextMarker());
        }
        objectListing = ossClient.listObjects(request);
        List<String> folders = objectListing.getCommonPrefixes();
        for (String folder : folders) {
            System.out.println(folder + " : " + (calculateFolderLength(ossClient, bucketName, folder) / 1024) + "KB");
        }
        List<OSSObjectSummary> sums = objectListing.getObjectSummaries();
        for (OSSObjectSummary s : sums) {
            System.out.println(s.getKey() + " : " + (s.getSize() / 1024) + "KB");
        }
    } while (objectListing.isTruncated());
    ossClient.shutdown();
}
```

1.4.1.8.6. Query objects

This topic describes how to call SelectObject to query CSV and JSON objects by using OSS SDK for Java.

The following code provides an example on how to query CSV and JSON objects:

```
package samples;
import com.aliyun.oss.model.*;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.InputStreamReader;
/**
 * Examples of create select object metadata and select object.
 */
public class SelectObjectSample {
    private static String endpoint = "<endpoint, http://oss-cn-hangzhou.aliyuncs.com>";
    private static String accessKeyId = "<yourAccessKeyId>";
    private static String accessKeySecret = "<yourAccessKeySecret>";
    private static String bucketName = "<yourBucketName>";
    public static void main(String[] args) throws Exception {
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        selectCsvSample("test.csv", ossClient);
        selectJsonSample("test.json", ossClient);
        ossClient.shutdown();
    }
    private static void selectCsvSample(String key, OSS ossClient) throws Exception {
```

```

String content = "name,school,company,age\r\n" +
    "Lora Francis,School A,Staples Inc,27\r\n" +
    "Eleanor Little,School B,\"Conectiv, Inc\",43\r\n" +
    "Rosie Hughes,School C,Western Gas Resources Inc,44\r\n" +
    "Lawrence Ross,School D,MetLife Inc.,24";
ossClient.putObject(bucketName, key, new ByteArrayInputStream(content.getBytes()));
SelectObjectMetadata selectObjectMetadata = ossClient.createSelectObjectMetadata(
    new CreateSelectObjectMetadataRequest(bucketName, key)
        .withInputSerialization(
            new InputSerialization().withCsvInputFormat(
                new CSVFormat().withHeaderInfo(CSVFormat.Header.Use).withRecordDelimiter("\r\n")));
System.out.println(selectObjectMetadata.getCsvObjectMetadata().getTotalLines());
System.out.println(selectObjectMetadata.getCsvObjectMetadata().getSplits());
SelectObjectRequest selectObjectRequest =
    new SelectObjectRequest(bucketName, key)
        .withInputSerialization(
            new InputSerialization().withCsvInputFormat(
                new CSVFormat().withHeaderInfo(CSVFormat.Header.Use).withRecordDelimiter("\r\n"))
            .withOutputSerialization(new OutputSerialization().withCsvOutputFormat(new CSVFormat()));
selectObjectRequest.setExpression("select * from ossobject where _4 > 40");
OSSObject ossObject = ossClient.selectObject(selectObjectRequest);
// read object content from ossObject
BufferedReader reader = new BufferedReader(new InputStreamReader(ossObject.getObjectContent()));
while (true) {
    String line = reader.readLine();
    if (line == null) {
        break;
    }
    System.out.println(line);
}
reader.close();
ossClient.deleteObject(bucketName, key);
}

private static void selectJsonSample(String key, OSS ossClient) throws Exception {
    final String content = "{\n" +
        "\t\"name\": \"Lora Francis\",\n" +
        "\t\"age\": 27,\n" +
        "\t\"company\": \"Staples Inc\"\n" +
        "}\n" +
        "{\n" +
        "\t\"name\": \"Eleanor Little\",\n" +
        "\t\"age\": 43,\n" +
        "\t\"company\": \"Conectiv, Inc\"\n" +
        "}\n" +
        "{\n" +
        "\t\"name\": \"Rosie Hughes\",\n" +
        "\t\"age\": 44,\n" +
        "\t\"company\": \"Western Gas Resources Inc\"\n" +
        "}\n" +
        "{\n" +
        "\t\"name\": \"Lawrence Ross\",\n" +
        "\t\"age\": 24,\n" +
        "\t\"company\": \"MetLife Inc.\"\n" +
        "}";
    ossClient.putObject(bucketName, key, new ByteArrayInputStream(content.getBytes()));
    SelectObjectRequest selectObjectRequest =
        new SelectObjectRequest(bucketName, key)
            .withInputSerialization(new InputSerialization()
                .withCompressionType(CompressionType.NONE)
                .withJsonInputFormat(new JsonFormat().withJsonType(JsonType.LINES)))
            .withOutputSerialization(new OutputSerialization()
                .withCrcEnabled(true)
                .withJsonOutputFormat(new JsonFormat()))
            .withExpression("select * from ossobject as s where s.age > 40");
    OSSObject ossObject = ossClient.selectObject(selectObjectRequest);
    // read object content from ossObject
    BufferedReader reader = new BufferedReader(new InputStreamReader(ossObject.getObjectContent()));
    while (true) {
        String line = reader.readLine();
        if (line == null) {
            break;
        }
        System.out.println(line);
    }
    reader.close();
    ossClient.deleteObject(bucketName, key);
}
}

```

1.4.1.8.7. Delete objects

This topic describes how to delete objects.



Warning Deleted objects cannot be recovered. Exercise caution when you delete objects.

Delete a single object

The following code provides an example on how to delete a single object:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete the object. To delete a folder, set ObjectName to the folder name. If the folder is not empty, you must delete all objects in the folder before you can delete the folder.
ossClient.deleteObject(bucketName, objectName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete multiple objects

A maximum of 1,000 objects can be deleted each time. The deletion results can be returned in the following two modes:

- **Verbose**: returns a list of deleted objects. The default mode is Verbose.
- **quiet**: returns a list of objects you failed to delete.

The following table describes the parameters for DeleteObjectsRequest.

Parameter	Description	Configuration method
Keys	Specifies the objects you want to delete.	setKeys(List<String>)
quiet	Specifies the return mode. true indicates quiet. false indicates Verbose. Default value: false.	setQuiet(boolean)
encodingType	Specifies the encoding type of the object name in the response. Only URL encoding is supported.	setEncodingType(String)

The following table describes the parameters you can configure for DeleteObjectsResult.

Parameter	Description	Configuration method
deletedObjects	Indicates the results of the operation. The list of deleted objects is returned when the Verbose mode is used. The list of objects that fail to be deleted is returned when the quiet mode is used.	List<String> getDeletedObjects()
encodingType	Indicates the method for encoding the objects names listed in deletedObjects. When the parameter is empty, the object names are not encoded.	getEncodingType()

The following code provides an example on how to delete multiple objects at a time:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete the objects. key is equivalent to ObjectName, which indicates the complete path of the object you want to delete from OSS, and must include the extension of the object name. Example: abc/efg/123.jpg.
List<String> keys = new ArrayList<String>();
keys.add("key0");
keys.add("key1");
keys.add("key2");
DeleteObjectsResult deleteObjectsResult = ossClient.deleteObjects(new DeleteObjectsRequest(bucketName).withKeys(keys));
List<String> deletedObjects = deleteObjectsResult.getDeletedObjects();
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For the complete code used to delete multiple objects, visit [GitHub](#).

Delete objects whose names contain a specified prefix

The following code provides an example on how to delete objects whose names contain a specified prefix:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Specify the prefix.
final String prefix = "<yourkeyPrefix>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List all objects whose names contain the specified prefix and delete the objects.
String nextMarker = null;
ObjectListing objectListing = null;
do {
    ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName)
        .withPrefix(prefix)
        .withMarker(nextMarker);
    objectListing = ossClient.listObjects(listObjectsRequest);
    if (objectListing.getObjectSummaries().size() > 0) {
        List<String> keys = new ArrayList<String>();
        for (OSSObjectSummary s : objectListing.getObjectSummaries()) {
            System.out.println("key name: " + s.getKey());
            keys.add(s.getKey());
        }
        DeleteObjectsRequest deleteObjectsRequest = new DeleteObjectsRequest(bucketName).withKeys(keys);
        ossClient.deleteObjects(deleteObjectsRequest);
    }
    nextMarker = objectListing.getNextMarker();
} while (objectListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.8.8. Copy objects

This topic describes how to copy a small object and a large object.

Copy a small object

You can use `ossClient.copyObject` to copy an object that is up to 1 GB in size from a source bucket to another destination bucket within the same region. The following table describes the configuration methods of `ossClient.copyObject`.

Configuration method	Description
<code>CopyObjectResult copyObject(String sourceBucketName, String sourceKey, String destinationBucketName, String destinationKey)</code>	Allows you to specify the source bucket, source object, destination bucket, and destination object. After an object is copied, the content and metadata of the destination object are the same as that of the source object. This method is called simple copy.
<code>CopyObjectResult copyObject(CopyObjectRequest copyObjectRequest)</code>	Allows you to specify metadata of the destination object and copy conditions. If the URLs of the source object and destination object for the copy operation are the same, the metadata of the source object replaces the metadata of the destination object.

The following table describes the parameters you can configure for `CopyObjectRequest`.

Parameter	Description	Configuration method
<code>sourceBucketName</code>	The name of the source bucket.	<code>setSourceBucketName(String sourceBucketName)</code>
<code>sourceKey</code>	The name of the source object.	<code>setSourceKey(String sourceKey)</code>
<code>destinationBucketName</code>	The name of the destination bucket.	<code>setDestinationBucketName(String destinationBucketName)</code>
<code>destinationKey</code>	The name of the destination object.	<code>setDestinationKey(String destinationKey)</code>
<code>newObjectMetadata</code>	The metadata of the destination object.	<code>setNewObjectMetadata(ObjectMetadata newObjectMetadata)</code>
<code>matchingETagConstraints</code>	The condition for the copy operation. If the ETag value of the source object is the same as the ETag value provided by the user, the copy operation is performed. Otherwise, an error is returned.	<code>setMatchingETagConstraints(List<String> matchingETagConstraints)</code>
<code>nonmatchingETagConstraints</code>	The condition for the copy operation. If the ETag value of the source object is different from the ETag value provided by the user, the copy operation is performed. Otherwise, an error is returned.	<code>setNonmatchingETagConstraints(List<String> nonmatchingETagConstraints)</code>
<code>unmodifiedSinceConstraint</code>	The condition for the copy operation. If the specified time is equal to or later than the actual modified time of the source object, the copy operation is performed. Otherwise, an error is returned.	<code>setUnmodifiedSinceConstraint(Date unmodifiedSinceConstraint)</code>
<code>modifiedSinceConstraint</code>	The condition for the copy operation. If the source object is modified after the specified time, the copy operation is performed. Otherwise, an error is returned.	<code>setModifiedSinceConstraint(Date modifiedSinceConstraint)</code>

The following table describes the parameters for CopyObjectRequest.

Parameter	Description	Configuration method
etag	The unique tag of the object.	String getETag()
lastModified	The last modified time of the object.	Date getLastModified()

Note This operation has the following limits:

- You must have read and write permissions on the source object.
- You cannot copy an object from a region to another region.
- The size of the object you want to copy cannot exceed 1 GB.

The following methods can be used to copy small objects.

- **Simple copy**

The following code provides an example on how to perform simple copy:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String sourceBucketName = "<yourSourceBucketName>";
String sourceObjectName = "<yourSourceObjectName>";
String destinationBucketName = "<yourDestinationBucketName>";
String destinationObjectName = "<yourDestinationObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Copy the object.
CopyObjectResult result = ossClient.copyObject(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);
System.out.println("ETag: " + result.getETag() + " LastModified: " + result.getLastModified());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- **Use CopyObjectRequest to copy an object**

The following code provides an example on how to copy an object by using CopyObjectRequest:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String sourceBucketName = "<yourSourceBucketName>";
String sourceObjectName = "<yourSourceObjectName>";
String destinationBucketName = "<yourDestinationBucketName>";
String destinationObjectName = "<yourDestinationObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a CopyObjectRequest object.
CopyObjectRequest copyObjectRequest = new CopyObjectRequest(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);
// Configure new object metadata.
ObjectMetadata meta = new ObjectMetadata();
meta.setContentType("text/html");
copyObjectRequest.setNewObjectMetadata(meta);
// Copy the object.
CopyObjectResult result = ossClient.copyObject(copyObjectRequest);
System.out.println("ETag: " + result.getETag() + " LastModified: " + result.getLastModified());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Copy a large object

To copy an object larger than 1 GB, you must use multipart copy. To implement multipart copy, perform the following operations:

1. Use `ossClient.initiateMultipartUpload` to initiate a multipart copy task.
2. Use `ossClient.uploadPartCopy` to perform the multipart copy task. All parts except for the last part must be larger than 100 KB.
3. Use `ossClient.completeMultipartUpload` to complete the multipart copy task.

For the complete code that is used to perform multipart copy, visit [GitHub](#).

The following code provides an example on how to perform multipart copy:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String sourceBucketName = "<yourSourceBucketName>";
String sourceObjectName = "<yourSourceObjectName>";
String destinationBucketName = "<yourDestinationBucketName>";
String destinationObjectName = "<yourDestinationObjectName>";
ObjectMetadata objectMetadata = ossClient.getObjectMetadata(sourceBucketName, sourceObjectName);
// Query the size of the object to copy.
long contentLength = objectMetadata.getContentLength();
// Set the size of each part to 10 MB.
long partSize = 1024 * 1024 * 10;
// Calculate the total number of parts.
int partCount = (int) (contentLength / partSize);
if (contentLength % partSize != 0) {
    partCount++;
}
System.out.println("total part count:" + partCount);
// Initiate a multipart copy task. You can call the InitiateMultipartUploadRequest operation to specify the metadata of the target object.
InitiateMultipartUploadRequest initiateMultipartUploadRequest = new InitiateMultipartUploadRequest(destinationBucketName, destinationObjectName);
InitiateMultipartUploadResult initiateMultipartUploadResult = ossClient.initiateMultipartUpload(initiateMultipartUploadRequest);
String uploadId = initiateMultipartUploadResult.getUploadId();
// Start the multipart copy task.
List<PartETag> partETags = new ArrayList<PartETag>();
for (int i = 0; i < partCount; i++) {
    // Calculate the size of each part.
    long skipBytes = partSize * i;
    long size = partSize < contentLength - skipBytes ? partSize : contentLength - skipBytes;
    // Create an UploadPartCopyRequest object. You can call the UploadPartCopyRequest operation to specify conditions.
    UploadPartCopyRequest uploadPartCopyRequest =
        new UploadPartCopyRequest(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);
    uploadPartCopyRequest.setUploadId(uploadId);
    uploadPartCopyRequest.setPartSize(size);
    uploadPartCopyRequest.setBeginIndex(skipBytes);
    uploadPartCopyRequest.setPartNumber(i + 1);
    UploadPartCopyResult uploadPartCopyResult = ossClient.uploadPartCopy(uploadPartCopyRequest);
    // Store the returned part ETag in partETags.
    partETags.add(uploadPartCopyResult.getPartETag());
}
// Complete the multipart copy task.
CompleteMultipartUploadRequest completeMultipartUploadRequest = new CompleteMultipartUploadRequest(
    destinationBucketName, destinationObjectName, uploadId, partETags);
ossClient.completeMultipartUpload(completeMultipartUploadRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.9. Versioning

1.4.1.9.1. Versioning

The versioning state of a bucket applies to all objects in the bucket. If you enable versioning for a bucket, you can recover any previous version of an object in the bucket when you accidentally overwrite or delete the object.

A bucket can be in one of the following versioning states: unversioned (default), versioning-enabled, or versioning-suspended.

 **Notice** If versioning is enabled for a bucket, the versioning state of the bucket cannot be set back to unversioned. However, you can set the versioning state of the bucket to suspended.

Configure versioning for a bucket

The following code provides an example on how to enable or suspend versioning for a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Set the versioning state of the bucket to Enabled.
BucketVersioningConfiguration configuration = new BucketVersioningConfiguration();
configuration.setStatus(BucketVersioningConfiguration.ENABLED);
SetBucketVersioningRequest request = new SetBucketVersioningRequest(bucketName, configuration);
ossClient.setBucketVersioning(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query the versioning state of a bucket

The following code provides an example on how to query the versioning state of a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Query the versioning status of the bucket.
BucketVersioningConfiguration versionConfiguration = ossClient.getBucketVersioning("<yourBucketName>");
System.out.println("bucket versioning status: " + versionConfiguration.getStatus());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.9.2. Upload objects

This topic describes how to upload objects to a versioned bucket.

Simple upload

If you initiate a PutObject request to a versioning-enabled bucket, Object Storage Service (OSS) automatically generates a unique version ID for the uploaded object and returns the version ID as the value of the x-oss-version-id response header. In a versioning-suspended bucket, the version ID added to an uploaded object is null. If you upload an object with the same name as an existing object to a versioning-suspended bucket, the existing object is overwritten. In this case, each object has only one version whose ID is null.

The following code provides an example on how to upload an object to a versioned bucket by using simple upload:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// In this example, a string is uploaded as an object.
String content = "Hello OSS";
PutObjectResult result = ossClient.putObject("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content.getBytes()));
// Display the version ID of the uploaded object.
System.out.println("result.versionid: " + result.getVersionId());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Append upload

In a versioning-enabled bucket, the AppendObject operation can be performed only on the current version of an appendable object.

Note

- When you perform the AppendObject operation on the current version of an appendable object, OSS does not generate a previous version for the object.
- When you perform the PutObject or DeleteObject operation on the current version of an appendable object, OSS saves the appendable object as a previous version that cannot be appended any more.
- The AppendObject operation cannot be performed on the current version of a non-appendable object such as a normal object or a delete marker.

The following code provides an example on how to upload an object by using append upload:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String content1 = "Hello OSS A \n";
String content2 = "Hello OSS B \n";
String content3 = "Hello OSS C \n";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
ObjectMetadata meta = new ObjectMetadata();
// Specify the type of content that you want to upload.
meta.setContentType("text/plain");
// Configure multiple parameters by using AppendObjectRequest.
AppendObjectRequest appendObjectRequest = new AppendObjectRequest("<yourBucketName>", "<yourObjectName>", new ByteArrayInputStream(content1.getBytes()), meta);
// Configure a single parameter by using AppendObjectRequest.
// Specify the bucket name.
//appendObjectRequest.setBucketName("<yourBucketName>");
// Specify the object name.
//appendObjectRequest.setKey("<yourObjectName>");
// Configure the type of the content that you want to append. Two content types are available: InputStream and File. The following code provides an example on how to specify content of the InputStream type.
//appendObjectRequest.setInputStream(new ByteArrayInputStream(content1.getBytes()));
// Configure the type of the content that you want to append. Two content types are available: InputStream and File. The following code provides an example on how to specify content of the File type.
//appendObjectRequest.setFile(new File("<yourLocalFile>"));
// Specify the metadata of the object. You can specify the metadata only when you perform the first append operation on the object.
//appendObjectRequest.setMetadata(meta);
// Start the first append.
// Configure the position from which the append operation starts.
appendObjectRequest.setPosition(0L);
AppendObjectResult appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Calculate the CRC-64 value of the object. The value is calculated based on the ECMA-182 standard.
System.out.println(appendObjectResult.getObjectCRC());
// Start the second append.
// nextPosition indicates the position from which the next append operation starts. This is the current length of the object.
appendObjectRequest.setPosition(appendObjectResult.getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayInputStream(content2.getBytes()));
appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Start the third append.
appendObjectRequest.setPosition(appendObjectResult.getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayInputStream(content3.getBytes()));
appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Multipart upload

If you call the `CompleteMultipartUpload` operation to complete the multipart upload task for an object in a versioned bucket, OSS generates a unique version ID for the object and returns the version ID as the value of the `x-oss-version-id` response header.

The following code provides an example on how to upload an object to a versioned bucket by using multipart upload:


```

// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
/* Step 1: Initiate a multipart upload event.
*/
InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest(bucketName, objectName);
InitiateMultipartUploadResult upresult = ossClient.initiateMultipartUpload(request);
// Obtain an upload ID. The upload ID uniquely identifies the multipart upload task. You can use the upload ID to initiate related requests such as canceling and querying the multipart upload task.
String uploadId = upresult.getUploadId();
/* Step 2: Upload parts.
*/
// partETags is a set of PartETags. The PartETag of an uploaded part consists of the part number and ETag value of the part.
List<PartETag> partETags = new ArrayList<PartETag>();
// Calculate the total number of parts.
final long partSize = 1 * 1024 * 1024L; // 1MB
final File sampleFile = new File("<localFile>");
long fileLength = sampleFile.length();
int partCount = (int) (fileLength / partSize);
if (fileLength % partSize != 0) {
    partCount++;
}
// Upload each part simultaneously until all parts are uploaded.
for (int i = 0; i < partCount; i++) {
    long startPos = i * partSize;
    long curPartSize = (i + 1 == partCount) ? (fileLength - startPos) : partSize;
    InputStream instream = new FileInputStream(sampleFile);
    // Skip parts that have been uploaded.
    instream.skip(startPos);
    UploadPartRequest uploadPartRequest = new UploadPartRequest();
    uploadPartRequest.setBucketName(bucketName);
    uploadPartRequest.setKey(objectName);
    uploadPartRequest.setUploadId(uploadId);
    uploadPartRequest.setInputStream(instream);
    // Specify the size of each part. Each part must be at least 100 KB in size, except the last part.
    uploadPartRequest.setPartSize(curPartSize);
    // Configure part numbers. Each part is configured with a part number. The value ranges from 1 to 10000. If you configure a number beyond the range, OSS returns an InvalidArgument error code.
    uploadPartRequest.setPartNumber(i + 1);
    // Parts are not necessarily uploaded in order. They can be uploaded from different OSS clients. OSS sorts the parts based on their part numbers and combines them into a complete object.
    UploadPartResult uploadPartResult = ossClient.uploadPart(uploadPartRequest);
    // After you upload a part, OSS returns a result that contains a PartETag. The PartETag is stored in partETags.
    partETags.add(uploadPartResult.getPartETag());
}
/* Step 3: Complete multipart upload.
*/
// You must provide all valid PartETags when you perform this operation. OSS verifies all parts one by one after it receives PartETags. After part verification is successful, OSS combines these parts into a complete object.
CompleteMultipartUploadRequest completeMultipartUploadRequest =
    new CompleteMultipartUploadRequest(bucketName, objectName, uploadId, partETags);
CompleteMultipartUploadResult completeMultipartUploadResult = ossClient.completeMultipartUpload(completeMultipartUploadRequest);
// Display the version ID of the uploaded object.
System.out.println("restore object versionid: " + completeMultipartUploadResult.getVersionId());
// Shut down the OSSClient instance.
ossClient.shutdown();

```

1.4.1.9.3. Download objects

By default, when you call the `GetObject` operation to download an object in a versioned bucket, only the current version of the object is returned.

When you call the `GetObject` operation to download an object in a bucket, the returned result is different based on the actual scenarios:

The following code provides an example on how to download an object:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
String versionId = "<yourObjectVersionId>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Encapsulate the GetObject request.
GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName, objectName);
// Specify the version ID of the object that you want to download.
getObjectRequest.setVersionId(versionId);
// ossObject specifies the bucket name, object name, object metadata, and an input stream.
OSSObject ossObject = ossClient.getObject(getObjectRequest);
// Display the version ID of the downloaded object.
System.out.println("Get Object versionId: " + ossObject.getObjectMetadata().getVersionId());
// Read the object content.
System.out.println("Object content:");
BufferedReader reader = new BufferedReader(new InputStreamReader(ossObject.getObjectContent()));
while (true) {
    String line = reader.readLine();
    if (line == null) break;
    System.out.println("\n" + line);
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.9.4. Copy objects

This topic describes how to copy objects in a versioned bucket. You can call `CopyObject` to copy an object up to 1 GB in size and call `UploadPartCopy` to copy an object larger than 1 GB in size.

Copy a small object

You can call `CopyObject` to copy an object up to 1 GB in size from a source bucket to a destination bucket within the same region.

- By default, the `x-oss-copy-source` header in a `CopyObject` request specifies the current version of the object that you want to copy. If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404, which indicates that the object does not exist. You can add `versionId` to the `x-oss-copy-source` request header to copy the specified object version. Delete markers cannot be copied.
- You can copy a previous version of an object to the same bucket. The copied previous version becomes the new current version of the object. This way, the previous version of the object is recovered.
- If versioning is enabled for the destination bucket, OSS generates a unique version ID for the destination object. The version ID is returned as the value of the `x-oss-version-id` header in the response. If versioning is disabled or suspended for the destination bucket, OSS generates a version whose ID is null for the destination object and overwrites the original version whose ID is null.
- Appendable objects in a bucket for which versioning is enabled or suspended cannot be copied.

The following code provides an example on how to copy a small object:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
String sourceBucketName = "<yourSourceBucketName>";
String sourceObjectName = "<yourSourceObjectName>";
String destinationBucketName = "<yourDestinationBucketName>";
String destinationObjectName = "<yourDestinationObjectName>";
String versionId = "<yourSourceObjectVersionId>";
CopyObjectRequest copyObjectRequest = new CopyObjectRequest(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);
copyObjectRequest.setSourceVersionId(versionId);
CopyObjectResult copyObjectResult = ossClient.copyObject(copyObjectRequest);
System.out.println("ETag: " + copyObjectResult.getETag() + " LastModified: " + copyObjectResult.getLastModified());
System.out.println("dest object versionId: " + copyObjectResult.getVersionId());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Copy a large object

To copy an object that is larger than 1 GB in size, you must split the object into parts and copy the parts by calling `UploadPartCopy`.

By default, the `UploadPartCopy` operation is called to upload a part by copying data from the current version of an existing object. You can add `versionId` to the `x-oss-copy-source` request header as the condition to upload a part by copying data from the specified version of an existing object. Example: `x-oss-copy-source: /SourceBucketName/SourceObjectName?versionId=111111`.

Note The value of `SourceObjectName` must be URL-encoded. The version ID of the copied object is returned as the value of the `x-oss-copy-source-version-id` header in the response.

If `versionId` is not specified in the request and the current version of the source object is a delete marker, OSS returns 404 Not Found. If `versionId` is specified in the request and the specified version of the source object is a delete marker, OSS returns 400 Bad Request.

The following code provides an example on how to copy a large object by part:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String sourceBucketName = "<yourSourceBucketName>";
String sourceObjectName = "<yourSourceObjectName>";
String destinationBucketName = "<yourDestinationBucketName>";
String destinationObjectName = "<yourDestinationObjectName>";
String sourceObjectVersionId = "<yourSourceObjectVersionId>";
ObjectMetadata objectMetadata = ossClient.getObjectMetadata(sourceBucketName, sourceObjectName, sourceObjectVersionId);
// Query the size of the object that you want to copy.
long contentLength = objectMetadata.getContentLength();
// Set the size of each part to 10 MB.
long partSize = 1024 * 1024 * 10;
// Calculate the total number of parts.
int partCount = (int) (contentLength / partSize);
if (contentLength % partSize != 0) {
    partCount++;
}
System.out.println("total part count:" + partCount);
// Initiate a multipart copy task. You can use InitiateMultipartUploadRequest to specify the metadata of the destination object.
InitiateMultipartUploadRequest initiateMultipartUploadRequest = new InitiateMultipartUploadRequest(destinationBucketName, destinationObjectName);
InitiateMultipartUploadResult initiateMultipartUploadResult = ossClient.initiateMultipartUpload(initiateMultipartUploadRequest);
String uploadId = initiateMultipartUploadResult.getUploadId();
// Start the multipart copy task.
List<PartETag> partETags = new ArrayList<PartETag>();
for (int i = 0; i < partCount; i++) {
    // Calculate the size of each part.
    long skipBytes = partSize * i;
    long size = partSize < contentLength - skipBytes ? partSize : contentLength - skipBytes;
    // Create an UploadPartCopyRequest object. You can call UploadPartCopyRequest to specify conditions.
    UploadPartCopyRequest uploadPartCopyRequest =
        new UploadPartCopyRequest(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);
    uploadPartCopyRequest.setUploadId(uploadId);
    uploadPartCopyRequest.setPartSize(size);
    uploadPartCopyRequest.setBeginIndex(skipBytes);
    uploadPartCopyRequest.setPartNumber(i + 1);
    // Specify the version ID of the object that you want to copy.
    uploadPartCopyRequest.setSourceVersionId(sourceObjectVersionId);
    UploadPartCopyResult uploadPartCopyResult = ossClient.uploadPartCopy(uploadPartCopyRequest);
    // Store the returned PartETags in partETags.
    partETags.add(uploadPartCopyResult.getPartETag());
}
// Complete the multipart copy task.
CompleteMultipartUploadRequest completeMultipartUploadRequest = new CompleteMultipartUploadRequest(
    destinationBucketName, destinationObjectName, uploadId, partETags);
CompleteMultipartUploadResult completeMultipartUploadResult = ossClient.completeMultipartUpload(completeMultipartUploadRequest);
// Display the version ID of the destination object.
System.out.println("object versionid: " + completeMultipartUploadResult.getVersionId());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.9.5. List objects

This topic describes how to list objects in a versioned bucket, including all objects, a specified number of objects, and objects whose names contain a specified prefix.

List the versions of all objects in a bucket

The following code provides an example on how to list the versions of all objects including delete markers within a specified bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List the versions of all objects and delete markers within the bucket.
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker);
    versionListing = ossClient.listVersions(listVersionsRequest);
    for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
        System.out.println("key name: " + ossVersion.getKey());
        System.out.println("versionid: " + ossVersion.getVersionId());
        System.out.println("Is latest: " + ossVersion.isLatest());
        System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List the version information about objects whose names contain a specified prefix

The following code provides an example on how to list the version information about objects whose names contain a specified prefix:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List the versions of objects whose names contain the "test-" prefix.
String prefix = "test-";
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker)
        .withEncodingType(OSSConstants.URL_ENCODING)
        .withPrefix(prefix);
    versionListing = ossClient.listVersions(listVersionsRequest);
    // Display the versions of the listed objects.
    for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
        System.out.println("key name: " + ossVersion.getKey());
        System.out.println("versionid: " + ossVersion.getVersionId());
        System.out.println("Is latest: " + ossVersion.isLatest());
        System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List the version information about a specific number of objects

The following code provides an example on how to list the version information about a specific number of objects:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Specify to return up to 200 versions.
ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
    .withBucketName(bucketName)
    .withMaxResults(200);
versionListing = ossClient.listVersions(listVersionsRequest);
for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
    System.out.println("key name: " + ossVersion.getKey());
    // If versioning is not enabled for the bucket, the version IDs of the returned object are none.
    System.out.println("versionid: " + ossVersion.getVersionId());
    System.out.println("Is latest: " + ossVersion.isLatest());
    System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List the versions of all objects by page

The following code provides an example on how to list the versions of all objects by page:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List the versions of all objects in the bucket by page.
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker);
    versionListing = ossClient.listVersions(listVersionsRequest);
    for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
        System.out.println("key name: " + ossVersion.getKey());
        System.out.println("versionid: " + ossVersion.getVersionId());
        System.out.println("Is latest: " + ossVersion.isLatest());
        System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List the versions of objects whose names contain special characters

If the name of an object contains any of the following special characters, you must encode the object name before you transmit the object. OSS supports only URL encoding.

- Single quotation marks (')
- Double quotations marks (")
- Ampersands (&)
- Angle brackets (<>)
- Pause markers (, .)
- Chinese characters

The following code provides an example on how to list objects whose names contain special characters:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Specify that the object names in the response are URL-encoded.
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker)
        .withEncodingType(OSSConstants.URL_ENCODING);
    versionListing = ossClient.listVersions(listVersionsRequest);
    // Display the versions of the listed objects.
    for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
        System.out.println("key name: " + ossVersion.getKey());
        System.out.println("versionid: " + ossVersion.getVersionId());
        System.out.println("Is latest: " + ossVersion.isLatest());
        System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

List objects by directory

OSS uses a flat structure instead of a hierarchical structure to store objects. A directory is an object whose size is 0 and whose name ends with a forward slash (/). You can upload and download this object. By default, objects whose names end with a forward slash (/) are displayed as directories in the OSS console.

You can specify the delimiter and prefix parameters in the request to list objects by directory.

- If you set prefix to a directory name in the request, objects and subdirectories whose names contain the prefix are listed.
- If you also set delimiter to a forward slash (/) in the request, objects and subdirectories whose names start with the specified prefix in the directory are listed. Each subdirectory is listed as a single result element in CommonPrefixes. The objects and directories in these subdirectories are not listed.

Example: The following four objects are stored in a bucket: *oss.jpg*, *fun/test.jpg*, *fun/movie/001.avi*, and *fun/movie/007.avi*. The forward slash (/) is specified as the directory delimiter. The following examples describe how to list objects in simulated directories.

- List the version information about objects within the root directory of a bucket

The following code provides an example on how to list the version information about objects within the root directory of a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a
high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM co
nsole.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Set the delimiter parameter to a forward slash (/) to list the versions of objects and the names of subdirectories within the root directory.
String delimiter = "/";
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker)
        .withEncodingType(OSSConstants.URL_ENCODING)
        .withDelimiter(delimiter);
    versionListing = ossClient.listVersions(listVersionsRequest);
    // Display the versions of the listed objects.
    for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
        System.out.println("key name: " + ossVersion.getKey());
        System.out.println("versionid: " + ossVersion.getVersionId());
        System.out.println("Is latest: " + ossVersion.isLatest());
        System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
    }
    // Display the name of directories whose names are end with a forward slash (/) in the root folder.
    for (String commonPrefix : versionListing.getCommonPrefixes()) {
        System.out.println("commonPrefix: " + commonPrefix);
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Returned result

```
key name: oss.jpg
versionid: CAEQEhiBgMCw8Y7FqBciIGIzMDz3MTEzOWRiMDRmZmFhMmRlMj1jZWl0MWU4****
Is latest: true
Is delete marker: false
commonPrefix: fun/
```

- List objects and subdirectories in a specified directory

The following code provides an example on how to list objects and subdirectories within a directory in a bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a
high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM co
nsole.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Configure the prefix parameter to list all objects and subdirectories within the fun directory. Set the delimiter parameter to a forward slash (
/).
String prefix = "fun/";
String delimiter = "/";
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker)
        .withEncodingType(OSSConstants.URL_ENCODING)
        .withDelimiter(delimiter)
        .withPrefix(prefix);
    versionListing = ossClient.listVersions(listVersionsRequest);
    // Display the versions of the listed objects.
    for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
        System.out.println("key name: " + ossVersion.getKey());
        System.out.println("versionid: " + ossVersion.getVersionId());
        System.out.println("Is latest: " + ossVersion.isLatest());
        System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
    }
    // Display the names of the subdirectories within the fun directory.
    for (String commonPrefix : versionListing.getCommonPrefixes()) {
        System.out.println("commonPrefix: " + commonPrefix);
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Returned results

```
key name: fun/test.jpg
versionid: CAEQEhiBgIC48Y7FqBciIDU4NjAzMjczMtY5NDRjYmVhNGY4NTM2YTdjY2Ji****
Is latest: true
Is delete marker: false
commonPrefix: fun/movie/
```

1.4.1.9.6. Delete objects

This topic describes how to delete a single object, multiple objects, or objects whose names contain a specific prefix from a versioned bucket.

Delete operations in a versioned bucket

When you delete an object from a versioned bucket, you must determine whether to specify a version ID in the request.

- Delete an object without specifying a version ID (temporary deletion)

By default, if you do not specify the version ID of the object that you want to delete in the request, Object Storage Service (OSS) does not delete the current version of the object but adds a delete marker to the object as the new current version. If you perform the `GetObject` operation on the object, OSS identifies the current version of the object as a delete marker and returns `404 Not Found`. In addition, the `x-oss-delete-marker` header and the `x-oss-version-id` header that indicates the version ID of the created delete marker are included in the response.

The value of the `x-oss-delete-marker` header in the response is true, which indicates that the `x-oss-version-id` value is the version ID of a delete marker.

- Delete an object by specifying a version ID (permanent deletion)

If you specify the version ID of the object that you want to delete in the request, OSS permanently deletes the version specified by the `versionId` field in the `params` parameter. To delete a version whose ID is null, add `params['versionId'] = "null"` to `params` in the request. OSS identifies the string "null" as the ID of the version to delete and deletes the version whose ID is null.

Delete a single object

The following examples show how to permanently or temporarily delete a single object from a versioned bucket.

- Permanently delete an object from a versioned bucket

The following code provides an example on how to permanently delete an object from a versioned bucket by specifying the version ID of the object in the request:


```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a
high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a R
AM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Specify the complete name of the object that you want to delete, including the object extension. Example: example/test.jpg.
String objectName = "<yourObjectName>";
// You can specify the version ID of an object or a delete marker.
String versionid = "<yourObjectVersionidOrDelMarkerVersionid>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete the object that has the specified version ID.
ossClient.deleteVersion(bucketName, objectName, versionid);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Temporarily delete an object from a versioned bucket

The following code provides an example on how to temporarily delete an object from a versioned bucket without specifying a version ID:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a
high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a R
AM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Temporarily delete the object. A delete marker is added to the object.
ossClient.deleteObject(bucketName, objectName);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete multiple objects

The following examples show how to permanently or temporarily delete multiple objects from a versioned bucket.

- Permanently delete multiple objects from a versioned bucket

The following code provides an example on how to permanently delete multiple objects or delete markers with specified version IDs from a versioned bucket:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a
high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a R
AM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Permanently delete the objects or delete markers that have specified version IDs.
List<KeyVersion> keyVersionsList = new ArrayList<KeyVersion>();
keyVersionsList.add(new KeyVersion("<yourObject3Name>", "<yourObject3NameVersionid>"));
keyVersionsList.add(new KeyVersion("<yourObject4Name>", "<yourObject4DelMarkerVersionid>"));
DeleteVersionsRequest delVersionsRequest = new DeleteVersionsRequest(bucketName);
delVersionsRequest.setKeys(keyVersionsList);
// Initiate a deleteVersions request.
DeleteVersionsResult delVersionsResult = ossClient.deleteVersions(delVersionsRequest);
// Display the deletion result.
for (DeletedVersion delVer : delVersionsResult.getDeletedVersions()) {
    String keyName = URLDecoder.decode(delVer.getKey(), "UTF-8");
    String keyVersionId = delVer.getVersionId();
    String keyDelMarkerId = delVer.getDeleteMarkerVersionId();
    System.out.println("delete key: " + keyName);
    System.out.println("delete key versionid: " + keyVersionId);
    if (keyDelMarkerId != null && keyDelMarkerId.length() != 0) {
        System.out.println("delete key del_marker versionid: " + keyDelMarkerId);
    }
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Temporarily delete multiple objects from a versioned bucket

The following code provides an example on how to delete multiple objects by sending a request in which no version ID is specified:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a
// high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM
// user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String versionId = "<yourObjectVersionId>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Delete multiple objects at a time.
List<String> KeysList = new ArrayList<String>();
KeysList.add("<yourObject1Name>");
KeysList.add("<yourObject2Name>");
DeleteObjectsRequest request = new DeleteObjectsRequest(bucketName);
request.setKeys(KeysList);
// Initiate a deleteObjects request.
DeleteObjectsResult delObjResult = ossClient.deleteObjects(request);
// Display the deletion result.
for (String o : delObjResult.getDeletedObjects()) {
    String keyName = URLDecoder.decode(o, "UTF-8");
    System.out.println("delete key name: " + keyName);
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Delete objects whose names contain a specified prefix

The following code provides an example on how to delete objects whose names contain a specified prefix:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a hi
// gh-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM u
// ser, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
// Specify the prefix that is contained in the names of the objects that you want to delete.
final String prefix = "<yourkeyPrefix>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List the versions of all objects whose names contain the specified prefix and delete the versions.
String nextKeyMarker = null;
String nextVersionMarker = null;
VersionListing versionListing = null;
do {
    ListVersionsRequest listVersionsRequest = new ListVersionsRequest()
        .withBucketName(bucketName)
        .withKeyMarker(nextKeyMarker)
        .withVersionIdMarker(nextVersionMarker)
        .withPrefix(prefix);
    versionListing = ossClient.listVersions(listVersionsRequest);
    if (versionListing.getVersionSummaries().size() > 0) {
        List<KeyVersion> keyVersionsList = new ArrayList<KeyVersion>();
        for (OSSVersionSummary ossVersion : versionListing.getVersionSummaries()) {
            System.out.println("key name: " + ossVersion.getKey());
            System.out.println("versionid: " + ossVersion.getVersionId());
            System.out.println("Is delete marker: " + ossVersion.isDeleteMarker());
            keyVersionsList.add(new KeyVersion(ossVersion.getKey(), ossVersion.getVersionId()));
        }
        DeleteVersionsRequest delVersionsRequest = new DeleteVersionsRequest(bucketName).withKeys(keyVersionsList);
        ossClient.deleteVersions(delVersionsRequest);
    }
    nextKeyMarker = versionListing.getNextKeyMarker();
    nextVersionMarker = versionListing.getNextVersionIdMarker();
} while (versionListing.isTruncated());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.9.7. Query object metadata

By default, when you call the `HeadObject` operation on an object in a versioned bucket, only the metadata of the current version of the object is returned.

 **Note** If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404 Not Found. If you specify a version ID in the request, OSS returns the metadata of the specified version of the object.

The following code provides an example on how to query the metadata of an object:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Query partial metadata of the specified version of the object.
GenericRequest getSimplifiedObjectMetaRequest = new GenericRequest("<yourBucketName>", "<yourObjectName>", "<yourVersionId>");
SimplifiedObjectMeta objectMeta = ossClient.getSimplifiedObjectMeta(getSimplifiedObjectMetaRequest);
System.out.println(objectMeta.getSize());
System.out.println(objectMeta.getETag());
System.out.println(objectMeta.getLastModified());
// Query all metadata of the specified version of the object.
GenericRequest getObjectMetadataRequest = new GenericRequest("<yourBucketName>", "<yourObjectName>", "<yourObjectVersionId>");
ObjectMetadata metadata = ossClient.getObjectMetadata(getObjectMetadataRequest);
System.out.println(metadata.getContentType());
System.out.println(metadata.getLastModified());
System.out.println(metadata.getExpirationTime());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.9.8. Manage the ACL of an object

This topic describes how to manage the access control lists (ACLs) of objects in a versioned bucket.

Configure the ACL of an object

By default, the PutObjectACL operation configures the ACL of the current version of an object. If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404 Not Found. You can specify a version ID in the request to configure the ACL of a specified version of an object.

The following code provides an example on how to configure the ACL of an object:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
String versionId = "<yourObjectVersionId>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a SetObjectAclRequest object. In this example, the ACL of the object is set to public-read.
SetObjectAclRequest setObjectAclRequest = new SetObjectAclRequest(bucketName, objectName, versionId, CannedAccessControlList.PublicRead);
// Configure the ACL of the specified version of the object.
ossClient.setObjectAcl(setObjectAclRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Query the ACL of an object.

By default, the GetObjectACL operation queries the ACL of the current version of an object. If the current version of the object is a delete marker, OSS returns 404 Not Found. You can specify a version ID in the request to query the ACL of the specified version of an object.

The following code provides an example on how to query the ACL of an object:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
String versionId = "<yourObjectVersionId>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Query the ACL of the object.
GenericRequest genericRequest = new GenericRequest(bucketName, objectName, versionId);
ObjectAcl objectAcl = ossClient.getObjectAcl(genericRequest);
System.out.println("get object acl: " + objectAcl.getPermission().toString());
System.out.println("object versionid: " + objectAcl.getVersionId());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.10. Authorized access

This topic describes how to use Security Token Service (STS) and a signed URL to authorize access.

Use STS to authorize temporary access

Alibaba Cloud Security Token Service (STS) is a web service that provides a temporary access token to a cloud computing user. You can use STS to grant a third-party application or your RAM user an access credential with a customized validity period and permissions.

STS has the following benefits:

- You need only to generate an access token and send the access token to a third-party application, instead of exposing your long-term AccessKey pair to the third-party application. You can customize the access permissions and validity period of this token.
- The access token automatically expires when the validity period ends.

The following code provides an example on how to create a signature request by using STS:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String securityToken = "<yourSecurityToken>";
// After you obtain a temporary STS credential for your OSSClient, an OSSClient instance is created based on the security token and temporary AccessKey pair (AccessKey ID and AccessKey secret) contained in the credential.
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToken);
// Perform an operation in OSS. For example, you can upload or download an object.
// Upload an object to OSS.
// ossClient.putObject(putObjectRequest);
// Download an object to a local file. If the name of the object is the same as that of the local file, the object replaces the local file. If the name of the object is different from that of the local file, the object is downloaded.
// ossClient.getObject(new GetObjectRequest(bucketName, objectName), new File("<yourLocalFile>"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use a signed URL to authorize temporary access

- Generate a signed URL

You can generate a signed URL and provide it to a visitor to grant temporary access. When you generate a signed URL, you can specify the validity period of the URL to restrict the period of access from visitors.

- Generate a signed URL to allow HTTP GET requests

The following code provides an example on how to generate a signed URL to allow HTTP GET requests:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Set the validity period of the URL to one hour.
Date expiration = new Date(new Date().getTime() + 3600 * 1000);
// Generate a signed URL to allow HTTP GET requests. Visitors can use a browser for access.
URL url = ossClient.generatePresignedUrl(bucketName, objectName, expiration);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Generate a signed URL to allow other HTTP methods

If you want to allow visitors to temporarily perform other operations such as uploading or deleting objects, you must generate a corresponding signed URL. The following code provides an example on how to upload an object by using a signed URL to allow HTTP PUT requests:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String securityToken = "<yourSecurityToken>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// After you obtain a temporary STS credential for your OSSClient, an OSSClient instance is created based on the security token and temporary AccessKey pair (AccessKey ID and AccessKey secret) contained in the credential.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToken);
GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(bucketName, objectName, HttpMethod.PUT);
// Set the validity period of the URL to one hour.
Date expiration = new Date(new Date().getTime() + 3600 * 1000);
request.setExpiration(expiration);
// Set ContentType.
request.setContentType(DEFAULT_OBJECT_CONTENT_TYPE);
// Set user metadata.
request.addUserMetadata("author", "aliy");
// Generate a signed URL to allow HTTP PUT requests.
URL signedUrl = ossClient.generatePresignedUrl(request);
Map<String, String> requestHeaders = new HashMap<String, String>();
requestHeaders.put(HttpHeaders.CONTENT_TYPE, DEFAULT_OBJECT_CONTENT_TYPE);
requestHeaders.put(OSS_USER_METADATA_PREFIX + "author", "aliy");
// Use the signed URL to upload an object.
ossClient.putObject(signedUrl, new ByteArrayInputStream("Hello OSS".getBytes()), -1, requestHeaders, true);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Visitors can specify the `HttpMethod.PUT` parameter and use the signed URL to upload objects.

- Generate a signed URL with specified parameters

The following code provides an example on how to generate a signed URL with specified parameters:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a request.
GeneratePresignedUrlRequest generatePresignedUrlRequest = new GeneratePresignedUrlRequest(bucketName, objectName);
// Set HttpMethod to PUT.
generatePresignedUrlRequest.setMethod(HttpMethod.PUT);
// Add user metadata.
generatePresignedUrlRequest.addUserMetadata("author", "baymax");
// Set Content-Type.
generatePresignedUrlRequest.setContentType("application/octet-stream");
// Set the validity period of the URL to one hour.
Date expiration = new Date(new Date().getTime() + 3600 * 1000);
generatePresignedUrlRequest.setExpiration(expiration);
// Generate a signed URL.
URL url = ossClient.generatePresignedUrl(generatePresignedUrlRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Use a signed URL to upload or download an object

- Use a signed URL to download an object

The following code provides an example on how to download a specified object by using a signed URL:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
Date expiration = DateUtil.parseRfc822Date("Wed, 18 Mar 2022 14:20:00 GMT");
GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(bucketName, objectName, HttpMethod.GET);
// Set the validity period.
request.setExpiration(expiration);
// Generate a signed URL to allow HTTP GET requests.
URL signedUrl = ossClient.generatePresignedUrl(request);
System.out.println("signed url for getObject: " + signedUrl);
// Use the signed URL to send a request.
Map<String, String> customHeaders = new HashMap<String, String>();
// Specify the request headers of GetObject.
customHeaders.put("Range", "bytes=100-1000");
OSSObject object = ossClient.getObject(signedUrl, customHeaders);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

- Use a signed URL to upload an object

The following code provides an example on how to upload an object by using a signed URL:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Generate a signed URL.
Date expiration = DateUtil.parseRfc822Date("Thu, 19 Mar 2019 18:00:00 GMT");
GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(bucketName, objectName, HttpMethod.PUT);
// Set the validity period.
request.setExpiration(expiration);
// Set Content-Type.
request.setContentType("application/octet-stream");
// Add user metadata.
request.addUserMetadata("author", "aliy");
// Generate a signed URL to allow HTTP PUT requests.
URL signedUrl = ossClient.generatePresignedUrl(request);
System.out.println("signed url for putObject: " + signedUrl);
// Use the signed URL to send a request.
File f = new File("<yourLocalFile>");
FileInputStream fin = new FileInputStream(f);
// Specify the request headers of PutObject.
Map<String, String> customHeaders = new HashMap<String, String>();
customHeaders.put("Content-Type", "application/octet-stream");
customHeaders.put("x-oss-meta-author", "aliy");
PutObjectResult result = ossClient.putObject(signedUrl, fin, f.length(), customHeaders);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

1.4.1.11. IMG

OSS Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service that can process large amounts of data. After you upload source images to OSS, you can call RESTful API operations to process the images anytime, anywhere, and on any Internet device. You can use OSS IMG to build your own image processing service.

Basic features

OSS IMG provides the following features:

- Query image information.
- Convert image formats.
- Resize, crop, and rotate images.
- Apply image effect.
- Add image, text, and text-and-image watermarks to images.
- Customize image processing styles. Log on to the OSS console. Click a bucket name to go to the bucket details page. Click the **Image Processing (IMG)** tab and click **Create Rule** to customize an image processing style.
- Use cascade processing to call multiple IMG features.

Usage

You can use standard `HTTP GET` requests to access IMG. All processing parameters are encoded in the query string of a URL.

- Anonymous access

If the ACL of an image object is public-read, as described in the following table, anonymous users can access the image object.

Bucket ACL	Object ACL
public-read or public-read-write	default
Any ACL	public-read or public-read-write

You can anonymously access IMG by using a third-level domain in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=image/action,param_value
```

Parameter	Description
bucket	The name of the bucket.
endpoint	The domain name used to access the region.
object	The name of the image object.
image	The identifier reserved by IMG.
action	The operations performed on images, such as scaling, cropping, and rotating.
param	The parameters corresponding to the operations performed on the images.

- Basic operations

For example, you can scale an image to a width of 100 and adjust the height based on the ratio:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customize image styles

You can anonymously access IMG by using a third-level domain in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- style: the identifier reserved by IMG for the custom style.
- yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when you create the custom style in the console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade processing

You can use cascade processing to perform multiple operations in sequence on an image by using a URL in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Access by using HTTPS

IMG supports access by using HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Authorized access

If the ACL of an image object is private, you must have permissions on the image object before you can perform operations on it.

Bucket ACL	Object ACL
private	default
Any ACL	private

The following code provides an example on how to generate a signed URL to access IMG:

```
String endpoint = "<Your OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
String bucketName = "<bucketName>";
private static String key = "example.jpg";
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Configure the IMG style.
String style = "image/resize,m_fixed,w_100,h_100/rotate,90";
// Set the validity period to 10 minutes.
Date expiration = new Date(new Date().getTime() + 1000 * 60 * 10);
GeneratePresignedUrlRequest req = new GeneratePresignedUrlRequest(bucketName, key, HttpMethod.GET);
req.setExpiration(expiration);
req.setProcess(style);
URL signedUrl = ossClient.generatePresignedUrl(req);
System.out.println(signedUrl);
```

Note

- You can set Create Style, HTTPS, and Cascading Processing for authorized access.
- Use code in the following format to specify the expiration time: `Date expiration = DateUtil.parseRfc822Date("Wed, 21 Dec 2022 14:20:00 GMT");`.

- Access by using SDKs

You can use OSS SDKs to access and process image objects of any ACLs.

Note

- For the complete IMG code, visit [GitHub](#).
- You can use OSS SDKs to process image objects by configuring Create Style, HTTPS, and Cascading Processing.

- Basic operations

The following code provides an example on how to perform basic IMG operations including obtaining image information, converting image formats, resizing, cropping, rotating, watermarking, and applying effects:

```
String endpoint = "<Your OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
String bucketName = "<bucketName>";
String key = "example.jpg";
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Resize the image.
String style = "image/resize,m_fixed,w_100,h_100";
GetObjectRequest request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-resize.jpg"));
// Crop the image.
style = "image/crop,w_100,h_100,x_100,y_100,r_1";
request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-crop.jpg"));
// Rotate the image.
style = "image/rotate,90";
request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-rotate.jpg"));
// Sharpen the image.
style = "image/sharpen,100";
request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-sharpen.jpg"));
// Add watermarks to the image.
style = "image/watermark,text_SGVsbG8g5Zu-54mh5pyN5YqhIQ";
request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-watermark.jpg"));
// Convert the image format.
style = "image/format,png";
request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-format.png"));
// Obtain the image information.
style = "image/info";
request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-info.txt"));
ossClient.shutdown();
```

- Custom image styles

```
String endpoint = "<Your OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
String bucketName = "<bucketName>";
String key = "example.jpg";
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Customize the image style.
String style = "style/oss-pic-style-w-100";
GetObjectRequest request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-new.jpg"));
ossClient.shutdown();
```


- Cascade processing

```
String endpoint = "<Your OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
String bucketName = "<bucketName>";
private static String key = "example.jpg";
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);
// Perform cascade processing.
String style = "image/resize,m_fixed,w_100,h_100/rotate,90";
GetObjectRequest request = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-new.jpg"));
ossClient.shutdown();
```

1.4.1.12. Handle exceptions

If an exception occurs when you call the OSSClient class-related API operation, the related operation fails. Otherwise, the related operation succeeds. If an exception occurs, the data returned by the operation is invalid. OSS SDK for Java exceptions are classified into two types: `OSSEException` and `ClientException`. Both are subclasses of `RuntimeException`.

Examples

```
try {
    // Do some operations with the instance here, such as put object...
    client.putObject(...);
} catch (OSSEException oe) {
    System.out.println("Caught an OSSEException, which means your request made it to OSS, "
        + "but was rejected with an error response for some reason.");
    System.out.println("Error Message: " + oe.getErrorMessage());
    System.out.println("Error Code: " + oe.getErrorCode());
    System.out.println("Request ID: " + oe.getRequestId());
    System.out.println("Host ID: " + oe.getHostId());
} catch (ClientException ce) {
    System.out.println("Caught an ClientException, which means the client encountered "
        + "a serious internal problem while trying to communicate with OSS, "
        + "such as not being able to access the network.");
    System.out.println("Error Message: " + ce.getMessage());
} finally {
    if (client != null) {
        client.shutdown();
    }
}
```

ClientException

`ClientException` indicates the exception that occurs when a client sends a request or transmit data to OSS. For example, `ClientException` is returned when a request is sent under poor network conditions. `ClientException` is also returned when an I/O exception occurs during object upload.

OSSEException

`OSSEException` indicates an error response from OSS. The error code and error message are returned so that you can find the problem and handle it properly.

`OSSEException` involves the following error information:

- Code: the error code that OSS returns to the user.
- Message: the detailed error message provided by OSS.
- RequestId: the UUID of a request. If the problem persists, you can seek help from OSS development engineers and provide them with the request ID.
- HostId: the ID of a host to identify the accessed OSS cluster, which is the same as the host ID specified in the request.

Error codes

Error code	Error message	HTTP status code
AccessDenied	The error message returned because the access is denied.	403
BucketAlreadyExists	The error message returned because the bucket already exists.	409
BucketNotEmpty	The error message returned because the bucket is not empty.	409
EntityTooLarge	The error message returned because the entity exceeds the maximum allowed size.	400
EntityTooSmall	The error message returned because the entity is smaller than the minimum allowed size.	400
FileGroupTooLarge	The error message returned because the object group exceeds the maximum allowed size.	400
FilePartNotExist	The error message returned because the specified part does not exist.	400
FilePartStale	The error message returned because the part has expired.	400

Error code	Error message	HTTP status code
InvalidArgument	The error message returned because the parameter format is invalid.	400
InvalidAccessKeyId	The error message returned because the specified Accesskey ID does not exist.	403
InvalidBucketName	The error message returned because the bucket name is invalid.	400
InvalidDigest	The error message returned because the digest is invalid.	400
InvalidObjectName	The error message returned because the object name is invalid.	400
InvalidPart	The error message returned because the part is invalid.	400
InvalidPartOrder	The error message returned because the part sequence is invalid.	400
InvalidTargetBucketForLogging	The error message returned because the destination bucket specified for logging is invalid.	400
InternalError	The error message returned because an internal OSS error occurs.	500
MalformedXML	The error message returned because the XML format is invalid.	400
MethodNotAllowed	The error message returned because the method is not supported.	405
MissingArgument	The error message returned because the required parameters are not set.	411
MissingContentLength	The error message returned because the content length is not specified.	411
NoSuchBucket	The error message returned because the specified bucket does not exist.	404
NoSuchKey	The error message returned because the specified object does not exist.	404
NoSuchUpload	The error message returned because the specified multipart upload ID does not exist.	404
NotImplemented	The error message returned because the method cannot be implemented.	501
PreconditionFailed	The error message returned because a precondition error occurs.	412
RequestTimeTooSkewed	The error message returned because the difference between the time on the client when the request is sent and the time on the server when the request is received is greater than 15 minutes.	403
RequestTimeout	The error message returned because the request times out.	400
SignatureDoesNotMatch	The error message returned because the signature is invalid.	403
InvalidEncryptionAlgorithmError	The error message returned because the specified encryption algorithm based on entropy encoding is invalid.	400

1.4.1.13. FAQ

This topic lists the causes and solutions for common errors you may encounter when you use OSS SDK for Java.

JAR conflicts

If an error similar to the following is displayed when you use OSS SDK for Java, a JAR conflict is in your project:

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/http/ssl/TrustStrategy
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:268)
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:193)
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS.java:77)
Caused by: java.lang.ClassNotFoundException: org.apache.http.ssl.TrustStrategy
    at java.net.URLClassLoader$1.run (URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run (URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.net.URLClassLoader.findClass (URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:425)
    at sun.misc.Launcher$AppClassLoader.loadClass (Launcher.java:308)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:358)
    ... 3 more
```

Or

```
Exception in thread "main" java.lang.NoSuchFieldError: INSTANCE
    at org.apache.http.impl.io.DefaultHttpRequestWriterFactory.<init> (DefaultHttpRequestWriterFactory.java:52)
    at org.apache.http.impl.io.DefaultHttpRequestWriterFactory.<init> (DefaultHttpRequestWriterFactory.java:56)
    at org.apache.http.impl.io.DefaultHttpRequestWriterFactory.<clinit> (DefaultHttpRequestWriterFactory.java:46)
    at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<init> (ManagedHttpClientConnectionFactory.java:82)
    at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<init> (ManagedHttpClientConnectionFactory.java:95)
    at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<init> (ManagedHttpClientConnectionFactory.java:104)
    at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<clinit> (ManagedHttpClientConnectionFactory.java:62)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager$InternalConnectionFactory.<init> (PoolingHttpClientConnectionManager.java:572)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init> (PoolingHttpClientConnectionManager.java:174)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init> (PoolingHttpClientConnectionManager.java:158)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init> (PoolingHttpClientConnectionManager.java:149)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init> (PoolingHttpClientConnectionManager.java:125)
    at com.aliyun.oss.common.comm.DefaultServiceClient.createHttpClientConnectionManager (DefaultServiceClient.java:237)
    at com.aliyun.oss.common.comm.DefaultServiceClient.<init> (DefaultServiceClient.java:78)
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:268)
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:193)
    at OSSManagerImpl.upload (OSSManagerImpl.java:42)
    at OSSManagerImpl.main (OSSManagerImpl.java:63)
```

Cause: A possible cause is that the OSS Java SDK uses Apache httpclient 4.4.1, whereas your project uses a version of Apache httpclient or commons-httpclient that conflicts with Apache httpclient 4.4.1. To view the JAR package and its version that is used in your project, run the `mvn dependency:tree` command in the directory of the project. The following figure shows that the project uses Apache HttpClient 4.3.

```
[INFO] --- maven-dependency-plugin:2.2:tree (default-cli) @ maven-demo ---
[INFO] com.aliyun.oss:maven-demo:jar:0.1.1-SNAPSHOT
[INFO] +- junit:junit:jar:4.10:test
[INFO] | \- org.hamcrest:hamcrest-core:jar:1.1:test
[INFO] +- org.apache.httpcomponents:httpclient:jar:4.3:compile
[INFO] | +- org.apache.httpcomponents:httpcore:jar:4.3:compile
[INFO] | +- commons-logging:commons-logging:jar:1.1.3:compile
[INFO] | \- commons-codec:commons-codec:jar:1.6:compile
[INFO] \- com.aliyun.oss:aliyun-sdk-oss:jar:2.2.1:compile
[INFO]     +- org.jdom:jdom:jar:1.1:compile
[INFO]     \- net.sf.json-lib:json-lib:jar:jdk15:2.4:compile
[INFO]         +- commons-beanutils:commons-beanutils:jar:1.8.0:compile
[INFO]         +- commons-collections:commons-collections:jar:3.2.1:compile
[INFO]         +- commons-lang:commons-lang:jar:2.5:compile
[INFO]         \- net.sf.ezmorph:ezmorph:jar:1.0.6:compile
```

Solution: Use one of the following methods to resolve JAR conflicts:

- Use consistent versions of Apache HttpClient. If your project uses a version of Apache HttpClient that conflicts with Apache HttpClient 4.4.1, use Apache HttpClient 4.4.1 and delete all other Apache HttpClient version dependencies in the pom.xml file. If your project uses Commons HttpClient, conflicts may also occur. To resolve these conflicts, delete Commons HttpClient.
- Resolve dependency conflicts. If your project is dependent on multiple third-party packages and the third-party packages are dependent on different versions of Apache HttpClient, your project may contain dependency conflicts. Use exclusions to resolve these conflicts. For more information, visit [Maven Guides](#).

OSS SDK for Java is dependent on the following package versions. The solution to conflicts is similar to that for HttpClient.

```
[INFO] com.aliyun.oss:maven-demo:jar:0.1.1-SNAPSHOT
[INFO] +- junit:junit:jar:4.10:test
[INFO] | \- org.hamcrest:hamcrest-core:jar:1.1:test
[INFO] \- com.aliyun.oss:aliyun-sdk-oss:jar:2.2.1:compile
[INFO]     +- org.apache.httpcomponents:httpclient:jar:4.4.1:compile
[INFO]     | +- org.apache.httpcomponents:httpcore:jar:4.4.1:compile
[INFO]     | +- commons-logging:commons-logging:jar:1.2:compile
[INFO]     | \- commons-codec:commons-codec:jar:1.9:compile
[INFO]     +- org.jdom:jdom:jar:1.1:compile
[INFO]     \- net.sf.json-lib:json-lib:jar:jdk15:2.4:compile
[INFO]         +- commons-beanutils:commons-beanutils:jar:1.8.0:compile
[INFO]         +- commons-collections:commons-collections:jar:3.2.1:compile
[INFO]         +- commons-lang:commons-lang:jar:2.5:compile
[INFO]         \- net.sf.ezmorph:ezmorph:jar:1.0.6:compile
```

Missing packages

An error similar to the following is displayed when you use OSS SDK for Java:

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/http/auth/Credentials
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:268)
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:193)
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS.java:76)
Caused by: java.lang.ClassNotFoundException: org.apache.http.auth.Credentials
    at java.net.URLClassLoader$1.run (URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run (URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.net.URLClassLoader.findClass (URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:425)
    at sun.misc.Launcher$AppClassLoader.loadClass (Launcher.java:308)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:358)
    ... 3 more
```

Or

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/http/protocol/HttpContext
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:268)
    at com.aliyun.oss.OSSClient.<init> (OSSClient.java:193)
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS.java:76)
Caused by: java.lang.ClassNotFoundException: org.apache.http.protocol.HttpContext
    at java.net.URLClassLoader$1.run (URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run (URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.net.URLClassLoader.findClass (URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:425)
    at sun.misc.Launcher$AppClassLoader.loadClass (Launcher.java:308)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:358)
    ... 3 more
```

Or

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/jdom/input/SAXBuilder
    at com.aliyun.oss.internal.ResponseParsers.getXmlRootElement (ResponseParsers.java:645)
    at ...
    at com.aliyun.oss.OSSClient.doesBucketExist (OSSClient.java:471)
    at com.aliyun.oss.OSSClient.doesBucketExist (OSSClient.java:465)
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS.java:82)
Caused by: java.lang.ClassNotFoundException: org.jdom.input.SAXBuilder
    at java.net.URLClassLoader$1.run (URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run (URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.net.URLClassLoader.findClass (URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:425)
    at sun.misc.Launcher$AppClassLoader.loadClass (Launcher.java:308)
    at java.lang.ClassLoader.loadClass (ClassLoader.java:358)
    ... 11 more
```

Cause: The JAR package used to compile or run OSS SDK for Java may be missing in your project. OSS SDK for Java depends on the following JAR packages:

- aliyun-sdk-oss-2.2.1.jar
- hamcrest-core-1.1.jar
- jdom-1.1.jar
- commons-codec-1.9.jar
- httpclient-4.4.1.jar
- commons-logging-1.2.jar
- httpcore-4.4.1.jar
- log4j-1.2.15.jar

All packages except the log4j-1.2.15.jar package are required for OSS Java SDK. However, if you want to enable logging, you must also include the log4j-1.2.15.jar package.

Solution: Add packages on which OSS SDK for Java is dependent to your projects by using the following methods:

- If you run your project in Eclipse, import dependencies to Eclipse, as instructed in the "Method 2: Import a JAR package" section of [Installation](#).
- If you run your project in Ant, add the packages on which OSS SDK for Java is dependent to the lib directory of the project.
- If you use the `javac` or `java` file, use the `-classpath` or `-cp` option to specify the path in which OSS Java SDK dependent packages are added, or save the Java SDK dependencies to classpath.

Connection timeout

If a similar exception is displayed when you run OSS SDK for Java:

```
com.aliyun.oss.ClientException: SocketException
    at com.aliyun.oss.common.utils.ExceptionFactory.createNetworkException (ExceptionFactory.java:71)
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore (DefaultServiceClient.java:116)
    at com.aliyun.oss.common.comm.ServiceClient.sendRequestImpl (ServiceClient.java:121)
    at com.aliyun.oss.common.comm.ServiceClient.sendRequest (ServiceClient.java:67)
    at com.aliyun.oss.internal.OSSOperation.send (OSSOperation.java:92)
    at com.aliyun.oss.internal.OSSOperation.doOperation (OSSOperation.java:140)
    at com.aliyun.oss.internal.OSSOperation.doOperation (OSSOperation.java:111)
    at com.aliyun.oss.internal.OSSBucketOperation.getBucketInfo (OSSBucketOperation.java:1152)
    at com.aliyun.oss.OSSClient.getBucketInfo (OSSClient.java:1220)
    at com.aliyun.oss.OSSClient.getBucketInfo (OSSClient.java:1214)
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS.java:94)
Caused by: org.apache.http.conn.HttpHostConnectException: Connect to oss-test.oss-cn-hangzhou-internal.aliyuncs.com:80 [oss-test.oss-cn-hangzhou-internal.aliyuncs.com/10.84.135.99] failed: Connection timed out: connect
    at org.apache.http.impl.conn.DefaultHttpClientConnectionOperator.connect (DefaultHttpClientConnectionOperator.java:151)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.connect (PoolingHttpClientConnectionManager.java:353)
    at org.apache.http.impl.execchain.MainClientExec.establishRoute (MainClientExec.java:380)
    at org.apache.http.impl.execchain.MainClientExec.execute (MainClientExec.java:236)
    at org.apache.http.impl.execchain.ProtocolExec.execute (ProtocolExec.java:184)
    at org.apache.http.impl.execchain.RedirectExec.execute (RedirectExec.java:110)
    at org.apache.http.impl.client.InternalHttpClient.doExecute (InternalHttpClient.java:184)
    at org.apache.http.impl.client.CloseableHttpClient.execute (CloseableHttpClient.java:82)
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore (DefaultServiceClient.java:113)
    ... 9 more
```

Cause: The endpoint is incorrect or the network is disconnected.

Solution: Solve the network problem and try again.

org.apache.http.NoHttpResponseException: The target server failed to respond

If a similar exception is displayed when you run OSS SDK for Java:

```
com.aliyun.oss.ClientException: unknown
    at com.aliyun.oss.common.utils.ExceptionFactory.createNetworkException(ExceptionFactory.java:68) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore(DefaultServiceClient.java:115) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.common.comm.ServiceClient.sendRequestImpl(ServiceClient.java:121) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.common.comm.ServiceClient.sendRequest(ServiceClient.java:67) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSOperation.send(OSSOperation.java:92) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSOperation.doOperation(OSSOperation.java:140) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSOperation.doOperation(OSSOperation.java:111) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSMultiPartOperation.initiateMultipartUpload(OSSMultiPartOperation.java:206) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.OSSClient.initiateMultipartUpload(OSSClient.java:765) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.taobao.agoo.dump.client.OssTools.multipartUpload(OssTools.java:79) ~[agoo-dump-client-2.0.0-SNAPSHOT.jar:na]
    at com.taobao.agoo.dump.biz.manager.TaskExecutorManager$UploadTask.run(TaskExecutorManager.java:114) ~[agoo-dump-biz-2.0.0-SNAPSHOT.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471) [na:1.7.0_51]
    at java.util.concurrent.FutureTask.run(FutureTask.java:262) [na:1.7.0_51]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:615) [na:1.7.0_51]
    at java.lang.Thread.run(Thread.java:744) [na:1.7.0_51]
Caused by: org.apache.http.NoHttpResponseException: The target server failed to respond
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:143) ~[httpclient-4.4.jar:4.4]
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:57) ~[httpclient-4.4.jar:4.4]
    at org.apache.http.impl.io.AbstractMessageParser.parse(AbstractMessageParser.java:261) ~[httpcore-4.4.jar:4.4]
    at org.apache.http.impl.defaultBHttpClientConnection.receiveResponseHeader(DefaultBHttpClientConnection.java:165) ~[httpcore-4.4.jar:4.4]
    at org.apache.http.impl.conn.CPoolProxy.receiveResponseHeader(CPoolProxy.java:167) ~[httpclient-4.4.jar:4.4]
    at org.apache.http.protocol.HttpRequestExecutor.doReceiveResponse(HttpRequestExecutor.java:272) ~[httpcore-4.4.jar:4.4]
```

Cause: The preceding error may occur if expired connections are used. This error occurs only when a Java SDK version earlier than 2.1.2 is used.

Solution: Upgrade OSS SDK for Java to V2.1.2 or later.

OSS SDK for Java stops responding when OSS SDK for Java is called

If OSS SDK for Java stops responding when you call OSS SDK for Java, call `jstack -l <pid>` to view the stack. The similar error is displayed at the following positions:

```
"main" prio=6 tid=0x00000000291e000 nid=0xc40 waiting on condition [0x000000002dae000]
java.lang.Thread.State: WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <0x000000007d85697f8> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:186)
    at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2043)
    at org.apache.http.pool.PoolEntryFuture.await(PoolEntryFuture.java:138)
    at org.apache.http.pool.AbstractConnPool.getPoolEntryBlocking(AbstractConnPool.java:306)
    at org.apache.http.pool.AbstractConnPool.access$000(AbstractConnPool.java:64)
    at org.apache.http.pool.AbstractConnPool$2.getPoolEntry(AbstractConnPool.java:192)
    at org.apache.http.pool.AbstractConnPool$2.getPoolEntry(AbstractConnPool.java:185)
    at org.apache.http.pool.PoolEntryFuture.get(PoolEntryFuture.java:107)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.leaseConnection(PoolingHttpClientConnectionManager.java:276)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager$1.get(PoolingHttpClientConnectionManager.java:263)
    at org.apache.http.impl.execchain.MainClientExec.execute(MainClientExec.java:190)
    at org.apache.http.impl.execchain.ProtocolExec.execute(ProtocolExec.java:184)
    at org.apache.http.impl.execchain.RedirectExec.execute(RedirectExec.java:110)
    at org.apache.http.impl.client.InternalHttpClient.doExecute(InternalHttpClient.java:184)
    at org.apache.http.impl.client.CloseableHttpClient.execute(CloseableHttpClient.java:82)
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore(DefaultServiceClient.java:113)
    at com.aliyun.oss.common.comm.ServiceClient.sendRequestImpl(ServiceClient.java:123)
    at com.aliyun.oss.common.comm.ServiceClient.sendRequest(ServiceClient.java:68)
    at com.aliyun.oss.internal.OSSOperation.send(OSSOperation.java:94)
    at com.aliyun.oss.internal.OSSOperation.doOperation(OSSOperation.java:146)
    at com.aliyun.oss.internal.OSSOperation.doOperation(OSSOperation.java:113)
    at com.aliyun.oss.internal.OSSObjectOperation.getObject(OSSObjectOperation.java:229)
    at com.aliyun.oss.OSSClient.getObject(OSSClient.java:629)
    at com.aliyun.oss.OSSClient.getObject(OSSClient.java:617)
    at samples.HelloOSS.main(HelloOSS.java:49)
```

Cause: Connection leaks occur in the connection pool. It may be because `OSSClient.getObject` is not closed after use.

Solution: Check your program and ensure that no connection leaks occur. The following code provides an example on how to close `ossObject`:

```
// Read an object.
OSSObject ossObject = ossClient.getObject(bucketName, objectName);
// Perform operations in OSS.
// Close your ossObject.
ossObject.close();
```

Connection closure

An error similar to the following is displayed when you use `ossClient.getObject`:

```
Exception in thread "main" org.apache.http.ConnectionClosedException: Premature end of Content-Length delimited message body (expected: 11990526; received: 202880)
    at org.apache.http.impl.io.ContentLengthInputStream.read(ContentLengthInputStream.java:180)
    at org.apache.http.impl.io.ContentLengthInputStream.read(ContentLengthInputStream.java:200)
    at org.apache.http.impl.io.ContentLengthInputStream.close(ContentLengthInputStream.java:103)
    at org.apache.http.impl.execchain.ResponseEntityProxy.streamClosed(ResponseEntityProxy.java:128)
    at org.apache.http.conn.EofSensorInputStream.checkClose(EofSensorInputStream.java:228)
    at org.apache.http.conn.EofSensorInputStream.close(EofSensorInputStream.java:174)
    at java.io.FilterInputStream.close(FilterInputStream.java:181)
    at java.io.FilterInputStream.close(FilterInputStream.java:181)
    at com.aliyun.oss.event.ProgressInputStream.close(ProgressInputStream.java:147)
    at java.io.FilterInputStream.close(FilterInputStream.java:181)
    at samples.HelloOSS.main(HelloOSS.java:39)
```

Cause: The interval between the two data reading attempts exceeded one minute. OSS closes the connection if no data is sent or received after more than one minute. This error occurs when your application functions as follows and the time for the application to process data varies: Read part of the data, process the data, read the data, and then process the data.

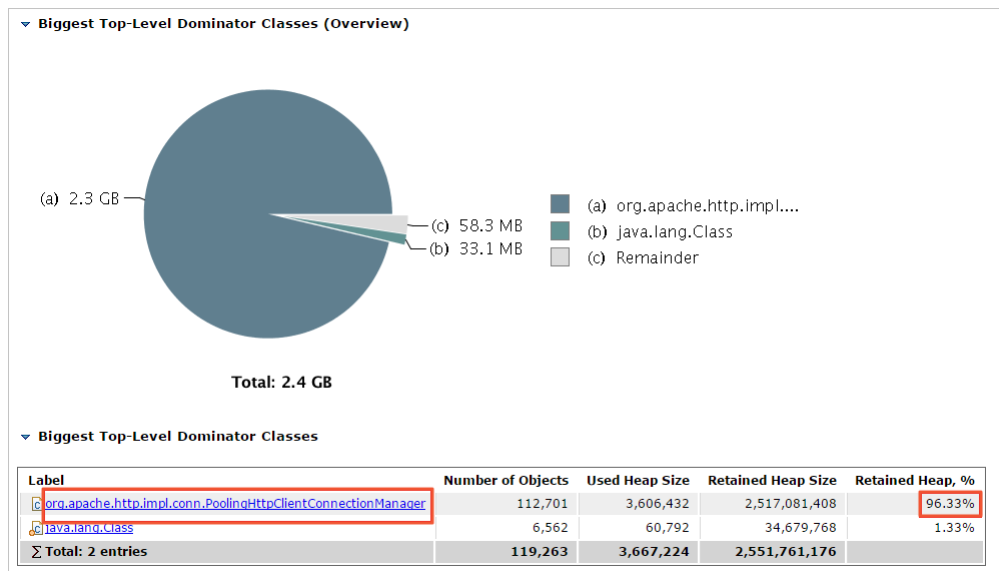
Solution: If you need to read part of the data during unspecified time periods, you can use range download to avoid connection closure during data reading.

Memory leaks

Memory leaks occur when OSS SDK for Java program runs for an extended period of time. This period ranges from several hours to several days based on the amount of access traffic.

Cause: Use the memory analysis tool to analyze the memory usage. We recommend that you use [Eclipse Memory Analyzer \(MAT\)](#) to analyze memory usage. For more information, visit [Use MAT for Heap Dump File Analysis](#).

If the analysis result is similar to that shown in the following figure, PoolingHttpClientConnectionManager takes up 96% of memory. The cause is that the new OSSClient command is run multiple times while ossClient.shutdown is not called. As a result, memory leaks occur.



Solution: You must ensure that `ossClient.shutdown` is called after you run the new `OSSClient` command.

InterruptedException occurs when OSSClient.shutdown is called

A similar exception is displayed when `OSSClient.shutdown` is called:

```
java.lang.InterruptedException: sleep interrupted
    at java.lang.Thread.sleep(Native Method)
    at com.aliyun.oss.common.comm.IdleConnectionReaper.run(IdleConnectionReaper:76)
```

Cause: A background thread `IdleConnectionReaper` from `OSSClient` shuts down idle connections in a timely manner. If the shutdown method is called when the `IdleConnectionReaper` thread is asleep, the preceding exception is displayed.

Solution:

- Upgrade OSS SDK for Java to version 2.3.0 or later because this problem is fixed in OSS SDK for Java V2.3.0.
- For versions earlier than OSS Java SDK version 2.3.0, use the following code to ignore the exception:

```
try {
    serviceClient.shutdown();
} catch (Exception e) {
}
```

1.4.2. Python-SDK

1.4.2.1. Preface

This topic describes the sample code of OSS SDK for Python in various use cases.

Source code

For the source code of OSS SDK for Python, visit [GitHub](#).

Sample programs

The sample code of OSS SDK for Python is in the examples directory. For more information, visit [GitHub](#). The following table describes the sample files in the sample programs.

Sample file	Description
object_basic.py	Shows you how to perform basic object-related operations such as object upload, download, list, and deletion.

Sample file	Description
object_extra.py	Shows you how to perform advanced object-related operations such as configure the language and user metadata, copy objects, and append objects.
upload.py	Shows you how to perform advanced object upload operations such as resumable upload and multipart upload.
download.py	Shows you how to perform object download operations such as object download, range download, and resumable download.
object_check.py	Shows you how to verify data during upload and download, including MD5 verification and CRC.
object_progress.py	Shows you how to use a progress bar during upload and download.
object_callback.py	Shows you how to perform upload callback.
object_post.py	Shows you how to use PostObject. The PostObject implementation is independent of the SDK for Python.
sts.py	Shows you how to use STS, including obtain the temporary AccessKey pair by assuming a role and use the temporary AccessKey pair to access OSS.
live_channel.py	Shows you how to use LiveChannel.
image.py	Shows you how to use Image Processing (IMG).
bucket.py	Shows you how to manage buckets such as bucket creation, deletion, and listing.

1.4.2.2. Installation

This topic describes how to install OSS SDK for Python.

Preparation

The version of OSS SDK for Python in this topic is compatible with Python 2.6, 2.7, 3.3, 3.4, 3.5, and 3.6.

- Install the environment

For more information about how to install an appropriate version of Python, visit [python](#).

- Check the Python version

Run the `python -V` command to check Python version.

In Windows, if a message is displayed, indicating that it is not an internal or external command, edit the environment variable `Path`. Add the installation path of Python and pip. The installation path of pip is the Scripts directory under the installation path of Python.

Note You may need to restart your computer to ensure that the environment variables take effect.

Download OSS SDK for Python

- [Download from GitHub](#)
- [Download previous versions](#)

For more information, see [API Reference](#).

Install python-devel

OSS SDK for Python uses the `crcmod` library to calculate the value of CRC. The `crcmod` library depends on the `Python.h` file. If the `Python.h` file is missing, SDK installation is not affected but the installation of `crcmod` extensions in C fails. Therefore, the efficiency of operations such as object upload and download is low. Install `python-devel` if it does not exist.

- When you install Python in Windows or Mac OS X systems, the `Python.h` file is installed together with Python. Therefore, you do not need to install `python-devel`.
- When you install Python in CentOS, RHEL, or Fedora systems, run the following command to install `python-devel`:

```
yum install python-devel
```

- When you install Python in Debian or Ubuntu systems, run the following command to install `python-devel`:

```
apt-get install python-dev
```

Install OSS SDK for Python

- Use pip

Run the following command:

```
pip install oss2
```

Note If pip is not installed in your environment, visit the [official pip website](#) to install pip.

- Use the source code

Click [here](#) to visit GitHub and download the OSS SDK for Python package of an appropriate version. Decompress the package. Check whether the `setup.py` file exists in the directory and then run the following command:

```
python setup.py install
```

Verification

- Verify the version of OSS SDK for Python
 - Enter `python` in the command line and press Enter to go to the Python environment.
 - Run the following command to verify the version of OSS SDK for Python:

```
>>> import oss2
>>> oss2.__version__
'2.0.0'
```

If an output similar to the preceding example is displayed, OSS SDK for Python version 2.0.0 is installed.

- Verify `crcmod`

OSS SDK for Python uses the following methods to calculate the value of CRC: the C extension mode, which uses the `crcmod` dependent library to call the `_crcfunext.so` extension library to calculate the CRC value, and the Python-only mode. The performance of the C extension mode is superior than the Python-only mode. The `crcmod` library is automatically installed together with `oss2`. For more information about `crcmod`, visit [CRCMOD introduction](#).

When you use OSS SDK for Python, if the object upload and download efficiency remains much lower than that of other tools or other SDKs, the installation of `crcmod` extensions in C may fail. In this case, the Python-only mode is used to calculate the value of CRC for object upload and download.

You can use the following methods to determine whether the C extension mode of the `crcmod` library is installed:

- Enter `python` in the command line and press the Enter key to go the Python environment.
- Enter `import crcmod._crcfunext` and press Enter.
 - If no error message is displayed, the `crcmod` extensions in C are installed.
 - If the following error message is displayed, the installation of `crcmod` extensions in C failed:

```
>>> import crcmod._crcfunext
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named _crcfunext
```

When you compile the `crcmod` library, the `_crcfunext.so` file depends on the `Python.h` file. The error occurs because the `Python.h` file does not exist in the system, which causes the `_crcfunext.so` file failed to be generated. Therefore, the Python-only mode is used to calculate the value of CRC. OSS SDK for Python is installed, but the efficiency of operations such as object upload and download is low.

If the error occurs, perform the following operations:

- Run the following command to uninstall the `crcmod` library:

```
pip uninstall crcmod
```

- Install `python-devel`.

- Run the following command to re-install the `crcmod` library:

```
pip install crcmod
```

If the `crcmod` library fails to be installed after you perform the preceding operations, uninstall the `crcmod` library. Run the following command to view the detailed causes of the installation failures.

```
pip install crcmod -v
```

Uninstall OSS SDK for Python

If OSS SDK for Python fails to be installed, uninstall OSS SDK for Python by using `pip` and reinstall OSS SDK for Python. Run the following code to uninstall OSS SDK for Python:

```
pip uninstall oss2
```

1.4.2.3. Getting started

This topic describes how to use Object Storage Service (OSS) SDK for Python to perform routine operations, such as creating buckets, uploading objects, and downloading objects.

Create a bucket

A bucket is a global namespace in OSS. A bucket is a container that is used to store files.

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set the access control list (ACL) of the bucket to private.
bucket.create_bucket(oss2.models.BUCKET_ACL_PRIVATE)
```

For more information about how to obtain an AccessKey ID and AccessKey secret, see [Obtain an AccessKey pair](#).

For more information about how to obtain an endpoint, see [Obtain an endpoint](#).

Upload objects

The following code provides an example on how to upload objects to OSS by using stream upload:


```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# <yourObjectName> indicates the full path of the object that you want to upload to OSS. The full path of the object must include the extension of the
object name. Example: abc/efg/123.jpg.
# <yourLocalFile> consists of a local file path and a file name with an extension. Example: /users/local/myfile.txt.
bucket.put_object_from_file('<yourObjectName>', '<yourLocalFile>')
```

Download objects

The following code provides an example on how to download a specified object and save the object as a local file:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# <yourObjectName> indicates the full path of the object that you want to download from OSS. The full path of the object must include the extension of
the object name. Example: abc/efg/123.jpg.
# <yourLocalFile> consists of a local file path and a file name with an extension. Example: /users/local/myfile.txt.
bucket.get_object_to_file('<yourObjectName>', '<yourLocalFile>')
```

List objects

The following code provides an example on how to list 10 objects in a specified bucket:

```
# -*- coding: utf-8 -*-
import oss2
from itertools import islice
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# oss2.ObjectIterator is used to traverse objects.
for b in islice(oss2.ObjectIterator(bucket), 10):
    print(b.key)
```

Delete objects

The following code provides an example on how to delete an object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# <yourObjectName> indicates the full path of the object that you want to delete from OSS. The full path of the object must include the extension of
the object name. Example: abc/efg/123.jpg.
bucket.delete_object('<yourObjectName>')
```

1.4.2.4. Initialization

This topic describes how to initialize OSS SDK for Python.

Most operations of OSS SDK for Python are performed based on the `oss2.Service` and `oss2.Bucket` classes.

- `oss2.Service` is used to list buckets.
- `oss2.Bucket` is used to upload, download, and delete objects as well as configure buckets.

You must specify the endpoint when you initiate the two classes. `oss2.Service` does not support access to buckets by using custom domain names that are mapped to the buckets. For more information about endpoints, see [Obtain an endpoint](#).

Initialize `oss2.Service`

For more information, see [List buckets](#).

Initialize `oss2.Bucket`

- Use the endpoint of the region where the bucket is located to initialize the class

The following code provides an example on how to initialize `oss2.Bucket` by using the endpoint of the region where the bucket is located:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
endpoint = 'https://oss-cn-hangzhou.aliyuncs.com'
bucket = oss2.Bucket(auth, endpoint, '<yourBucketName>')
```

- Configure the connection timeout period

The following code provides an example on how to configure the connection timeout period:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
endpoint = 'https://oss-cn-hangzhou.aliyuncs.com'
# Set the connection timeout period to 30 seconds.
bucket = oss2.Bucket(auth, endpoint, '<yourBucketName>', connect_timeout=30)
```

- **Disable CRC**

CRC is enabled by default when you upload or download objects to ensure data integrity. The following code provides an example on how to disable CRC:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
endpoint = 'https://oss-cn-hangzhou.aliyuncs.com'
bucket = oss2.Bucket(auth, endpoint, '<yourBucketName>', enable_crc=False)
```

 **Warning** We recommend that you do not disable CRC. If you disable CRC, data integrity may be affected during object uploads and downloads.

1.4.2.5. Buckets

1.4.2.5.1. Create a bucket

A bucket is a container used to store objects in Object Storage Service (OSS). Each object is contained in a bucket. This topic describes how to create a bucket.

The following code provides an example on how to create a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# After you specify the endpoint and bucket name, you can create a bucket in the specified region. In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Create the bucket. By default, the ACL of the created bucket is private.
bucket.create_bucket()
```

1.4.2.5.2. List buckets

A bucket is a container that is used to store objects in Object Storage Service (OSS). All objects in OSS are contained in buckets. Bucket names are listed in alphabetical order. You can list buckets that belong to the current Alibaba Cloud account in all regions and meet the specific conditions.

The following sample code provides an example on how to list all buckets that belong to an Alibaba Cloud account:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
service = oss2.Service(auth, 'http://oss-cn-hangzhou.aliyuncs.com')
print([b.name for b in oss2.BucketIterator(service)])
```

1.4.2.5.3. Determine whether a bucket exists

A bucket is a container used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. This topic describes how to determine whether a bucket exists.

The following code provides an example on how to determine whether a bucket exists:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
def does_bucket_exist(bucket):
    try:
        bucket.get_bucket_info()
    except oss2.exceptions.NoSuchBucket:
        return False
    except:
        raise
    return True
exist = does_bucket_exist(bucket)
# If the returned value is true, a bucket with the specified name exists. If the returned value is false, a bucket with the specified name does not exist.
if exist:
    print('bucket exist')
else:
    print('bucket not exist')
```

1.4.2.5.4. Manage bucket ACLs

A bucket is a container used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. This topic describes how to configure and query the access control list (ACL) of a bucket.

Configure the ACL of a bucket

The following table describes the ACLs that you can configure for a bucket.

ACL	Description	Value
Private	Only the bucket owner and authorized users have read and write permissions on objects in the bucket. Other users cannot access the objects in the bucket.	oss2.BUCKET_ACL_PRIVATE
Public read	Only the bucket owner and authorized users have read and write permissions on objects in the bucket. Other users have only read permissions on the objects in the bucket. Exercise caution when you set the ACL of the bucket to this value.	oss2.BUCKET_ACL_PUBLIC_READ
Public read/write	All users have read and write permissions on objects in the bucket. Exercise caution when you set the ACL of the bucket to this value.	oss2.BUCKET_ACL_PUBLIC_READ_WRITE

The following code provides an example on how to configure the ACL of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set the ACL of the bucket to private.
bucket.put_bucket_acl(oss2.BUCKET_ACL_PRIVATE)
```

Query the ACL of a bucket

The following code provides an example on how to query the ACL of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
print(bucket.get_bucket_acl().acl)
```

1.4.2.5.5. Bucket tagging

You can configure bucket tagging to classify and manage buckets in Object Storage Service (OSS). For example, you can configure that only buckets with specific tags are listed in ListBucket operations.

Note

- Only the bucket owner or authorized Resource Access Management (RAM) users can configure tagging for the bucket. Otherwise, 403 Forbidden is returned with the error code AccessDenied.
- You can configure up to 20 tags for a bucket. A tag is a key-value pair.
- The key of a tag can be up to 64 bytes in length and cannot start with `http://`, `https://`, or `Aliyun`. The key of a tag cannot be empty.
- The tag value is optional. The tag value can be up to 128 bytes in length. You can leave the parameter empty.
- The key and value of a tag must be UTF-8-encoded.
- When you use PutBucketTags to configure a tag for a bucket, the new tag overwrites the existing tag.

Configure tags for a bucket

The following code provides an example on how to configure tags for a bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import Tagging, TaggingRule
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Create a tagging rule.
rule = TaggingRule()
rule.add('key1', 'value1')
rule.add('key2', 'value2')
# Create tags.
tagging = Tagging(rule)
# Configure tags for the bucket.
result = bucket.put_bucket_tagging(tagging)
# Display the HTTP status code.
print('http status:', result.status)
```

Query the tags of a bucket

The following code provides an example on how to query the tags of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Query the tags of the bucket.
result = bucket.get_bucket_tagging()
# Display the tagging rule.
tag_rule = result.tag_set.tagging_rule
print('tag rule:', tag_rule)
```

List buckets that have specified tags

The following code provides an example on how to list buckets that have specified tags:

```
# -*- coding: utf-8 -*-
import oss2
# Create a server object.
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
service = oss2.Service(auth, 'http://oss-cn-hangzhou.aliyuncs.com')
# Add the tag-key and tag-value fields to the params parameter of the ListBuckets API operation.
params = {}
params['tag-key'] = '<yourTagging_key>'
params['tag-value'] = '<yourTagging_value>'
# List buckets that have the specified tags
result = service.list_buckets(params=params)
# Display the query results.
for bucket in result.buckets:
    print('result bucket_name:', bucket.name)
```

Delete the tags of a bucket

The following code provides an example on how to delete the tags of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Delete the tags of the bucket.
result = bucket.delete_bucket_tagging()
# Display the HTTP status code.
print('http status:', result.status)
```

1.4.2.5.6. Delete buckets

If you no longer use a bucket, delete the bucket to stop unnecessary charges.

Before you delete a bucket, ensure that all objects, LiveChannels, and parts that are generated during multipart upload in the bucket are deleted. The following code provides an example on how to delete buckets:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
try:
    # Delete the bucket.
    bucket.delete_bucket()
except oss2.exceptions.BucketNotEmpty:
    print('bucket is not empty.')
except oss2.exceptions.NoSuchBucket:
    print('bucket does not exist')
```

To delete a bucket that is not empty, you must clear the bucket by listing and deleting objects in the bucket. If a multipart upload task is ongoing, you must terminate the task before you can delete the bucket.

1.4.2.5.7. Manage lifecycle rules

Object Storage Service (OSS) allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to Infrequent Access (IA) or Archive to reduce your storage costs. This topic describes how to manage lifecycle rules for buckets.

Background

Each lifecycle rule contains the following information:

- Policy: specifies the mode to match objects and parts.
 - Match by prefix: matches objects and parts by prefix. You can create multiple rules to configure different prefixes. Each prefix must be unique.
 - Match by bucket: matches all objects and parts contained in the bucket. You can create only one rule if you select this method.
- Object lifecycle policy: specifies the validity period or expiration data and the operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired objects are deleted.
 - Expiration date: specifies a date. Objects that are last modified before this date expire and are deleted.
- Part lifecycle policy: specifies the validity period or expiration time and the delete operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted.
 - Expiration date: specifies a date. Parts that are last modified before this date expire and are deleted.

Lifecycle rules also apply to the parts uploaded by using uploadPart. In this case, the last modified time of an object is the time the multipart upload task is initiated.

Configure lifecycle rules for a bucket

The following code provides an example on how to configure lifecycle rules for a bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import LifecycleExpiration, LifecycleRule, BucketLifecycle, AbortMultipartUpload
import datetime
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Specify that objects expire three days after they are last modified.
rule1 = LifecycleRule('rule1', 'tests/',
    status=LifecycleRule.ENABLED,
    expiration=LifecycleExpiration(days=3))
# Specify an expiration date. Objects that are created before this date expires.
rule2 = LifecycleRule('rule2', 'logging-',
    status=LifecycleRule.DISABLED,
    expiration=LifecycleExpiration(created_before_date=datetime.date(2018, 12, 12)))
# Specify that parts expire three days after they are last modified.
rule3 = LifecycleRule('rule3', 'tests1/',
    status=LifecycleRule.ENABLED,
    abort_multipart_upload=AbortMultipartUpload(days=3))
# Specify an expiration date. Parts that are last modified before this date expires.
rule4 = LifecycleRule('rule4', 'logging1-',
    status=LifecycleRule.DISABLED,
    abort_multipart_upload = AbortMultipartUpload(created_before_date=datetime.date(2018, 12, 12)))
lifecycle = BucketLifecycle([rule1, rule2, rule3, rule4])
bucket.put_bucket_lifecycle(lifecycle)
```

Query the lifecycle rules of a bucket

The following code provides an example on how to query the lifecycle rules of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
lifecycle = bucket.get_bucket_lifecycle()
for rule in lifecycle.rules:
    if rule.abort_multipart_upload is None:
        print('id={0}, prefix={1}, status={2}, days={3}, created_before_date={4}'
              .format(rule.id, rule.prefix, rule.status,
                      rule.expiration.days,
                      rule.expiration.created_before_date))
    else:
        print('id={0}, prefix={1}, status={2}, days={3}, created_before_date={4}'
              .format(rule.id, rule.prefix, rule.status,
                      rule.abort_multipart_upload.days,
                      rule.abort_multipart_upload.created_before_date))
```

Clear the lifecycle rules of a bucket

The following code provides an example on how to clear the lifecycle rules of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.delete_bucket_lifecycle()
# An exception occurs when you query the lifecycle rules of the bucket again.
try:
    lifecycle = bucket.get_bucket_lifecycle()
except oss2.exceptions.NoSuchLifecycle:
    print('lifecycle is not configured')
```

1.4.2.5.8. Static website hosting

You can enable static website hosting for your bucket. After you enable static website hosting for a bucket, you can visit the website by directly accessing the bucket domain name. In this case, you are redirected to the index page or error page that are specified for the hosted website.

Configure static website hosting

The following code provides an example on how to configure static website hosting for a bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketWebsite
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Enable static website hosting for the bucket. Set the index page to index.html and the error page to error.html.
bucket.put_bucket_website(BucketWebsite('index.html', 'error.html'))
```

Query the static website hosting configurations of a bucket

The following code provides an example on how to query the static website hosting configurations of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
try:
    # If static website hosting is not enabled for the bucket, OSS returns NoSuchWebsite for the get_bucket_website method.
    website = bucket.get_bucket_website()
    print('Index file is {0}, error file is {1}'.format(website.index_file, website.error_file))
except oss2.exceptions.NoSuchWebsite as e:
    print('Website is not configured, request_id={0}'.format(e.request_id))
```

Delete the static website hosting configurations of a bucket

The following code provides an example on how to delete the static website hosting configurations of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.delete_bucket_website()
```

1.4.2.5.9. CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to different regions. Object Storage Service (OSS) provides CORS API operations to control cross-origin access.

Configure CORS rules

The following code provides an example on how to configure CORS rules for a specific bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketCors, CorsRule
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
rule = CorsRule(allowed_origins=['*'],
                 allowed_methods=['GET', 'HEAD'],
                 allowed_headers=['*'],
                 max_age_seconds=1000)
# The existing rules are overwritten.
bucket.put_bucket_cors(BucketCors([rule]))
```

Query CORS rules

The following code provides an example on how to query the CORS rules that are configured for a specific bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
try:
    cors = bucket.get_bucket_cors()
except oss2.exceptions.NoSuchCors:
    print('cors is not set')
else:
    for rule in cors.rules:
        print('AllowedOrigins={0}'.format(rule.allowed_origins))
        print('AllowedMethods={0}'.format(rule.allowed_methods))
        print('AllowedHeaders={0}'.format(rule.allowed_headers))
        print('ExposeHeaders={0}'.format(rule.expose_headers))
        print('MaxAgeSeconds={0}'.format(rule.max_age_seconds))
```

Delete CORS rules

The following code provides an example on how to delete the CORS rules that are configured for a specific bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.delete_bucket_cors()
```

1.4.2.5.10. Hotlink protection

Object Storage Service (OSS) provides hotlink protection to prevent unauthorized domain names from accessing your OSS data. You can specify the Referer field in HTTP request headers to configure hotlink protection.

To configure hotlink protection, you must configure the following parameters:

- **Referer Whitelist**: specifies that only specified domain names are allowed to access your resources.
- **Allow Empty Referer**: determines whether requests that contain an empty Referer field are allowed. If you specify that an empty Referer field is not allowed, only HTTP and HTTPS requests that contain an allowed Referer field can access your OSS resources.

Configure hotlink protection for a bucket

The following code provides an example on how to configure a Referer whitelist for a bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketReferer
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set BucketReferer to True. True indicates that empty Referer fields are allowed and False indicates that empty Referer fields are not allowed. Then
, configure the Referer whitelist.
bucket.put_bucket_referer(BucketReferer(True, ['http://aliyun.com', 'http://*.aliyuncs.com']))
```

Query the hotlink protection configurations of a bucket

The following code provides an example on how to query the hotlink protection configuration of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
config = bucket.get_bucket_referer()
print('allow empty referer={0}, referers={1}'.format(config.allow_empty_referer, config.referers))
```

Delete the hotlink protection configurations of a bucket

The following code provides an example on how to delete the hotlink protection configurations of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# The Referer whitelist of a bucket cannot be directly cleared. You must create a Referer whitelist that allows empty Referer fields to overwrite the existing Referer whitelist.
bucket.put_bucket_referer(BucketReferer(True, []))
```

1.4.2.5.11. Configure logging

You can enable logging to record access to a bucket in log objects, which are stored in a specified bucket.

The names of log objects are in the following format: `<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString`.

Enable logging for a bucket

The following code provides an example on how to enable logging for a bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketLogging
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Enable logging for the bucket. Save the log objects to the current bucket, and set the directory in which the log objects are stored to 'logging/'.

logging = bucket.put_bucket_logging(BucketLogging(bucket.bucket_name, 'logging/'))
if logging.status == 200:
    print("Enable access logging")
else:
    print("request_id :", logging.request_id)
    print("resp : ", logging.resp.response)
```

Query the logging configurations of a bucket

The following code provides an example on how to query the logging configurations of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
logging = bucket.get_bucket_logging()
print('TargetBucket={0}, TargetPrefix={1}'.format(logging.target_bucket, logging.target_prefix))
```

Disable logging for a bucket

The following code provides an example on how to disable logging for a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
logging = bucket.delete_bucket_logging()
if logging.status == 204:
    print("Disable access logging")
else:
    print("request_id :", logging.request_id)
    print("resp : ", logging.resp.response)
```

1.4.2.6. Upload objects

1.4.2.6.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for Python provides a variety of methods to upload objects.

OSS SDK for Python supports the following object upload methods:

- **Simple upload:** uploads an object up to 5 GB in size.
- **Append upload:** uploads an object up to 5 GB in size.
- **Resumable upload:** supports concurrent upload and resumable upload. You can define the size of each part. This method is suitable for the upload of large objects. You can use this method to upload an object up to 48.8 TB in size.
- **Multipart upload:** uploads an object up to 48.8 TB in size. This method is suitable for the upload of large objects.

During object upload, you can configure object metadata and view upload progress by using a progress bar. For more information, see [Use progress bars](#).

1.4.2.6.2. Simple upload

You can use simple upload to upload the following types of data: strings, bytes, Unicode characters, network streams, and local files.

You can use the `bucket.put_object` method to upload an object. The following table describes the types of inputs supported by this method.

Type	Upload method
Strings	Strings are uploaded directly.
Bytes	Bytes are uploaded directly.
Unicode	Unicode characters are converted to bytes encoded in UTF-8 for upload.
Local files	Local files are uploaded as file objects. File objects must be opened in binary mode such as the <code>rb</code> mode.
Network streams	Network streams are uploaded as iterable objects by using chunked encoding.

Note If you upload an object with the same name as that of an accessible existing object, the existing object is overwritten by the uploaded object and 200 OK is returned.

Upload strings

The following code provides an example on how to upload a string to an OSS object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Obtain the return result.
result = bucket.put_object('<yourObjectName>', 'content of object')
# Display the returned HTTP status code.
print('http status: {}'.format(result.status))
# Display the request ID. A request ID uniquely identifies a request. We recommend that you add this parameter in the logs.
print('request_id: {}'.format(result.request_id))
# Obtain the ETag value returned by the put_object method.
print('ETag: {}'.format(result.etag))
# Display the HTTP response headers.
print('date: {}'.format(result.headers['date']))
```

Upload bytes

The following code provides an example on how to upload bytes:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.put_object('<yourObjectName>', b'content of object')
```

Upload Unicode characters

The following code provides an example on how to upload Unicode:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.put_object('<yourObjectName>', u'content of object')
```

Upload a local file

The following code provides an example on how to upload a local file to OSS:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# The object must be opened in binary mode because the number of bytes in the object must be available before the upload.
with open('<yourLocalFile>', 'rb') as fileobj:
    # Use the seek method to read data from byte 1,000 of the file. The data is uploaded from byte 1,000 to the last byte of the local file.
    fileobj.seek(1000, os.SEEK_SET)
    # Use the tell method to obtain the current position.
    current = fileobj.tell()
    bucket.put_object('<yourObjectName>', fileobj)
```

OSS SDK for Python also provides a more convenient method to upload local files:

```
bucket.put_object_from_file('<yourObjectName>', '<yourLocalFile>')
```

Upload a network stream

The following code provides an example on how to upload a network stream:

```
# -*- coding: utf-8 -*-
import oss2
import requests
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# The requests.get method returns an iterable object. OSS SDK for Python uploads the object by means of chunked encoding.
input = requests.get('http://www.aliyun.com')
bucket.put_object('<yourObjectName>', input)
```

1.4.2.6.3. Append upload

Object Storage Service (OSS) allows you to use append upload to append content directly to the end of an object. You can apply this method only to appendable objects.

The following code provides an example on how to upload local files to OSS by using append upload:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set the position from which the first append operation starts to 0.
result = bucket.append_object('<yourObjectName>', 0, 'content of first append')
# If you have appended content to the object, you can obtain the position from which the operation starts this time from the next_position field in the response returned by the last operation or by using the bucket.head_object method.
bucket.append_object('<yourObjectName>', result.next_position, 'content of second append')
```

If the object already exists, exceptions occur in the following cases:

- If the object is not appendable, `ObjectNotAppendable` is returned.
- If the object is appendable but the append position does not match the current object size, `PositionNotEqualToLength` is returned.

1.4.2.6.4. Resumable upload

If the system stops responding during multipart upload, you can resume the upload by using the `ListMultipartUploads` and `ListParts` operations to retrieve all multipart upload tasks on an object and list the uploaded parts in each task. This method allows an upload task to be resumed from the last uploaded part.

You can use the `oss2.resumable_upload` method to perform resumable upload for a specified object. The following table describes the parameters you can configure for the `oss2.resumable_upload` method.

Parameter	Description	Required	Default value
bucket	The name of the bucket.	Yes	None
key	The name of the object.	Yes	None
filename	The local file path from which to upload the object.	Yes	None
store	The directory where the checkpoint information is stored.	No	The <code>.py-oss-upload</code> directory created in the HOME directory.
headers	The HTTP headers.	No	None
multipart_threshold	The object length threshold for multipart upload. If the object length exceeds the threshold, multipart upload is used.	No	10 MB

Parameter	Description	Required	Default value
part_size	The size of the part.	No	Automatically calculated
progress_callback	The callback function to return upload progress.	No	None
num_threads	The number of concurrent upload threads.	No	1

The following code provides an example on how to perform resumable upload:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Use multipart upload when the object length is greater than or equal to the multipart_threshold value. The multipart_threshold parameter is optional, and the default value of multipart_threshold is 10 MB. If you do not specify a directory by using the store parameter, the .py-oss-upload directory is created in the HOME directory to store the checkpoint information.
oss2.resumable_upload(bucket, '<yourObjectName>', '<yourLocalFile>')
```

OSS SDK for Python version 2.1.0 and later support resumable upload by using optional parameters. The following code provides an example on how to use optional parameter in resumable upload:

```
# If you specify a directory by using the store parameter, the checkpoint information is stored in the specified directory. If you use num_threads to specify the number of concurrent upload threads, ensure that oss2.defaults.connection_pool_size is set to a value greater than the number of concurrent upload threads. The default number of concurrent upload threads is 1.
oss2.resumable_upload(bucket, '<yourObjectName>', '<yourLocalFile>',
    store=oss2.ResumableStore(root='/tmp'),
    multipart_threshold=100*1024,
    part_size=100*1024,
    num_threads=4)
```

Note

- Multiple threads are used in resumable upload. Therefore, you do not need to encapsulate multiple upload threads when you call the resumable upload method. Otherwise, data may be transferred repeatedly.
- We recommend that you increase the part size when network conditions are good, and decrease the part size when network conditions are poor.

1.4.2.6.5. Multipart upload

OSS allows you to use multipart upload to split an object into several parts and upload them separately. After all parts are uploaded, you can combine the uploaded parts into a complete object to complete the upload.

To implement multipart upload, perform the following operations:

The following code provides an example on how to perform multipart upload:

```
# -*- coding: utf-8 -*-
import os
from oss2 import SizedFileAdapter, determine_part_size
from oss2.models import PartInfo
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
key = '<yourObjectName>'
filename = '<yourLocalFile>'
total_size = os.path.getsize(filename)
# Use the determine_part_size method to determine the part size.
part_size = determine_part_size(total_size, preferred_size=100 * 1024)
# Initiate a multipart upload task.
# To set the storage class for the object when you initiate the multipart upload task, set the headers parameter in the init_multipart_upload function.
# headers = dict()
# headers["x-oss-storage-class"] = "Standard"
# upload_id = bucket.init_multipart_upload(key, headers=headers).upload_id
upload_id = bucket.init_multipart_upload(key).upload_id
parts = []
# Upload the parts one by one.
with open(filename, 'rb') as fileobj:
    part_number = 1
    offset = 0
    while offset < total_size:
        num_to_upload = min(part_size, total_size - offset)
        # The SizedFileAdapter(fileobj, size) method generates a new object, and resets the append position.
        result = bucket.upload_part(key, upload_id, part_number,
                                    SizedFileAdapter(fileobj, num_to_upload))
        parts.append(PartInfo(part_number, result.etag))
        offset += num_to_upload
        part_number += 1
# Complete the multipart upload task.
# To configure ACL for the object when you complete the multipart upload task, set the headers parameter in the complete_multipart_upload function.
# headers = dict()
# headers["x-oss-object-acl"] = oss2.OBJECT_ACL_PRIVATE
# bucket.complete_multipart_upload(key, upload_id, parts, headers=headers)
bucket.complete_multipart_upload(key, upload_id, parts)
# Verify multipart upload.
with open(filename, 'rb') as fileobj:
    assert bucket.get_object(key).read() == fileobj.read()
```

Cancel a multipart upload task

You can call `bucket.abort_multipart_upload` to cancel a multipart upload task. If you cancel a multipart upload task, you cannot use the upload ID to perform any operations on the parts. The uploaded parts are deleted.

The following code provides an example on how to cancel a multipart upload task:

```
# -*- coding: utf-8 -*-
import os
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
key = '<yourObjectName>'
# Obtain the upload ID returned by the init_multipart_upload operation.
upload_id = '<yourUploadId>'
# When you cancel the multipart upload task by using the specified upload ID, the uploaded parts are deleted.
bucket.abort_multipart_upload(key, upload_id)
```

List uploaded parts

The following code provides an example on how to list uploaded parts:

```
# -*- coding: utf-8 -*-
import os
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
key = '<yourObjectName>'
# Obtain the upload ID returned by the init_multipart_upload operation.
upload_id = '<yourUploadId>'
# List uploaded parts by using the specified upload ID.
for part_info in oss2.PartIterator(bucket, key, upload_id):
    print('part_number:', part_info.part_number)
    print('etag:', part_info.etag)
    print('size:', part_info.size)
```

List multipart upload tasks

- List multipart upload tasks of a specified object

The following code provides an example on how to list multipart upload tasks of a specified object:

```
# -*- coding: utf-8 -*-
import os
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
key = '<yourObjectName>'
# List all multipart upload tasks of the object. Each time init_multipart_upload is called for the same object, different upload_id values are returned.
# One upload ID corresponds to one multipart upload task.
for upload_info in oss2.ObjectUploadIterator(bucket, key):
    print('key:', upload_info.key)
    print('upload_id:', upload_info.upload_id)
```

- List all multipart upload tasks in a bucket

The following code provides an example on how to list all multipart upload tasks in a bucket:

```
# -*- coding: utf-8 -*-
import os
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# List all multipart upload tasks in the bucket.
for upload_info in oss2.MultipartUploadIterator(bucket):
    print('key:', upload_info.key)
    print('upload_id:', upload_info.upload_id)
```

- List all multipart upload tasks for objects whose names contain a specified prefix in a bucket

The following code provides an example on how to list multipart upload tasks for objects whose names contain a specified prefix in a bucket:

```
# -*- coding: utf-8 -*-
import os
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# List multipart upload tasks for objects whose names contain the prefix test in the bucket.
for upload_info in oss2.MultipartUploadIterator(bucket, prefix='test'):
    print('key:', upload_info.key)
    print('upload_id:', upload_info.upload_id)
```

1.4.2.6.6. Use progress bars

Progress bars are used to indicate the progress of an upload or download.

For the complete code of progress bar, visit [GitHub](#).

The following code uses `bucket.put_object` as an example to describe how to use a progress bars:

```
# -*- coding: utf-8 -*-
from __future__ import print_function
import os, sys
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# If the length of the data to upload cannot be determined, the value of total_bytes is None.
def percentage(consumed_bytes, total_bytes):
    if total_bytes:
        rate = int(100 * (float(consumed_bytes) / float(total_bytes)))
        print('\r{0}% '.format(rate), end='')
        sys.stdout.flush()
# progress_callback is an optional parameter used to return progress information.
bucket.put_object('<yourObjectName>', 'a'*1024*1024, progress_callback=percentage)
```

1.4.2.7. Download objects

1.4.2.7.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for Python provides the following object download methods:

You can use one of the following methods to download an object:

- [Streaming download](#)
- [Download to local files](#)
- [Range download](#)
- [Resumable download](#)

You can view the download progress by using a progress bar. For more information, see [Use progress bars](#).

1.4.2.7.2. Streaming download

If you must download a large object or if the download requires a long period of time to complete, you can perform streaming download to download the object in increments.

The following code provides an example on how to download an object by using streaming download:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# The value returned by bucket.get_object is a file-like object that can be iterated.
object_stream = bucket.get_object('<yourObjectName>')
print(object_stream.read())
# You must call read() to read the object from the stream that is returned by get_object before you can calculate the CRC-64 checksum of the object.
if object_stream.client_crc != object_stream.server_crc:
    print "The CRC checksum between client and server is inconsistent!"
```

The following code provides an example on how to download data to a local file in streaming mode:

```
import shutil
# object_stream is a file-like object. You can use the shutil.copyfileobj method to download the data to your local file.
object_stream = bucket.get_object('<yourObjectName>')
with open('<yourLocalFile>', 'wb') as local_fileobj:
    shutil.copyfileobj(object_stream, local_fileobj)
```

The following code provides an example on how to copy data to another object in streaming mode:

```
# object_stream is an object that can be iterated. You can copy object_stream to another object in streaming mode.
object_stream = bucket.get_object('<yourObjectName>')
bucket.put_object('<yourBackupObjectName>', object_stream)
```

1.4.2.7.3. Download to local files

This topic describes how to download an object to a local file.

The following code provides an example on how to download a specified object to a local file:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Download the object to a local file. If the specified local file already exists, the downloaded object replaces the local file. If the specified local file does not exist, it is created.
# <yourLocalFile> consists of a local file path and an object name with an extension. Example: /users/local/myfile.txt.
bucket.get_object_to_file('<yourObjectName>', '<yourLocalFile>')
```

1.4.2.7.4. Range download

You can use range download to download a specified range of data of an object.

Specify a valid range to download data

The following code provides an example on how to specify a valid range to download data:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# If an object is 1,000 bytes in size, the valid range is from byte 0 to byte 999.
# Query data that is within the range from byte 0 to byte 999, which includes a total of 1,000 bytes. If the specified range is invalid, the entire object is downloaded. For example, if the specified range includes a negative number, or if the specified value is larger than the object size, all data of the object is downloaded.
object_stream = bucket.get_object('<yourObjectName>', byte_range=(0, 999))
```

Specify an invalid range

The valid range for an object whose size is 1,000 bytes is from byte 0 to byte 999. If the specified range is not within the range, the range does not take effect. Then, OSS returns the 200 HTTP status code and the data of the entire object. The following list provides examples of invalid requests and the returned results:

- If you set Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns the 200 HTTP status code and the data of the entire object.
- If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns the 200 HTTP status code and the data of the entire object.

Specify the compatible behavior to download data by range

If you add x-oss-range-behavior:standard to the request header, the download behavior is modified when the specified range is not within the valid range. Example: Use range download to download an object whose size is 1,000 bytes.

- If you set Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns HTTP status code 206 and the data that is within the range from

byte 500 to byte 999.

- If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns HTTP status code 416 and error code InvalidRange.

The following code provides an example on how to specify the compatible behavior to download data by range:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Create an object whose size is 1,000 bytes.
object_name = 'rangeTest.txt'
content = 'a' * 1000
bucket.put_object(object_name, content)
headers = {'x-oss-range-behavior': 'standard'}
# Specify the compatible behavior.
# When the end value of the range is invalid, OSS returns the data that is within the range from byte 500 to byte 999 and HTTP status code 206.
object_stream = bucket.get_object(object_name, byte_range=(500, 2000), headers=headers)
print('standard get 500-2000 http status code:', object_stream.status)
print('standard get 500-2000 content length:', object_stream.content_length)
try:
    # When the start value of the range is invalid, OSS returns HTTP status code 416 and the InvalidRange error code.
    object_stream = bucket.get_object(object_name, byte_range=(1000, 2000), headers=headers)
except oss2.exceptions.ServerError as e:
    print('standard get 1000-2000 http status code:', e.status)
    print('standard get 1000-2000 error code:', e.code)
```

1.4.2.7.5. Resumable download

You may fail to download a large object if the network is unstable or if the program stops responding. In some cases, you may still fail to download the object even after multiple attempts. To solve this problem, OSS SDK for Java provides the resumable download feature.

You can perform the following steps to perform resumable download:

1. Create a temporary local file with a name that consists of the original object name and a random suffix.
2. Read the object based on the range specified in the Range header in the HTTP request and write the read content to the corresponding position of the temporary local file.
3. After the download is complete, rename the temporary file as the destination file. If the destination file already exists, the downloaded data overwrites the data in the existing file. Otherwise, a new file is created.

You can use the `oss2.resumable_download` method to perform resumable download. The following table describes the parameters used in this method.

Parameter	Description	Required	Default value
bucket	The name of the bucket.	Yes	None
key	The name of the object.	Yes	None
filename	The path to the local file. The local file to which the object is downloaded.	Yes	None
store	The file that records the results of multipart download. This file stores information about download progress. If you fail to download a part, the download continues based on the recorded progress.	No	The <code>.py-oss-upload</code> directory created in the HOME directory.
multipart_threshold	The threshold for the object size. If the object size exceeds the threshold, resumable download is enabled.	No	10 MB
part_size	The size of the part.	No	Automatically calculated.
progress_callback	The callback function used to return download progress information.	No	None
num_threads	The number of threads for simultaneous downloads.	No	1

Run the following code for resumable download:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
oss2.resumable_download(bucket, '<yourObjectName>', '<yourLocalFile>')
```

You can configure optional parameters for resumable upload when you use OSS SDK for Python V2.1.0 or later version. The following code provides you an example:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set oss2.defaults.connection_pool_size to a value greater than or equal to the number of threads. Then set part_size to a value greater than or equal to the oss2.defaults.multiget_part_size value.
oss2.resumable_download(bucket, '<yourObjectName>', '<yourLocalFile>',
    store=oss2.ResumableDownloadStore(root='/tmp'),
    multiget_threshold=20*1024*1024,
    part_size=10*1024*1024,
    num_threads=3)
```

Note Do not use multiple programs or threads to call the method simultaneously to download the same object to a same destination file. This is because one piece of checkpoint information overwrites another on the local disk, or one temporary file name conflicts with another.

1.4.2.7.6. Use progress bars

Progress bars are used to indicate the progress of an upload or download.

For the complete code of progress bar, visit [GitHub](#).

The following code provides an example on how to use a progress bar in object download performed by using the bucket.get_object_to_file method:

```
# -*- coding: utf-8 -*-
from __future__ import print_function
import os, sys
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# If Content-Length is not configured in the HTTP response header, the value of total_bytes is None.
def percentage(consumed_bytes, total_bytes):
    if total_bytes:
        rate = int(100 * (float(consumed_bytes) / float(total_bytes)))
        print('\r(0)% ' .format(rate), end='')
        sys.stdout.flush()
# progress_callback is an optional parameter used to implement a progress bar.
bucket.get_object_to_file('<yourObjectName>', '<yourLocalFile>', progress_callback=percentage)
```

1.4.2.8. Manage objects

1.4.2.8.1. Overview

You can manage objects in a bucket by using a variety of operations such as SetObjectAcl, GetObjectAcl, ListObjects, DeleteObject, and CopyObject.

By using the preceding operations, you can perform the following operations:

- [Determine whether an object exists](#)
- [Manage object ACLs](#)
- [Manage object metadata](#)
- [List objects](#)
- [Query objects](#)
- [Delete objects](#)
- [Copy an object](#)

1.4.2.8.2. Determine whether an object exists

This topic describes how to determine whether an object exists.

The following code provides an example on how to determine whether an object exists:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
exist = bucket.object_exists('<yourObjectName>')
# If the returned value is true, the specified object exists. If the returned value is false, the specified object does not exist.
if exist:
    print('object exist')
else:
    print('object not exist')
```

1.4.2.8.3. Manage object ACLs

This topic describes how to manage the access control list (ACL) of an object.

The following table describes the ACLs that you can configure for an object.

ACL	Description	Value
Inherited from the bucket	The ACL of the object is the same as the ACL of the bucket in which the object is stored.	oss2.OBJECT_ACL_DEFAULT
Private	Only the object owner and authorized users are granted read and write permissions on the object.	oss2.OBJECT_ACL_PRIVATE
Public read	Only the object owner and authorized users are granted read and write permissions on the object. Other users are granted only read permissions on the object. Exercise caution when you set the ACL of the object to this value.	oss2.OBJECT_ACL_PUBLIC_READ
Public read/write	All users are granted read and write permissions on the object. Exercise caution when you set the ACL of the object to this value.	oss2.OBJECT_ACL_PUBLIC_READ_WRITE

The ACL of the object takes precedence over the ACL of the bucket. For example, if the ACL of an object in a private bucket is public read/write, all users have read and write permissions on the object. If the ACL of an object is not configured, the ACL of the object is the same as the ACL of the bucket in which the object is stored.

Configure the ACL of an object

The following code provides an example on how to configure the ACL of an object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Configure the object ACL.
bucket.put_object_acl('<yourObjectName>', oss2.OBJECT_ACL_PUBLIC_READ)
```

Query the ACL of an object

The following code provides an example on how to query the ACL of a specified object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
print(bucket.get_object_acl('<yourObjectName>').acl)
```

1.4.2.8.4. Manage object metadata

Object metadata includes HTTP headers and user metadata.

Configure HTTP headers

The following code provides an example on how to configure the HTTP headers of an object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.put_object('<yourObjectName>', '{"age": 1}', headers={'Content-Type': 'application/json; charset=utf-8'})
```

For more information about HTTP headers, see [RFC 2616](#).

Configure user metadata

You can configure the user metadata of an object to describe the object.

The following code provides an example on how to configure the user metadata of an object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.put_object('<yourObjectName>', 'a novel', headers={'x-oss-meta-author': 'O. Henry', 'Content-Type': 'application/json; charset=utf-8'})
```

Modify object metadata

The following code provides an example on how to modify the metadata of an object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
bucket.update_object_meta('<yourObjectName>', {'x-oss-meta-author': 'O. Henry'})
# Each time you call bucket.update_object_meta, the existing user metadata is updated.
bucket.update_object_meta('<yourObjectName>', {'x-oss-meta-price': '100 dollar'})
```

The following code provides an example on how to modify object metadata such as Content-Type:

```
bucket.update_object_meta('<yourObjectName>', {'x-oss-meta-author': 'O. Henry'})
# Each time you call bucket.update_object_meta, the existing user metadata is updated.
bucket.update_object_meta('<yourObjectName>', {'Content-Type': 'text/plain'})
```

Query object metadata

You can use the API operations provided by OSS SDK for Python to query object metadata.

Method	Description	Advantage
get_object_meta	Obtains part of object metadata such as the ETag, size, and last modified time of the object.	More lightweight and faster
head_object	Obtains all object metadata.	None

The following code provides an example on how to query object metadata:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
// Query part of the object metadata.
simplifiedmeta = bucket.get_object_meta("<yourObjectName>")
print(simplifiedmeta.headers['Last-Modified'])
print(simplifiedmeta.headers['Content-Length'])
print(simplifiedmeta.headers['ETag'])
// Query all object metadata.
objectmeta = bucket.head_object("<yourObjectName>")
print(objectmeta.headers['Content-Type'])
print(objectmeta.headers['Last-Modified'])
print(objectmeta.headers['x-oss-object-type'])
```

1.4.2.8.5. List objects

Objects in a bucket are sorted alphabetically. OSS SDK for Python provides a series of object listing methods.

Simple list

The following code provides an example on how to list 10 objects in a specified bucket:

```
# -*- coding: utf-8 -*-
import oss2
from itertools import islice
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
for b in islice(oss2.ObjectIterator(bucket), 10):
    print(b.key)
```

List all objects in a bucket

The following code provides an example on how to list all objects in a bucket:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# List all objects in the specified bucket.
for obj in oss2.ObjectIterator(bucket):
    print(obj.key)
```

List objects whose names contain a specified prefix

Example: A bucket contains the following objects: oss.jpg, fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. The forward slash (/) is used as the folder delimiter.

The following code provides an example on how to list all objects whose names contain a specified prefix:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# List all objects in the fun directory and its subdirectories.
for obj in oss2.ObjectIterator(bucket, prefix='fun/'):
    print(obj.key)
```

The following figure shows the result.

```
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
```

List objects and subdirectories in a specified directory

OSS uses a flat structure for objects instead of a hierarchical structure to store objects. A directory is an object whose size is 0 and whose name ends with a forward slash (/). You can upload and download this object. By default, objects whose names end with a forward slash (/) are displayed as directories in the OSS console. For the complete sample code that is used to create directories, visit [GitHub](#).

You can specify the delimiter and prefix parameters to list objects by directory.

- If you set prefix to the name of a directory, objects and subdirectories whose names contain the prefix are listed.
- If you set delimiter to a forward slash (/), only the objects and subdirectories in the directory are listed. The objects and directories in subdirectories are not listed.

The following code provides an example on how to list objects and subdirectories in a specified directory:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# List objects and subdirectories in the fun directory. Objects in the subdirectories are not listed.
for obj in oss2.ObjectIterator(bucket, prefix = 'fun/', delimiter = '/'):
    # Use is_prefix to determine whether obj is a directory.
    if obj.is_prefix(): # Determine that obj is a directory.
        print('directory: ' + obj.key)
    else: # Determine that obj is an object.
        print('file: ' + obj.key)
```

The following figure shows the result.

```
directory: fun/movie/
file: fun/test.jpg
```

List the sizes of objects in a specified directory

The following code provides an example on how to query the sizes of objects in a specified directory:

```
# -*- coding: utf-8 -*-
import oss2
def CalculateFolderLength(bucket, folder):
    length = 0
    for obj in oss2.ObjectIterator(bucket, prefix=folder):
        length += obj.size
    return length
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
for obj in oss2.ObjectIterator(bucket, delimiter='/'):
    if obj.is_prefix(): # Determine that obj is a directory.
        length = CalculateFolderLength(bucket, obj.key)
        print('directory: ' + obj.key + ' length: ' + str(length) + "Byte.")
    else: # Determine that obj is an object.
        print('file: ' + obj.key + ' length: ' + str(obj.size) + "Byte.")
```

1.4.2.8.6. Query objects

This topic describes how to use SelectObject in OSS SDK for Python to query CSV and JSON objects.

OSS SDK for Python examples

The following code provides an example on how to query CSV and JSON objects:

```

import oss2
def select_call_back(consumed_bytes, total_bytes = None):
    print('Consumed Bytes:' + str(consumed_bytes) + '\n')
access_key_id = '<yourAccessKeyId>'
access_key_secret = '<yourAccessKeySecret>'
bucket_name = '<yourBucketName>'
endpoint = '<yourEndpoint>'
# Create an OSS bucket. All object-related methods must be called based on the bucket.
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret), endpoint, bucket_name)
key = 'python_select.csv'
content = 'Tom Hanks,USA,45\r\n'*1024
filename = 'python_select.csv'
# Upload a CSV object.
bucket.put_object(key, content)
# Set the parameters for the SelectObject API operation.
csv_meta_params = {'RecordDelimiter': '\r\n'}
select_csv_params = {'CsvHeaderInfo': 'None',
                    'RecordDelimiter': '\r\n',
                    'LineRange': (500, 1000)}
csv_header = bucket.create_select_object_meta(key, csv_meta_params)
print(csv_header.rows)
print(csv_header.splits)
result = bucket.select_object(key, "select * from ossobject where _3 > 44", select_call_back, select_csv_params)
select_content = result.read()
print(select_content)
result = bucket.select_object_to_file(key, filename,
                                     "select * from ossobject where _3 > 44", select_call_back, select_csv_params)
bucket.delete_object(key)
###JSON DOCUMENT
key = 'python_select.json'
content = "{\"contacts\": [{\"key1\":1,\"key2\":\"hello world1\"},{\"key1\":2,\"key2\":\"hello world2\"}]}"
filename = 'python_select.json'
# Upload a JSON Document object.
bucket.put_object(key, content)
select_json_params = {'Json_Type': 'DOCUMENT'}
result = bucket.select_object(key, "select s.key2 from ossobject.contacts[*] s where s.key1 = 1", None, select_json_params)
select_content = result.read()
print(select_content)
result = bucket.select_object_to_file(key, filename,
                                     "select s.key2 from ossobject.contacts[*] s where s.key1 = 1", None, select_json_params)
bucket.delete_object(key)
###JSON LINES
key = 'python_select_lines.json'
content = "{\"key1\":1,\"key2\":\"hello world1\"}\n{\"key1\":2,\"key2\":\"hello world2\"}"
filename = 'python_select.json'
# Upload a JSON Lines object.
bucket.put_object(key, content)
select_json_params = {'Json_Type': 'LINES'}
json_header = bucket.create_select_object_meta(key, select_json_params)
print(json_header.rows)
print(json_header.splits)
result = bucket.select_object(key, "select s.key2 from ossobject s where s.key1 = 1", None, select_json_params)
select_content = result.read()
print(select_content)
result = bucket.select_object_to_file(key, filename,
                                     "select s.key2 from ossobject s where s.key1 = 1", None, select_json_params)
bucket.delete_object(key)

```

Python Select API

The following section describes the elements used for the SelectObject API operation in Python in detail. These elements include `select_object`, `select_object_to_file`, and `create_select_object_meta`.

- `select_object`
 - The following code provides an example on how to set parameters for `select_object`:

```

def select_object(self, key, sql,
                 progress_callback=None,
                 select_params=None,
                 byte_range=None,
                 headers=None
                 ):

```

The preceding code example is used to execute an SQL statement on the object that has the specified key and return the query results.

- `sql`: the SQL statement that can be directly used as the value of the `sql` parameter and does not need to be Base64-encoded.
- `Progress_callback`: optional. This parameter specifies a callback function used to report the query progress.
- `select_params`: the parameters and actions of the SelectObject operation.
- `headers`: the header information included in the request. The header information functions the same as that for the GetObject operation. For example, you can configure the `bytes` field in the request header to specify the range of a CSV object to query.

- The following table describes the parameters supported by the `select_params` parameter.

Parameter	Description
Json_Type	- By default, the object is a CSV object if this parameter is not specified. - The object is a JSON Document object if this parameter is set to DOCUMENT. - The object is a JSON Lines object if this parameter is set to LINES.
CsvHeaderInfo	The header information of the CSV object. Valid values: <i>None</i>, <i>Ignore</i>, and <i>Use</i>. <i>None</i>: This object has no header information configured. <i>Ignore</i>: This object has header information configured, which is not used when the SQL statement is executed. <i>Use</i>: This object has header information configured. The column names in the header information are used when the SQL statement is executed.
CommentCharacter	The comment character in the CSV object. This parameter value can be only one character in length. The default value is None, which indicates that no comment characters are allowed.
RecordDelimiter	The row delimiter in the CSV object. This parameter value can be only one or two characters in length. Default value: \n.
OutputRecordDelimiter	The row delimiter in the output result of the SELECT statement. Default value: \n.
FieldDelimiter	The column delimiter in the CSV object. This parameter value can be only one character in length. Default value: comma (,).
OutputFieldDelimiter	The column delimiter in the output result of the SELECT statement. Default value: comma (,).
QuoteCharacter	The quote character for the columns in the CSV object. This parameter value can be only one character in length. Default value: double quotation marks ("). Row and column delimiters enclosed in quote characters are processed as normal characters.
SplitRange	The split range in multipart query. The parameter value is a closed interval in the (start, end) format, which indicates the range of splits to query.
LineRange	The row range in multipart query. The parameter value is a closed interval in the (start, end) format, which indicates the range of rows to query.
CompressionType	The format in which the object is compressed. Valid value: GZIP. Default value: None.
KeepAllColumns	If this parameter is set to true, columns that are excluded by the SELECT statement in the CSV object are left empty in the output result. However, the column positions are kept. Default value: false. Example: The columns in the CSV object are <code>firstname</code>, <code>lastname</code>, and <code>age</code>. The SQL statement is <code>select firstn ame, age from ossobject</code>. - If <code>KeepAllColumns</code> is set to <i>true</i>, the output result is <code>firstname,,age</code>, in which a comma (,) is added to indicate the position of the excluded <code>lastname</code> column. - If <code>KeepAllColumns</code> is set to <i>false</i>, the output result is <code>firstname,age</code>. Note By using this parameter, you can directly use the code that is used to process <code>GetObject</code> to process <code>SelectObject</code> without modifications.
OutputRawData	- If this parameter is set to <i>True</i>, <code>SelectObject</code> directly returns the original data. If data is not returned for a long time, a timeout error may occur. - If this parameter is set to <i>False</i>, the output data is encapsulated in frames. Default value: <i>False</i>.
EnablePayloadCrc	Specifies whether a cyclic redundancy check (CRC) value is calculated for each frame. Default value: <i>False</i> .
OutputHeader	The header information in the first line of the output result. This parameter applies only to CSV objects.
SkipPartialDataRecord	If this parameter is set to <i>True</i> for a CSV object, the current record is skipped when a column of this record has no data. If this parameter is set to <i>True</i> for a JSON object, the current record is skipped when a key of this record does not exist. If this parameter is set to <i>False</i>, a column without data is left empty in the output result. Example: A row includes the columns <code>firstname</code>, <code>lastname</code>, and <code>age</code>. The SQL statement is <code>select _1, _4 from ossobject</code>. - If this parameter is set to <i>True</i>, this row is skipped. - If this parameter is set to <i>False</i>, <code>firstname,\n</code> is returned.
MaxSkippedRecordsAllowed	The maximum number of skipped rows. The default value is 0, which indicates that an error is returned if a row is skipped.
ParseJsonNumberAsString	- If this parameter is set to <i>True</i>, all numbers in the JSON object are parsed as strings. - If this parameter is set to <i>False</i>, all numbers in the JSON object are parsed as integers or floating-point numbers. Default value: <i>False</i>. High-accuracy floating-point numbers in a JSON object suffer loss of accuracy when they are parsed as floating-point numbers. To keep the accuracy, you can set this parameter to <i>True</i> and use the <code>CAST</code> function to convert the parsed data to the decimal type.

- Returned results of `select_object`: A `SelectObjectResult` object is returned. You can use the `read()` function or the `__iter__` method to obtain all results.

Note If you call the `read()` function to read all results when there are multiple results to return, excess memory resources are used and you have to wait for a long time. We recommend that you use the `__iter__` method (foreach chunk in result) to obtain results and process each chunk in the results. The `__iter__` method reduces memory usage and enables the client to process each chunk request processed by the OSS server in a timely manner. This way, the client does not need to wait until all results are returned.

- `select_object_to_file`

```
def select_object_to_file(self, key, filename, sql,
                        progress_callback=None,
                        select_params=None,
                        headers=None
                        ):
    pass
```

The preceding code example is used to execute an SQL statement on the object that has the specified key and write the query results to another specified object.

- `create_select_object_meta`

- Syntax of `create_select_object_meta`:

```
def create_select_object_meta(self, key, select_meta_params=None, header=None):
    pass
```

The preceding code example is used to create Select Meta for or obtain Select Meta from the object that has the specified key. Select Meta includes the total number of rows, total number of columns (for CSV objects), and total number of splits in an object.

If Select Meta has been created for an object, this function does not re-create Select Meta unless the value of the `OverwriteIfExists` parameter is set to true.

To create Select Meta for an object, the object must be completely scanned.

- The following table describes the parameters supported by the `select_meta_params` parameter.

Parameter	Description
<code>Json_Type</code>	- By default, the object is a CSV object if this parameter is not specified. - If this parameter is specified, the parameter value must be <code>LINES</code>, which indicates that the object is a JSON Lines object. Note This operation does not apply to JSON Document objects.
<code>RecordDelimiter</code>	The row delimiter in the CSV object.
<code>FieldDelimiter</code>	The column delimiter in the CSV object.
<code>QuoteCharacter</code>	The quote character in the CSV object. Row and column delimiters enclosed in quote characters are processed as normal characters.
<code>CompressionType</code>	The format in which the object is compressed. No compression formats are supported. The default value is <code>None</code> .
<code>OverwriteIfExists</code>	Specifies whether the created Select Meta overwrites the original Select Meta. You do not need to set this parameter in most cases.

- Returned results of `create_select_object_meta`: A `GetSelectObjectMetaResult` object is returned, which includes the rows and splits attributes. If the object is a CSV object, the `select_resp` object in the result includes the columns attribute, which indicates the number of columns in the CSV object.

1.4.2.8.7. Delete objects

This topic describes how to delete objects.

Warning Deleted objects cannot be recovered. Exercise caution when you delete objects.

Delete a single object

The following code provides an example on how to delete a single object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Delete the object. <yourObjectName> indicates the full path of the object that you want to delete from OSS. The path must include the extension of
the object name. Example: abc/efg/123.jpg.
# To delete a directory, set <yourObjectName> to the name of the directory you want to delete. If the specified directory is not empty, you must delete
all objects in the directory before you can delete the directory.
bucket.delete_object('<yourObjectName>')
```

Delete multiple objects

The following code provides an example on how to delete multiple objects at a time:

```
# Delete three objects at a time. You can delete up to 1,000 objects at a time.
result = bucket.batch_delete_objects(['<yourObjectName-a>', '<yourObjectName-b>', '<yourObjectName-c>'])
# Display the names of the deleted objects.
print('\n'.join(result.deleted_keys))
```

Delete objects whose names contain a specified prefix

The following code provides an example on how to delete objects whose names contain a specified prefix:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
prefix = "<yourKeyPrefix>"
# Delete the objects whose names contain the specified prefix.
for obj in oss2.ObjectIterator(bucket, prefix=prefix):
    bucket.delete_object(obj.key)
```

1.4.2.8.8. Copy objects

You can copy an object from a source bucket to a destination bucket within the same region.

When you copy a object, take note of the following items:

- You must have read permissions on the source object and write permissions on the destination bucket.
- The source bucket and the destination bucket must be in the same region.

Simple copy

You can use simple copy for objects up to 1 GB in size. The following code provides an example on how to perform simple copy:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourDestinationBucketName>')
bucket.copy_object('<yourSourceBucketName>', '<yourSourceObjectName>', '<yourDestinationObjectName>')
```

Copy a large object

To copy an object whose size is larger than 1 GB, you must split the object into parts and sequentially copy the parts by using UploadPartCopy. To implement multipart copy, perform the following steps:

1. Use `bucket.init_multipart_upload` to initiate a multipart copy task.
2. Use `bucket.upload_part_copy` to copy the parts. Each part must be larger than 100 KB in size, except the last part.
3. Use `bucket.complete_multipart_upload` to complete the multipart copy task.

The following code provides an example on how to perform multipart copy:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import PartInfo
from oss2 import determine_part_size
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
src_object = '<yourSourceObjectName>'
dst_object = '<yourDestinationObjectName>'
total_size = bucket.head_object(src_object).content_length
part_size = determine_part_size(total_size, preferred_size=100 * 1024)
# Initiate a multipart upload task.
upload_id = bucket.init_multipart_upload(dst_object).upload_id
parts = []
# Copy each part sequentially.
part_number = 1
offset = 0
while offset < total_size:
    num_to_upload = min(part_size, total_size - offset)
    byte_range = (offset, offset + num_to_upload - 1)
    result = bucket.upload_part_copy(bucket.bucket_name, src_object, byte_range, dst_object, upload_id, part_number)
    parts.append(PartInfo(part_number, result.etag))
    offset += num_to_upload
    part_number += 1
# Complete the multipart copy task.
bucket.complete_multipart_upload(dst_object, upload_id, parts)
```

1.4.2.9. Versioning

1.4.2.9.1. Manage versioning

The versioning state of a bucket applies to all objects in the bucket. If you enable versioning for a bucket, you can recover any previous version of an object in the bucket when you accidentally overwrite or delete the object.

A bucket can be in one of the following versioning states: unversioned (default), versioning-enabled, or versioning-suspended.

 **Notice** If versioning is enabled for a bucket, the versioning state of the bucket cannot be set back to unversioned. However, you can set the versioning state of the bucket to suspended.

Configure the versioning state of a bucket

The following code provides an example on how to set the versioning state of a bucket to enabled or suspended:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketVersioningConfig
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in Object Storage Service (OSS) is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Initialize versioning configurations for the bucket.
config = BucketVersioningConfig()
# Set the versioning state to enabled or suspended.
config.status = oss2.BUCKET_VERSIONING_ENABLE
# Configure the versioning state of the bucket.
result = bucket.put_bucket_versioning(config)
# Display the HTTP status code.
print('http response code:', result.status)
```

Query the versioning state of a bucket

The following code provides an example on how to query the versioning state of a bucket:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console .
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Query the versioning state of the bucket.
versioning_info = bucket.get_bucket_versioning()
# Display the versioning state of the bucket. If versioning has been enabled, Enabled or Suspended is returned. If versioning has not been enabled, None is returned.
print('bucket versioning status:', versioning_info.status)
```

1.4.2.9.2. Upload objects

This topic describes how to upload objects to a versioned bucket by using simple upload, append upload, and multipart upload.

Simple upload

If you initiate a request to upload an object to a versioning-enabled bucket, OSS generates a unique version ID for the uploaded object and includes the version ID in the response as the value of the x-oss-version-id header. If you upload an object to a versioning-suspended bucket, OSS generates a version ID of null for the object. If the object that you upload has the same name as an existing object, the existing object is overwritten by the uploaded object. This way, each object has only a single version whose version ID is null.

The following code provides an example on how to upload an object to a versioned bucket by using simple upload:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console .
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Upload an object.
result = bucket.put_object('<yourObjectName>', 'content of object')
# Display the returned HTTP status code.
print('http response code: {}'.format(result.status))
# Display the version ID of the uploaded object.
print('put object version:', result.versionid)
```

Append upload

In a versioning-enabled bucket, the AppendObject operation can be performed only on the current version of an appendable object.

Note

- When you perform the AppendObject operation on an object whose current version is an appendable object, OSS does not generate a previous version for the appendable object.
- When you perform the PutObject or DeleteObject operation on an object whose current version is an appendable object, OSS stores the appendable object as a previous version and prevents the object from being further appended.
- The AppendObject operation cannot be performed on the current version of a non-appendable object such as a normal object or a delete marker.

The following code provides an example on how to upload an object to a versioned bucket by using append upload:

```
# -*- coding: utf-8 -*-
import oss2

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console .

auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set the position from which the first append operation starts to 0.
result = bucket.append_object('<yourObjectName>', 0, 'content of first append')
# Display the version ID of the object that is appended.
print('append object versionid:', result.versionid)
# If you have appended content to the object, you can obtain the position from which the operation starts this time from the next_position field in the response returned by the last operation or by using the bucket.head_object method.
bucket.append_object('<yourObjectName>', result.next_position, 'content of second append')
```

Multipart upload

If you call CompleteMultipartUpload to complete the multipart upload task for an object in a versioning-enabled bucket, OSS generates a unique version ID for the object and returns the version ID as the value of the x-oss-version-id header in the response.

The following code provides an example on how to upload an object to a versioned bucket by using multipart upload:

```
# -*- coding: utf-8 -*-
import os
import oss2
from oss2 import SizedFileAdapter, determine_part_size
from oss2.models import PartInfo

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console .

auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
key = '<yourObjectName>'
filename = '<yourLocalFile>'
total_size = os.path.getsize(filename)
# Specify the determine_part_size method to determine the part size.
part_size = determine_part_size(total_size, preferred_size=100 * 1024)
# Initiate a multipart upload task.
upload_id = bucket.init_multipart_upload(key).upload_id
parts = []
# Upload the parts one by one.
with open(filename, 'rb') as fileobj:
    part_number = 1
    offset = 0
    while offset < total_size:
        num_to_upload = min(part_size, total_size - offset)
        # The SizedFileAdapter(fileobj, size) method generates a new object and recalculates the position from which the append operation starts.
        result = bucket.upload_part(key, upload_id, part_number,
                                   SizedFileAdapter(fileobj, num_to_upload))
        parts.append(PartInfo(part_number, result.etag))
        offset += num_to_upload
        part_number += 1
# Complete the multipart upload task.
result = bucket.complete_multipart_upload(key, upload_id, parts)
# Display the version ID of the uploaded object, which is returned in the response.
print('result.versionid:', result.versionid)
# Verify the result of the multipart upload task.
with open(filename, 'rb') as fileobj:
    assert bucket.get_object(key).read() == fileobj.read()
```

1.4.2.9.3. Download objects

By default, when you call the GetObject operation to download an object in a versioned bucket, only the current version of the object is returned.

When you call the GetObject operation to download an object in a bucket, the returned result is different based on the actual scenarios:

- If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404 Not Found.
- If the version ID of the object is specified in the request, OSS returns the specified version of the object. If the version ID is set to null in the request, OSS returns the version whose version ID is null.
- If the version ID specified in the request is the version ID of a delete marker, OSS returns 405 Method Not Allowed.

The following code provides an example on how to download an object:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Download the specified version of the object.
params = dict()
params['versionId'] = '<yourObjectVersionId>'
object_stream = bucket.get_object('<yourObjectName>', params=params)
# Read the content of the downloaded object.
read_content = object_stream.read()
print('get object content:', read_content)
# Display the version ID of the downloaded object.
print('get object versionid:', object_stream.versionid)
# You must call read() to read the object from the stream returned by get_object before you can calculate the CRC-64 checksum of the object.
if object_stream.client_crc != object_stream.server_crc:
    print "The CRC checksum between client and server is inconsistent!"
```

1.4.2.9.4. Copy objects

This topic describes how to copy objects in a versioned bucket. You can call CopyObject to copy an object that is up to 1 GB in size and call UploadPartCopy to copy an object larger than 1 GB.

Copy a small object

You can call CopyObject to copy an object that is up to 1 GB in size from a source bucket to a destination bucket within the same region.

- By default, the x-oss-copy-source header in a CopyObject request specifies the current version of the object to copy. If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404, which indicates that the object does not exist. You can add a version ID to x-oss-copy-source to copy the specified object version. Delete markers cannot be copied.
- You can copy a previous version of an object to the same bucket. The copied previous version becomes the new version of the object. This way, the previous version of the object is recovered.
- If versioning is enabled for the destination bucket, OSS generates a unique version ID for the destination object. The version ID is returned in the response as the x-oss-version-id header value. If versioning is disabled or suspended for the destination bucket, OSS generates a version whose ID is null for the destination object and overwrites the original version whose ID is null.
- Appendable objects in a bucket for which versioning is enabled or suspended cannot be copied.

The following code provides an example on how to copy a small object:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Copy the specified version of the object.
params = dict()
params['versionId'] = '<yourObjectVersionId>'
result = bucket.copy_object('<yourBucketName>', '<yourSourceObjectName>', '<yourDestinationObjectName>', params=params)
# Display the version ID of the destination object.
print('result.versionid:', result.versionid)
```

Copy a large object

To copy an object that is larger than 1 GB in size, you must split the object into parts and copy the parts by calling UploadPartCopy.

By default, the UploadPartCopy operation uploads a part by copying data from the current version of an existing object. You can add versionId to the x-oss-copy-source request header as the condition to upload a part by copying data from the specified version of an existing object. Example: x-oss-copy-source: /SourceBucketName/SourceObjectName?versionId=111111.

 **Note** The value of SourceObjectName must be URL-encoded. The version ID of the copied object version is returned as the x-oss-copy-source-version-id header value in the response.

If versionId is not specified in the request and the current version of the source object is a delete marker, OSS returns 404 Not Found. If versionId is specified in the request and the specified version of the source object is a delete marker, OSS returns 400 Bad Request.

The following code provides an example on how to use multipart copy to copy a large object:

```

# -*- coding: utf-8 -*-
import os
import oss2
from oss2 import determine_part_size
from oss2.models import PartInfo
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
src_object_name = '<yourSrcObject>'
dest_object_name = '<yourDestObject>'
params = dict()
params['versionId'] = '<yourObjectVersionId>'
# Query the size of the specified version of the source object.
head_info = bucket.head_object(src_object_name, params=params)
total_size = head_info.content_length
print('src object size:', total_size)
# Use the determine_part_size method to determine the part size.
part_size = determine_part_size(total_size, preferred_size=100 * 1024)
print('part size:', part_size)
# Initiate a multipart upload task.
upload_id = bucket.init_multipart_upload(dest_object_name).upload_id
parts = []
# Upload the parts one by one.
part_number = 1
offset = 0
while offset < total_size:
    num_to_upload = min(part_size, total_size - offset)
    end = offset + num_to_upload - 1;
    result = bucket.upload_part_copy(bucket.bucket_name, src_object_name, (offset, end), dest_object_name, upload_id, part_number, params=params)
    # Save the part information.
    parts.append(PartInfo(part_number, result.etag))
    offset += num_to_upload
    part_number += 1
# Complete the multipart upload task.
result = bucket.complete_multipart_upload(dest_object_name, upload_id, parts)
# Display the version ID of the destination object.
print('result version id:', result.versionid)
# Query the metadata of the destination object.
head_info = bucket.head_object(dest_object_name)
# Display the size of the destination object.
dest_object_size = head_info.content_length
print('dest object size:', dest_object_size)
# Compare the size of the destination object with that of the source object.
assert dest_object_size == total_size

```

1.4.2.9.5. List objects

This topic describes how to list objects in a versioned bucket, including all objects, a specified number of objects, and objects whose names contain a specified prefix.

List the versions of all objects in a bucket

The following code provides an example on how to list the versions of all objects including delete markers in a specified bucket:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in Object Storage Service (OSS) is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Call the list_object_versions operation to list the versions of objects in a versioned bucket.
# List the versions of all objects including delete markers in the bucket.
result = bucket.list_object_versions()
# List the versions of all objects in the bucket.
next_key_marker = None
next_versionid_marker = None
while True:
    result = bucket.list_object_versions(key_marker=next_key_marker, versionid_marker=next_versionid_marker)
    # Display the versions of the listed objects.
    for version_info in result.versions:
        print('version_info.versionid:', version_info.versionid)
        print('version_info.key:', version_info.key)
        print('version_info.is_latest:', version_info.is_latest)
    # Display the versions of the listed delete markers.
    for del_maker_info in result.delete_marker:
        print('del_maker.key:', del_maker_info.key)
        print('del_maker.versionid:', del_maker_info.versionid)
        print('del_maker.is_latest:', version_info.is_latest)
    is_truncated = result.is_truncated
    # Check whether the versions of all objects are listed. If the versions of all objects are incompletely listed, the list operation continues. If the versions of all objects are completely listed, the list operation stops.
    if is_truncated:
        next_key_marker = result.next_key_marker
        next_versionid_marker = result.next_versionid_marker
    else:
        break
```

List the versions of the objects whose names contain a specified prefix

The following code provides an example on how to list the versions of the objects whose names contain a specified prefix:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console .
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Call the list_object_versions operation to list the versions of objects in a versioned bucket.
# List the versions of all objects including delete markers in the bucket.
result = bucket.list_object_versions()
# List the versions of the objects whose names contain the test- prefix.
prefix = 'test-'
next_key_marker = None
next_versionid_marker = None
while True:
    result = bucket.list_object_versions(prefix=prefix, key_marker=next_key_marker, versionid_marker=next_versionid_marker)
    # Display the versions of the listed objects.
    for version_info in result.versions:
        print('version_info.versionid:', version_info.versionid)
        print('version_info.key:', version_info.key)
        print('version_info.is_latest:', version_info.is_latest)
    # Display the versions of the listed delete markers.
    for del_maker_info in result.delete_marker:
        print('del_maker.key:', del_maker_info.key)
        print('del_maker.versionid:', del_maker_info.versionid)
        print('del_maker.is_latest:', version_info.is_latest)
    is_truncated = result.is_truncated
    # Check whether the versions of all objects are listed. If the versions of all objects are incompletely listed, the list operation continues. If the versions of all objects are completely listed, the list operation stops.
    if is_truncated:
        next_key_marker = result.next_key_marker
        next_versionid_marker = result.next_versionid_marker
    else:
        break
```

List the versions of a specified number of objects

The following code provides an example on how to list the versions of a specified number of objects:

```
# -*- coding: utf-8 -*-
import oss2

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Call the list_object_versions operation to list the versions of objects in a versioned bucket.
# List the versions of all objects including delete markers in the bucket.
result = bucket.list_object_versions()
# List up to 200 object versions.
max_keys = 200
result = bucket.list_object_versions(max_keys=max_keys)
# Display the versions of the listed objects.
for version_info in result.versions:
    print('version_info.versionid:', version_info.versionid)
    print('version_info.key:', version_info.key)
    print('version_info.is_latest:', version_info.is_latest)
# Display the versions of the listed delete markers.
for del_maker_info in result.delete_marker:
    print('del_maker.key:', del_maker_info.key)
    print('del_maker.versionid:', del_maker_info.versionid)
    print('del_maker.is_latest:', version_info.is_latest)
# Check whether the listed results are truncated.
# If the number of objects in the bucket is greater than 200, the value of is_truncated is True, which indicates that the listed results are truncated. If the number of objects in the bucket is less than 200, the value of is_truncated is False, which indicates that the listed results are not truncated.
print('is truncated', result.is_truncated)
```

List objects by directory

OSS uses a flat structure instead of a hierarchical structure to store objects. A directory is an object whose size is 0 and whose name ends with a forward slash (/). You can upload and download this object. By default, objects whose names end with a forward slash (/) are displayed as directories in the OSS console.

You can specify the delimiter and prefix parameters to list objects by directory.

- If you set prefix to a directory name in the request, objects and subdirectories whose names contain the prefix are listed.
- If you also set delimiter to a forward slash (/) in the request, objects and subdirectories whose names start with the specified prefix in the directory are listed. Each subdirectory is listed as a single result element in CommonPrefixes. The objects and directories in these subdirectories are not listed.

Example: A bucket named examplebucket contains the following objects: *oss.jpg*, *fun/test.jpg*, *fun/movie/001.avi*, and *fun/movie/007.txt*. The forward slash (/) is used as the directory delimiter. The following structure shows the objects and directories in the bucket:

```
examplebucket
├── oss.jpg
├── fun
│   ├── test.jpg
│   └── movie
│       ├── 001.avi
│       └── 007.txt
```

The following code provides examples on how to list objects by directory.

- List the versions of objects within the root directory of a bucket
 - The following code provides an example on how to list the versions of objects within the root directory of a bucket:

```
# -*- coding: utf-8 -*-
import oss2

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Call the list_object_versions operation to list the versions of objects in a versioned bucket.
# List the versions of all objects including delete markers in the bucket.
result = bucket.list_object_versions()
# Set delimiter to a forward slash (/).
delimiter = "/"
next_key_marker = None
next_versionid_marker = None
while True:
    result = bucket.list_object_versions(delimiter=delimiter, key_marker=next_key_marker, versionid_marker=next_versionid_marker)
    # Display the versions of the listed objects.
    for version_info in result.versions:
        print('version_info.versionid:', version_info.versionid)
        print('version_info.key:', version_info.key)
        print('version_info.is_latest:', version_info.is_latest)
    # Display the versions of the listed delete markers.
    for del_maker_info in result.delete_marker:
        print('del_maker.key:', del_maker_info.key)
        print('del_maker.versionid:', del_maker_info.versionid)
        print('del_maker.is_latest:', version_info.is_latest)
    # Display the directories whose names end with a forward slash (/).
    for common_prefix in result.common_prefix:
        print("common_prefix:", common_prefix)
    is_truncated = result.is_truncated
    # Check whether the versions of all objects are listed. If the versions of all objects are incompletely listed, the list operation continues. If the versions of all objects are completely listed, the list operation stops.
    if is_truncated:
        next_key_marker = result.next_key_marker
        next_versionid_marker = result.next_versionid_marker
    else:
        break
```

Returned results

```
('version_info.versionid:', 'CAEQEhiBgMCw8Y7FqBciIGIzMDz3MTEzOWRiMDRmZmFhMmRlMjIjZWlOMWU4****')
('version_info.key:', 'oss.jpg')
('version_info.is_latest:', True)
('common_prefix:', 'fun/')
```

- List objects and subdirectories in a specified directory

The following code provides an example on how to list objects and subdirectories within a directory in a bucket:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Call the list_object_versions operation to list the versions of objects in a versioned bucket.
# List the versions of all objects including delete markers in the bucket.
result = bucket.list_object_versions()
# Set delimiter to a forward slash (/) and prefix to fun/.
prefix = "fun/"
delimiter = "/"
next_key_marker = None
next_versionid_marker = None
while True:
    result = bucket.list_object_versions(prefix=prefix, delimiter=delimiter, key_marker=next_key_marker, versionid_marker=next_versionid_marker)
    # Display the versions of the listed objects.
    for version_info in result.versions:
        print('version_info.versionid:', version_info.versionid)
        print('version_info.key:', version_info.key)
        print('version_info.is_latest:', version_info.is_latest)
    # Display the versions of the listed delete markers.
    for del_maker_info in result.delete_marker:
        print('del_maker.key:', del_maker_info.key)
        print('del_maker.versionid:', del_maker_info.versionid)
        print('del_maker.is_latest:', version_info.is_latest)
    # Display the directories whose names end with a forward slash (/).
    for common_prefix in result.common_prefix:
        print("common_prefix:", common_prefix)
    is_truncated = result.is_truncated
    # Check whether the versions of all objects are listed. If the versions of all objects are incompletely listed, the list operation continues. If the versions of all objects are completely listed, the list operation stops.
    if is_truncated:
        next_key_marker = result.next_key_marker
        next_versionid_marker = result.next_versionid_marker
    else:
        break
```

Returned results

```
('version_info.versionid:', 'CAEQFRiBgMCh9JDkrxcIIGE30TNkYzPhYTc2YzZzOTQ4Y2MzYjg2YjQ4ODg*****')
('version_info.key:', 'fun/test.jpg')
('version_info.is_latest:', True)
('commonPrefix:', 'fun/movie/')

```

1.4.2.9.6. Delete objects

This topic describes how to delete a single object, multiple objects, or objects whose names contain a specified prefix from a versioned bucket.

Delete operations on a versioned bucket

When you delete an object from a versioned bucket, you must determine whether to specify a version ID in the request.

- Delete an object without specifying a version ID (temporary deletion)

By default, if you do not specify the version ID of the object that you want to delete in the request, Object Storage Service (OSS) does not delete the current version of the object but adds a delete marker to the object as the new current version. If you perform the `GetObject` operation on the object, OSS identifies the current version of the object as a delete marker and returns `404 Not Found`. In addition, the `x-oss-delete-marker` header and the `x-oss-version-id` header that indicates the version ID of the created delete marker are included in the response.

The value of the `x-oss-delete-marker` header in the response is true, which indicates that the `x-oss-version-id` value is the version ID of a delete marker.

- Delete an object by specifying a version ID (permanent deletion)

If you specify the version ID of the object that you want to delete in the request, OSS permanently deletes the version specified by the `versionId` field in the `params` parameter. To delete a version whose ID is null, add `params['versionId'] = "null"` to `params` in the request. OSS identifies the string "null" as the ID of the version to delete and deletes the version whose ID is null.

Delete a single object

The following examples show how to permanently or temporarily delete a single object from a versioned bucket.

- Permanently delete an object from a versioned bucket

The following code provides an example on how to permanently delete an object from a versioned bucket by specifying the version ID of the object in the request:

```
# -*- coding: utf-8 -*-
import os
import oss2

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.

auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Specify the full path of the object. The full path of the object cannot contain the bucket name. Example: example/test.txt.
object_name = '<yourObjectName>'
# Specify the version ID of the object or delete marker that you want to delete.
params = dict()
params['versionId'] = '<yourObjectVersionIdOrDeleteMarkerVersionId>'
# Delete the object with the specified version ID or the object whose current version is a delete marker with the specified ID.
result = bucket.delete_object(object_name, params=params)
print("delete object name: ", object_name)
# If the version ID of an object is specified in the request, the value of the delete_marker header in the response is None and the value of the versionId header in the response is the version ID specified in the request.
# If the version ID of a delete marker is specified in the request, the value of the delete_marker header in the response is True and the value of the versionId header in the response is the version ID specified in the request.
if result.delete_marker:
    print("delete del-marker versionid: ", result.versionid)
else:
    print("delete object versionid:", result.versionid)
```

- Temporarily delete an object from a versioned bucket

The following code provides an example on how to temporarily delete an object from a versioned bucket by sending a request in which no version ID is specified:

```
# -*- coding: utf-8 -*-
import os
import oss2

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.

auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
object_name = '<yourObjectName>'
# Temporarily delete an object without specifying a version ID. A delete marker is added to the object.
result = bucket.delete_object(object_name)
# Display the delete marker added to the object.
print("delete marker: ", result.delete_marker)
# Display the version ID of the added delete marker.
print("delete marker versionid: ", result.versionid)
```

Delete multiple objects

The following examples show how to permanently or temporarily delete multiple objects from a versioned bucket.

- Permanently delete multiple objects from a versioned bucket

The following code provides an example on how to permanently delete multiple objects or delete markers with specified version IDs from a versioned bucket:


```
# -*- coding: utf-8 -*-
import os
import oss2
from oss2.models import BatchDeleteObjectVersion
from oss2.models import BatchDeleteObjectVersionList
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
version_list = BatchDeleteObjectVersionList()
# Specify the version IDs of the objects or delete markers that you want to delete.
obj1_versionid = '<yourObject1VersionId>'
obj1_del_marker_versionid = '<yourObject1DelMarkerVersionId>'
obj2_versionid = '<yourObject2VersionId>'
obj2_del_marker_versionid = '<yourObject2DelMarkerVersionId>'
version_list.append(BatchDeleteObjectVersion(key='<yourObject1Name>', versionid=obj1_versionid))
version_list.append(BatchDeleteObjectVersion(key='<yourObject1Name>', versionid=obj1_del_marker_versionid))
version_list.append(BatchDeleteObjectVersion(key='<yourObject2Name>', versionid=obj2_versionid))
version_list.append(BatchDeleteObjectVersion(key='<yourObject2Name>', versionid=obj2_del_marker_versionid))
# Delete the objects or delete markers with the specified version IDs.
result = bucket.delete_object_versions(version_list)
# Display the version IDs of the deleted objects or delete markers.
for del_version in result.delete_versions:
    print('del object name:', del_version.key)
    # Check whether the deleted objects are delete markers.
    print('Is del marker:', del_version.delete_marker)
    # If a deleted object is a delete marker, display the version ID of the delete marker. If the deleted object is not a delete marker, display the version ID of the deleted object.
    if del_version.delete_marker:
        print('del object del_marker.versionid', del_version.delete_marker_versionid)
    else:
        print('del object versionid:', del_version.versionid)
```

- Temporarily delete multiple objects from a versioned bucket

The following code provides an example on how to delete multiple objects from a versioned bucket by sending a request in which no version IDs are specified:

```
# -*- coding: utf-8 -*-
import os
import oss2
from oss2.models import BatchDeleteObjectVersion
from oss2.models import BatchDeleteObjectVersionList
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
key_list = ['<yourObject1Name>', '<yourObject2Name>']
# If you delete an object without specifying a version ID in the request, a delete marker is added to the object.
result = bucket.batch_delete_objects(key_list)
for del_version in result.delete_versions:
    print('key name:', del_version.key)
    # Display the returned delete markers.
    print('Is del marker:', del_version.delete_marker)
    print('key del_marker.versionid', del_version.delete_marker_versionid)
```

Delete objects whose names contain a specified prefix

The following code provides an example on how to delete objects whose names contain a specified prefix:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
prefix = "<yourKeyPrefix>"
# List the version IDs of the objects whose names contain the specified prefix and delete these objects.
next_key_marker = None
next_versionid_marker = None
while True:
    result = bucket.list_object_versions(prefix=prefix, key_marker=next_key_marker, versionid_marker=next_versionid_marker)
    for version_info in result.versions:
        bucket.delete_object(version_info.key, params={'versionId': version_info.versionid})
    for del_marker_info in result.delete_marker:
        bucket.delete_object(del_marker_info.key, params={'versionId': del_marker_info.versionid})
    is_truncated = result.is_truncated
    if is_truncated:
        next_key_marker = result.next_key_marker
        next_versionid_marker = result.next_versionid_marker
    else:
        break
```

1.4.2.9.7. Obtain object metadata

By default, when you call the `HeadObject` operation on an object in a versioned bucket, only the metadata of the current version of the object is returned.

Note If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404 Not Found. If you specify the version ID in the request, OSS returns the metadata of the specified version of the object.

The following code provides an example on how to query object metadata:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Add the versionId field to param.
params = dict()
params['versionId'] = '<yourObjectVersionId>'
# Query parts of the object metadata.
simplifiedmeta = bucket.get_object_meta("<yourObjectName>", params=params)
print(simplifiedmeta.headers['Last-Modified'])
print(simplifiedmeta.headers['Content-Length'])
print(simplifiedmeta.headers['ETag'])
# Query all object metadata.
objectmeta = bucket.head_object("<yourObjectName>", params=params)
print(objectmeta.headers['Content-Type'])
print(objectmeta.headers['Last-Modified'])
print(objectmeta.headers['x-oss-object-type'])
```

1.4.2.9.8. Manage the ACL of an object

This topic describes how to manage the access control lists (ACLs) of objects in a versioned bucket.

Configure the ACL of an object

By default, the `PutObjectACL` operation configures the ACL of the current version of an object. If the current version of the object is a delete marker, Object Storage Service (OSS) returns 404 Not Found. You can specify a version ID in the request to configure the ACL of a specified version of an object.

The following code provides an example on how to configure the ACL of an object:

```
# -*- coding: utf-8 -*-
import oss2
# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Change the ACL of the specified version of an object. In this example, the ACL of the specified version of an object is changed to public read.
params = dict()
params['versionId'] = '<yourObjectVersionId>'
result = bucket.put_object_acl("<yourObjectName>", oss2.OBJECT_ACL_PUBLIC_READ, params = params)
# Query the version ID of the object whose ACL is changed.
print('set acl object versionid:', result.versionid)
```

Query the ACL of an object

By default, the `GetObjectACL` operation queries the ACL of the current version of an object. If the current version of the object is a delete marker, OSS returns 404 Not Found. You can specify a version ID in the request to query the ACL of the specified version of an object.

The following code provides an example on how to query the ACL of an object:

```
# -*- coding: utf-8 -*-
import oss2

# The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Query the ACL of the specified version of the object.
params = dict()
params['versionId'] = '<yourObjectVersionId>'
result = bucket.get_object_acl('<yourObjectName>', params = params)
# Display the ACL of the specified version of the object.
print('get object acl :', result.acl)
# Display the version ID of the object whose ACL is queried.
print('object version id:', result.versionid)
```

1.4.2.10. Authorized access

This topic describes how to use Security Token Service (STS) and a signed URL to authorize access.

Use STS to authorize temporary access

Alibaba Cloud STS is a web service that provides a temporary access token to a cloud computing user. You can use STS to grant a third-party application or your RAM user an access credential with a customized validity period and permissions.

STS has the following benefits:

- You need only to generate an access token and send the access token to a third-party application, instead of exposing your long-term AccessKey pair to the third-party application. You can customize the access permissions and validity period of this token.
- The access token automatically expires when the validity period ends.

First, run the following command to install the official Python STS client:

```
pip install aliyun-python-sdk-sts
```

The Python version must be 2.0.6 or later. The following code provides an example on how to use STS to authorize temporary access to upload an object:

```
# -*- coding: utf-8 -*-
from aliyunsdkcore import client
from aliyunsdksts.request.v20150401 import AssumeRoleRequest
import json
import oss2

# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
endpoint = 'oss-cn-hangzhou.aliyuncs.com'
access_key_id = '<yourAccessKeyId>'
access_key_secret = '<yourAccessKeySecret>'
bucket_name = '<yourBucketName>'
# The role_arn parameter specifies the resource descriptor of the role.
role_arn = '<yourRoleArn>'
clt = client.AcsClient(access_key_id, access_key_secret, 'cn-hangzhou')
req = AssumeRoleRequest.AssumeRoleRequest()
# Set the format of the returned value to JSON.
req.set_accept_format('json')
req.set_RoleArn(role_arn)
req.set_RoleSessionName('session-name')
body = clt.do_action(req)
# Use the AccessKey pair of the RAM user to apply for a temporary token from STS.
token = json.loads(body)
# Initialize the StsAuth instance based on the authentication information in the temporary token.
auth = oss2.StsAuth(token['Credentials']['AccessKeyId'],
                    token['Credentials']['AccessKeySecret'],
                    token['Credentials']['SecurityToken'])
# Initialize the bucket based on the StsAuth instance.
bucket = oss2.Bucket(auth, endpoint, bucket_name)
# Upload a string.
bucket.put_object('object-name.txt', b'hello world')
```

For the complete sample code of STS usage, visit [GitHub](#).

Download a private object by using a signed URL

You can generate a signed URL for users to access a private bucket.

```
# -*- coding: utf-8 -*-
import oss2

auth = oss2.Auth('Your AccessKey ID', 'Your AccessKey secret')
bucket = oss2.Bucket(auth, 'Your endpoint', 'Your bucket name')
print(bucket.sign_url('GET', 'object-in-bucket.txt', 60))
```

A signed URL is displayed after you run the preceding code. You can share the URL with other users so that they can download the objects in the bucket by accessing the URL through a browser or by using tools such as wget. The URL is valid only within 60 seconds after it is generated.

Use a signed URL to authorize temporary access

You can generate a signed URL and provide the URL to a visitor for temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of time during which the visitor can access OSS.

The following code provides an example on how to use a signed URL to download an object:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
# Set the validity period of the signed URL to 60 seconds.
print(bucket.sign_url('GET', '<yourObjectName>', 60))
```

1.4.2.11. Chinese characters and time

This topic describes how to use Chinese characters and time in OSS SDK for Python.

Chinese characters

To use Chinese characters in Python code, you must include a character encoding declaration in the beginning of the code. Otherwise, errors occur when you run the code. Example of a character encoding statement:

```
# -*- coding: utf-8 -*-
```

- Date types

The following table describes the data types supported by Python 2.x.

Date type	Description
str	The string type. Corresponds to the bytes type in Python 3.x.
unicode	Unicode streams. The length is calculated based on the number of Chinese characters. For example, the length of <code>u'中文'</code> is 2.

The following table describes the data types supported by Python 3.x.

Data type	Description
str	The string type. Corresponds to the unicode type in Python 2.x.
bytes	Byte streams. The length is calculated based on the number of bytes. For example, the length of <code>b'中文'</code> is calculated based on the encoding mode. If UTF-8 is used, the length is 6 bytes.

- Input and output limits

The following table describes the limits on input data.

Input data	Type	Description
OSS object name	str	If the input data type is bytes, it must be encoded in UTF-8.
Local file name	str, unicode	If the input data type is bytes, it must be encoded in UTF-8. Example: the <code>yourLocalFile</code> parameter value used in <code>bucket.get_object_to_file</code> .
Input data stream	bytes	Example: data in <code>bucket.put_object</code> .

The following table describes limits on output data.

Output data	Type	Description
The result obtained by parsing the XML document	str	Example: a string in the result obtained by using <code>bucket.list_object</code> .
Downloaded content	bytes	By default, data of the bytes type used by OSS SDK for Python must be encoded in UTF-8. Therefore, ensure that the source Python file is encoded in UTF-8.

- Type conversion functions

OSS SDK for Python provides three functions for type conversion:

Function	Description

Function	Description
to_bytes	- Converts the unicode type to the str type in Python 2.x. Data of other types is returned without changes. - Converts the str type to the bytes type in Python 3.x. Data of other types is returned without changes.
to_unicode	- Converts the str type to the unicode type in Python 2.x. Data of other types is returned without changes. - Converts the bytes type to the str type in Python 3.x. Data of other types is returned without changes.
to_string	This function is equivalent to to_bytes in Python 2.x. This function is equivalent to to_unicode in Python 3.x.

Time

OSS SDK for Python converts timestamp strings of the `datetime.datetime` type that are obtained from the server to a Unix timestamp corresponding to the amount of time in seconds elapsed since UTC 00:00 on January 1, 1970. For example, the `last_modified` parameter returned for `bucket.get_object` is an int Unix timestamp.

You can use the `datetime.datetime.fromtimestamp()` method to convert time to obtain timestamp strings.

1.4.2.12. IMG

OSS Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service that can process large amounts of data. After you upload source images to OSS, you can call `RESTful` API operations to process the images anytime, anywhere, and on any Internet device. IMG provides image processing API operations. To upload images, use the upload API operations provided by OSS. You can use OSS IMG to build your own image processing service.

Basic features

OSS IMG provides the following features:

- Query image information.
- Convert image formats.
- Resize, crop, and rotate images.
- Apply image effect.
- Add image, text, and text-and-image watermarks to images.
- Customize image processing styles. Log on to the OSS console. Click a bucket name to go to the bucket details page. Click the **Image Processing (IMG)** tab and click **Create Rule** to customize an image processing style.
- Call multiple IMG features by using cascade processing.

Usage

You can use standard `HTTP GET` requests to access IMG. All processing parameters are encoded in the query string of a `URL`.

- Anonymous access

If the ACL of an image object is `public-read`, as described in the following table, anonymous users can access the image object.

Bucket ACL	Object ACL
public-read or public-read-write	default
Any ACL	public-read or public-read-write

You can access IMG anonymously by using a third-level domain in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=image/action,param_value
```

- bucket: the name of the bucket.
- endpoint: the domain name used to access the region.
- object: the image object stored in OSS.
- image: the identifier reserved by IMG.
- action: the operation performed on the image, such as resizing, cropping, or rotating.
- param: the parameter corresponding to the operation performed on the image.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

If you process an image by using a custom style, you can access IMG anonymously through a third-level domain in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=x-oss-process=style/name
```

- style: the identifier reserved by the system for the custom style.
- name: the name of the custom style. The name is specified by the Rule Name parameter when you create the custom style in the console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

You can use cascade processing to perform multiple operations in sequence on an image by using a URL in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=image/action,param_value/action,param_value/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

IMG also supports HTTPS access. The following code provides an example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Authorized access

If the ACL of an image object is private, you must have permissions on the image object before you can perform operations on the image object.

Bucket ACL	Object ACL
private	default
Any permission	private

The following code provides an example on how to generate a signed URL to access IMG:

```
# -*- coding: utf-8 -*-
import oss2
endpoint = '<The OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>'
access_key_id = '<access_key_id>'
access_key_secret = '<access_key_secret>'
bucket_name = '<bucket_name>'
key = 'example.jpg'
# Create a bucket. All object-related API operations can be called based on the bucket.
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret), endpoint, bucket_name)
# Upload a sample image.
bucket.put_object_from_file(key, 'example.jpg')
# Generate a signed URL and set the validity period to 10 minutes.
style = 'image/resize,m_fixed,w_100,h_100/rotate,90'
url = bucket.sign_url('GET', key, 10 * 60, params={'x-oss-process': style})
print(url)
```

 Note

- You can configure Create Style, HTTPS, and Cascading Processing for authorized access.
- The validity period of sign_url is set in seconds.

- Access by using SDKs

You can use `OSS SDKs` to access and process image objects of any ACLs.

 Note

- For the complete IMG code, visit [GitHub](#).
- You can use OSS SDKs to process image objects by configuring Create Style, HTTPS, and Cascading Processing.

- Basic operations

The following code provides an example on how to perform basic IMG operations including obtaining image information, converting image formats, resizing, cropping, rotating, watermarking, and applying effects:

```
# -*- coding: utf-8 -*-
import os
import oss2
endpoint = '<The OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>'
access_key_id = '<access_key_id>'
access_key_secret = '<access_key_secret>'
bucket_name = '<bucket_name>'
key = 'example.jpg'
new_pic = 'example-new.jpg'
# Create a bucket. All object-related API operations can be called based on the bucket.
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret), endpoint, bucket_name)
# Upload a sample image.
bucket.put_object_from_file(key, 'example.jpg')
# Resize the image.
style = 'image/resize,m_fixed,w_100,h_100'
bucket.get_object_to_file(key, new_pic, process=style)
# Crop the image.
style = 'image/crop,w_100,h_100,x_100,y_100,r_1'
bucket.get_object_to_file(key, new_pic, process=style)
# Rotate the image.
style = 'image/rotate,90'
bucket.get_object_to_file(key, new_pic, process=style)
# Sharpen the image.
style = 'image/sharpen,100'
bucket.get_object_to_file(key, new_pic, process=style)
# Add a text watermark to the image.
style = 'image/watermark,text_SGVsbG8g5Zu-54mh5pyN5YghIQ'
bucket.get_object_to_file(key, new_pic, process=style)
# Convert the image format.
style = 'image/format,png'
bucket.get_object_to_file(key, new_pic, process=style)
# Delete the sample image.
bucket.delete_object(key)
# Delete the local file.
os.remove(new_pic)
```

 **Note** `get_object` also supports the IMG feature.

- Custom styles

```
# -*- coding: utf-8 -*-
import os
import oss2
endpoint = '<The OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>'
access_key_id = '<access_key_id>'
access_key_secret = '<access_key_secret>'
bucket_name = '<bucket_name>'
key = 'example.jpg'
new_pic = 'example-new.jpg'
# Create a bucket. All object-related API operations can be called based on the bucket.
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret), endpoint, bucket_name)
# Upload a sample image.
bucket.put_object_from_file(key, 'example.jpg')
# Customize the image style.
style = 'style/oss-pic-style-w-100'
# Process the image.
bucket.get_object_to_file(key, new_pic, process=style)
# Delete the sample image.
bucket.delete_object(key)
# Delete the local file.
os.remove(new_pic)
```

 **Note** `get_object` also supports the IMG feature.

- Cascading processing

```
# -*- coding: utf-8 -*-
import os
import oss2
endpoint = '<The OSS endpoint. Example: http://oss-cn-hangzhou.aliyuncs.com>'
access_key_id = '<access_key_id>'
access_key_secret = '<access_key_secret>'
bucket_name = '<bucket_name>'
key = 'example.jpg'
new_pic = 'example-new.jpg'
# Create a bucket. All object-related API operations can be called based on the bucket.
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret), endpoint, bucket_name)
# Upload a sample image.
bucket.put_object_from_file(key, 'example.jpg')
# Perform cascade processing.
style = 'image/resize,m_fixed,w_100,h_100/rotate,90'
# Process the image.
bucket.get_object_to_file(key, new_pic, process=style)
# Delete the sample image.
bucket.delete_object(key)
# Delete the local file.
os.remove(new_pic)
```

 **Note** `get_object` also supports the IMG feature.

1.4.2.13. Handle exceptions

OssError exceptions of OSS SDK for Python are classified into ClientError, RequestError, and ServerError. Definitions of these errors are provided in the submodule of `oss2.exceptions`.

The following table describes the variable, type, and description of exceptions.

Variable	Type	Description
status	int	• If a ServerError exception occurs, an HTTP status code is returned. • If a ClientError exception or a RequestError exception occurs, the error status is returned.
request_id	str	• If a ServerError exception occurs, the OSS server returns the request ID. • If a ClientError exception or a RequestError exception occurs, the request ID parameter in the response is empty.
code and message	str	This variable indicates the text in the Code and Message tags in an OSS error response.

Examples

The following code provides an example on how to handle an exception and display the HTTP status code of the error when you download an object that does not exist:

```
# -*- coding: utf-8 -*-
import oss2
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucketName>')
try:
    stream = bucket.get_object('random-key.txt')
except oss2.exceptions.NoSuchKey as e:
    print('status={0}, request_id={1}'.format(e.status, e.request_id))
```

ClientError

ClientError occurs due to incorrect inputs on the client. For example, the error occurs when you use `bucket.batch_delete_objects` and receive an empty object list. The status value of ClientError is `oss2.exceptions.OSS_CLIENT_ERROR_STATUS`.

ServerError

When the OSS server returns the HTTP error code, OSS SDK for Python converts the error code to ServerError. ServerError derives multiple subclasses based on the HTTP status code and OSS error code. The NotFound subclass corresponds to 404 HTTP status code. The Conflict subclass corresponds to 409 HTTP status code.

The following table describes common error codes.

Error	HTTP status code	OSS error code	Description
NotModified	304	None	The error message returned because no modifications are performed.
AccessDenied	403	AccessDenied	The error message returned because the access is denied.

Error	HTTP status code	OSS error code	Description
NoSuchBucket	404	NoSuchBucket	The error message returned because the specified bucket does not exist.
NoSuchKey	404	NoSuchKey	The error message returned because the specified object does not exist.
NoSuchUpload	404	NoSuchUpload	The error message returned because the specified multipart upload task does not exist.
NoSuchWebsite	404	NoSuchWebsiteConfiguration	The error message returned because static website hosting is not configured for the specified bucket.
NoSuchLifecycle	404	NoSuchLifecycle	The error message returned because no lifecycle rule is configured for the specified bucket.
NoSuchCors	404	NoSuchCORSConfiguration	The error message returned because CORS rules are not configured for the specified bucket.
BucketNotEmpty	409	BucketNotEmpty	The error message returned because the bucket is not empty.
PositionNotEqualToLength	409	PositionNotEqualToLength	The error message returned because the configured position is not equal to the object length.
ObjectNotAppendable	409	ObjectNotAppendable	The error message returned because the object is not appendable.

1.4.3. PHP-SDK

1.4.3.1. Preface

This topic provides the sample code of OSS SDK for PHP in various use cases.

Source code

For the source code of OSS SDK for PHP, visit [GitHub](#).

Sample codes

OSS SDK for PHP provides a variety of sample code for your reference or use. The following table describes the sample codes.

Sample file	Description
Object.php	Shows you how to perform object-related operations, including upload, download, copy, delete, and list objects as well as object metadata-related operations.
MultipartUpload.php	Shows you how to upload large objects and perform multipart upload.
Signature.php	Shows you how to generate a signed URL to authorize access.
Callback.php	Shows you how to perform upload callback.
Image.php	Shows you how to use Image Processing (IMG).
LiveChannel.php	Shows you how to use LiveChannel.
Bucket.php	Shows you how to manage buckets, including create, delete, and list buckets as well as configure ACL for a bucket.
BucketLifecycle.php	Shows you how to configure, query, and delete the lifecycle rules for a bucket.
BucketLogging.php	Shows you how to configure, query, and delete the logging configurations for a bucket.
BucketReferer.php	Shows you how to configure, query, and delete the hotlink protection configurations for a bucket.
BucketWebsite.php	Shows you how to configure, query, and delete the static website hosting configurations for a bucket.
BucketCors.php	Shows you how to configure, query, and delete the CORS configurations for a bucket.

1.4.3.2. Installation

This topic describes how to install OSS SDK for PHP.

Preparation

OSS SDK for PHP is applicable to PHP 5.3 and later versions. This topic uses PHP 5.6.22 as an example.

- Install the environment

You must install PHP and cURL extension:

- If a message that indicates the specified module is unavailable is displayed, set `extension_dir` to `C:/Windows/System32/`.
- In Ubuntu, run the `sudo apt-get install php-curl` command to install the cURL extension of PHP by using `apt-get`.
- In CentOS, run the `sudo yum install php-curl` command to install the cURL extension of PHP by using `yum`.
- Check the PHP version
 - You can run the `php -v` command to view the current PHP version.
 - You can run the `php -m` command to check whether the cURL extension is installed.

Download OSS SDK for PHP

- [Download from GitHub](#)
- [Download previous versions](#)

For more information, see [API Documentation](#).

Install OSS SDK for PHP

You can use one of the following method to install SDK:

- Composer
 - Go to the root directory of your project and run the `composer require aliyuncs/oss-sdk-php` command or add the following dependency to the `composer.json` file:

```
"require": {
    "aliyuncs/oss-sdk-php": "~2.x.x"
}
```

- Run the `composer install` command to install the dependency. After the dependency is installed, check whether your directory structure complies with the following structure:

```
.
├─ app.php
├─ composer.json
├─ composer.lock
└─ vendor
```

In the preceding directory structure, `app.php` indicates your application, and the `vendor/` directory contains the dependent library. You need to add the following dependency to `app.php`:

```
require_once __DIR__ . '/vendor/autoload.php';
```

Note

- If `autoload.php` is imported to your project, you do not need to import it again after you add the dependency of OSS SDK for PHP.
- If a network error occurs when you use Composer, you can use the [image source](#) of Composer China by running the `composer config -g repositories.packagist composer http://packagist.phpcomposer.com` command.

- PHAR
 - Select the corresponding version and download the packaged PHAR file from [GitHub](#).
 - Import the PHAR file in your code:

```
require_once '/path/to/oss-sdk-php.phar';
```

- Source code
 - Select the required version and download the ZIP package from [GitHub](#).
 - The decompressed root directory contains the `autoload.php` file. Import the file to the code:

```
require_once '/path/to/oss-sdk/autoload.php';
```

1.4.3.3. Initialization

An `OSSClient` serves as the OSS PHP client to manage OSS resources such as buckets and objects.

Create an `OSSClient` instance

To create an `OSSClient` instance, you must specify an endpoint. For more information about endpoints, see [Obtain an endpoint](#).

To create an `OSSClient`, you must specify an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

- Use an endpoint to create an `OSSClient`

The following code provides an example on how to create an `OSSClient` instance based on an OSS endpoint. `OSSClient` supports concurrent tasks. Therefore, a project can contain one or more `OSSClient` instances.

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
} catch (OssException $e) {
    print $e->getMessage();
}
}
```

- Use STS to create an OSSClient

The following code provides an example on how to create an OSSClient instance based on Security Token Service (STS):

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$securityToken = "<yourSecurityToken>";
try {
    $ossClient = new OssClient(
        $accessKeyId, $accessKeySecret, $endpoint, false, $securityToken);
} catch (OssException $e) {
    print $e->getMessage();
}
}
```

- Use a proxy server to create an OSSClient

You can use a proxy server to create an OSSClient when you use OSS SDK for PHP version 5.3 and later. The following code provides an example on how to use a proxy server to create an OSSClient:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// Specify the IP address of the proxy server. Example: http://<username>:<password>@<proxyip>:<proxyport>.
$requestProxy = "<yourRequestProxy>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false, $requestProxy);
} catch (OssException $e) {
    print $e->getMessage();
}
}
```

Configure network parameters

The following code provides an example on how to configure network parameters for an OSSClient:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
try {
    $ossClient = new OssClient(
        $accessKeyId, $accessKeySecret, $endpoint);
    // Set the timeout period for data transmission at the socket layer. Unit: seconds. Default value: 5184000.
    $ossClient->setTimeout(3600);
    // Set the timeout period for connection establishment. Unit: seconds. Default value: 10.
    $ossClient->setConnectTimeout(10);
} catch (OssException $e) {
    print $e->getMessage();
}
```

1.4.3.4. Getting Started

This topic describes how to use Object Storage Service (OSS) SDK for PHP to perform routine operations, such as creating buckets, uploading objects, and downloading objects.

Create a bucket

The following code provides an example on how to create a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->createBucket($bucket);
} catch (OssException $e) {
    print $e->getMessage();
}
```

For more information about how to obtain an AccessKey ID and AccessKey secret, see [Obtain an AccessKey pair](#).

For more information about how to obtain an endpoint, see [Obtain an endpoint](#).

Upload an object

The following code provides an example on how to upload objects to OSS by using stream upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// <yourObjectName> indicates the full path of the object that you want to upload to OSS. <yourObjectName> must include the file extension of the object. Example: abc/efg/123.jpg.
$object = "<yourObjectName>";
$content = "Hi, OSS.";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putObject($bucket, $object, $content);
} catch (OssException $e) {
    print $e->getMessage();
}
```

Download an object

The following code provides an example on how to download an object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// <yourObjectName> indicates the full path of the object that you want to download from OSS. <yourObjectName> must include the file extension of the object. Example: abc/efg/123.jpg.
$object = "<yourObjectName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $content = $ossClient->getObject($bucket, $object);
    print("object content: " . $content);
} catch (OssException $e) {
    print $e->getMessage();
}
```

List objects

The following code provides an example on how to list objects in a bucket. By default, 100 objects are listed.

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $listObjectInfo = $ossClient->listObjects($bucket);
    $objectList = $listObjectInfo->getObjectList();
    if (!empty($objectList)) {
        foreach ($objectList as $objectInfo) {
            print($objectInfo->getKey() . "\t" . $objectInfo->getSize() . "\t" . $objectInfo->getLastModified() . "\n");
        }
    }
} catch (OssException $e) {
    print $e->getMessage();
}
```

Delete an object

The following code provides an example on how to delete an object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// <yourObjectName> indicates the full path of the object that you want to delete from OSS. <yourObjectName> must include the file extension of the object. Example: abc/efg/123.jpg.
$object = "<yourObjectName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteObject($bucket, $object);
} catch (OssException $e) {
    print $e->getMessage();
}
```

1.4.3.5. Buckets

1.4.3.5.1. Create buckets

A bucket is a container used to store objects in Object Storage Service (OSS). Each object is contained in a bucket. This topic describes how to create a bucket.

The following code provides an example on how to create a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // Set the storage class of the bucket to Standard. The default storage class is Standard.
    $options = array(
        OssClient::OSS_STORAGE => OssClient::OSS_STORAGE_Standard
    );
    // Set the access control list (ACL) of the bucket to public read. The default bucket ACL is private.
    $ossClient->createBucket($bucket, OssClient::OSS_ACL_TYPE_PUBLIC_READ, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . " : FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . " : OK" . "\n");
```

For the complete code that is used to create a bucket, visit [GitHub](#).

1.4.3.5.2. List buckets

A bucket is a container that is used to store objects in Object Storage Service (OSS). All objects in OSS are contained in buckets. Bucket names are listed in alphabetical order. You can list buckets that belong to the current Alibaba Cloud account in all regions and meet the specific conditions.

The following sample code provides an example on how to list all buckets that belong to an Apsara Stack tenant account:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $bucketListInfo = $ossClient->listBuckets();
} catch (OssException $e) {
    printf(__FUNCTION__ . " : FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
$bucketList = $bucketListInfo->getBucketList();
foreach($bucketList as $bucket) {
    print($bucket->getLocation() . "\t" . $bucket->getName() . "\t" . $bucket->getCreatedate() . "\n");
}
```

1.4.3.5.3. Determine whether a bucket exists

A bucket is a container used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. This topic describes how to determine whether a bucket exists.

The following code provides an example on how to determine whether a bucket exists:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $res = $ossClient->doesBucketExist($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
if ($res === true) {
    print(__FUNCTION__ . ": OK" . "\n");
} else {
    print(__FUNCTION__ . ": FAILED" . "\n");
}
```

1.4.3.5.4. Manage the ACL of a bucket

A bucket is a container used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. This topic describes how to configure and query the access control list (ACL) of a bucket.

Configure the ACL for a bucket

The following table describes the ACLs that you can configure for a bucket.

ACL	Description	Value
Private	Only the bucket owner and authorized users have read and write permissions on objects in the bucket. Other users cannot access the objects in the bucket.	OssClient::OSS_ACL_TYPE_PRIVATE
Public read	Only the bucket owner and authorized users have read and write permissions on objects in the bucket. Other users have only read permissions on the objects in the bucket. Exercise caution when you set the ACL of the bucket to this value.	OssClient::OSS_ACL_TYPE_PUBLIC_READ
Public read/write	All users have read and write permissions on objects in the bucket. Exercise caution when you set the ACL of the bucket to this value.	OssClient::OSS_ACL_TYPE_PUBLIC_READ_WRITE

The following code provides an example on how to configure the ACL of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// Set the ACL of the bucket to private.
$acl = OssClient::OSS_ACL_TYPE_PRIVATE;
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putBucketAcl($bucket, $acl);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```


Query the ACL of a bucket

The following code provides an example on how to query the ACL of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $res = $ossClient->getBucketAcl($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print('acl: ' . $res);
```

1.4.3.5.5. Manage bucket tagging

You can use tags to identify buckets used for different purposes and manage these buckets.

② Note

- Only the bucket owner or authorized Resource Access Management (RAM) users can configure tagging for the bucket. Otherwise, 403 Forbidden is returned with the error code AccessDenied.
- You can configure up to 20 tags for a bucket. A tag is a key-value pair.
- The key of a tag can be up to 64 bytes in length and cannot start with `http://`, `https://`, or `Aliyun`. The key of a tag cannot be empty.
- The tag value is optional. The tag value can be up to 128 bytes in length. You can leave the parameter empty.
- The key and value of a tag must be UTF-8-encoded.
- When you use PutBucketTags to configure a tag for a bucket, the new tag overwrites the existing tag.

Configure tags for a bucket

The following code provides an example on how to configure tags for a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\TaggingConfig;
use OSS\Model\Tag;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in Object Storage Service (OSS) is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false);
try {
    // Configure tags for the bucket.
    $config = new TaggingConfig();
    $config->addTag(new Tag("key1", "value1"));
    $config->addTag(new Tag("key2", "value2"));
    $ossClient->putBucketTags($bucket, $config);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query the tags of a bucket

The following code provides an example on how to query the tags of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\TaggingConfig;
use OSS\Model\Tag;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is
// a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM
// console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false);
try {
    // Query the tags of the bucket.
    $config = $ossClient->getBucketTags($bucket);
    print_r($config->getTags());
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Delete the tags of a bucket

The following code provides an example on how to delete the tags of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\Tag;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is
// a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM
// console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false);
try {
    // Delete the specific tags that are configured for the bucket.
    $tags = array();
    $tags[] = new Tag("key1", "value1");
    $tags[] = new Tag("key2", "value2");
    $ossClient->deleteBucketTags($bucket, $tags);
    // Delete all tags that are configured for the bucket.
    $ossClient->deleteBucketTags($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.5.6. Delete buckets

If you no longer use a bucket, delete the bucket to stop unnecessary charges.

Before you delete a bucket, ensure that all objects, LiveChannels, and parts that are generated from multipart upload in the bucket are deleted. The following code provides an example on how to delete a bucket:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the bucket name.
$bucket = "<yourBucketName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteBucket($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.5.7. Manage lifecycle rules

Object Storage Service (OSS) allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to Infrequent Access (IA) or Archive to reduce your storage costs. This topic describes how to manage lifecycle rules for buckets.

Background

Each lifecycle rule contains the following information:

- Policy: specifies the mode to match objects and parts.
 - Match by prefix: matches objects and parts by prefix. You can create multiple rules to configure different prefixes. Each prefix must be unique.
 - Match by bucket: matches all objects and parts contained in the bucket. You can create only one rule if you select this method.
- Object lifecycle policy: specifies the validity period or expiration data and the operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired objects are deleted.
 - Expiration date: specifies a date. Objects that are last modified before this date expire and are deleted.
- Part lifecycle policy: specifies the validity period or expiration time and the delete operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted.
 - Expiration date: specifies a date. Parts that are last modified before this date expire and are deleted.

Lifecycle rules also apply to the parts uploaded by using `uploadPart`. In this case, the last modified time of an object is the time the multipart upload task is initiated.

Configure lifecycle rules

The following code provides an example on how to configure lifecycle rules for a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\LifecycleConfig;
use OSS\Model\LifecycleRule;
use OSS\Model\LifecycleAction;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Specify the rule ID and the prefix of the object names that match the rule.
$ruleId0 = "rule0";
$matchPrefix0 = "A0/";
$ruleId1 = "rule1";
$matchPrefix1 = "A1/";
$lifecycleConfig = new LifecycleConfig();
$actions = array();
// Specify that objects expire three days after they are last modified.
$actions[] = new LifecycleAction(OssClient::OSS_LIFECYCLE_EXPIRATION, OssClient::OSS_LIFECYCLE_TIMING_DAYS, 3);
$lifecycleRule = new LifecycleRule($ruleId0, $matchPrefix0, "Enabled", $actions);
$lifecycleConfig->addRule($lifecycleRule);
$actions = array();
// Specify that objects created before the specified date expire.
$actions[] = new LifecycleAction(OssClient::OSS_LIFECYCLE_EXPIRATION, OssClient::OSS_LIFECYCLE_TIMING_DATE, '2022-10-12T00:00:00.000Z');
$lifecycleRule = new LifecycleRule($ruleId1, $matchPrefix1, "Enabled", $actions);
$lifecycleConfig->addRule($lifecycleRule);
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putBucketLifecycle($bucket, $lifecycleConfig);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query lifecycle rules

The following code provides an example on how to query lifecycle rules of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$lifecycleConfig = null;
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $lifecycleConfig = $ossClient->getBucketLifecycle($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print($lifecycleConfig->serializeToXml() . "\n");
```

Delete lifecycle rules

The following code provides an example on how to delete lifecycle rules from a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteBucketLifecycle($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.5.8. Hotlink protection

Object Storage Service (OSS) provides hotlink protection to prevent unauthorized domain names from accessing your OSS data. You can configure the Referer field in the HTTP header.

To configure hotlink protection, you must configure the following parameters:

- **Referer Whitelist:** Only specified domain names are allowed to access OSS resources.
- **Allow Empty Referer:** If you do not allow empty Refer fields, a request is allowed to access OSS resources only if the request includes the Referer field configured in the HTTP or HTTPS header.

Configure hotlink protection

The following code provides an example on how to configure hotlink protection:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\RefererConfig;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$refererConfig = new RefererConfig();
// Allow empty Referers.
$refererConfig->setAllowEmptyReferer(true);
// Add a Referer allowlist. You can use asterisks (*) and question marks (?) as wildcard characters in Referers.
$refererConfig->addReferer("www.aliyun.com");
$refererConfig->addReferer("www.aliyuncs.com");
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putBucketReferer($bucket, $refererConfig);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query the hotlink protection configurations

The following code provides an example on how to query the hotlink protection configurations:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\RefererConfig;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$refererConfig = null;
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $refererConfig = $ossClient->getBucketReferer($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print($refererConfig->serializeToXml() . "\n");
```

Clear the hotlink protection configurations

The following code provides an example on how to clear the hotlink protection configurations:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\RefererConfig;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$refererConfig = new RefererConfig();
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // The hotlink protection configurations of a bucket cannot be directly cleared. You must configure a new hotlink protection rule that allows empty Referer fields to overwrite the existing hotlink protection configurations.
    $ossClient->putBucketReferer($bucket, $refererConfig);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.5.9. Configure logging

You can enable logging to record access to a bucket in log objects, which are stored in a specified bucket.

The log objects are in the following format: <TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString .

Enable logging

The following code provides an example on how to enable logging for a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$options = array();
// Specify the bucket that is used to store log objects.
$targetBucket = "<yourBucketName>";
$targetPrefix = "access.log";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // Enable logging.
    $ossClient->putBucketLogging($bucket, $targetBucket, $targetPrefix, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

View logging configurations

The following code provides an example on how to view the logging configurations of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$loggingConfig = null;
$options = array();
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // View the logging configurations.
    $loggingConfig = $ossClient->getBucketLogging($bucket, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print($loggingConfig->serializeToXml() . "\n");
```

Disable logging

The following code provides an example on how to disable logging for a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // Disable logging.
    $ossClient->deleteBucketLogging($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.5.10. Configure static website hosting

You can configure static website hosting for your bucket. After the configurations take effect, you can use a bucket domain name as a static website address that can be used to access objects stored in the bucket. For example, index pages and error pages.

For the complete code for static website hosting, visit [GitHub](#).

Configure static website hosting

The following code provides an example on how to configure static website hosting:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\WebsiteConfig;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Configure static web hosting. Set the index page to index.html and the error page to error.html.
$websiteConfig = new WebsiteConfig("index.html", "error.html");
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putBucketWebsite($bucket, $websiteConfig);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query static website hosting configurations

The following code provides an example on how to query static website hosting configurations:


```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$websiteConfig = null;
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $websiteConfig = $ossClient->getBucketWebsite($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print($websiteConfig->serializeToXml() . "\n");
```

Delete static website hosting configurations

The following code provides an example on how to delete static website hosting configurations:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteBucketWebsite($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.5.11. Configure CORS

CORS allows web applications to access resources that belong to different regions. OSS supports CORS API operations for cross-origin access control.

For the complete CORS rule code, visit [GitHub](#).

Configure CORS rules

The following code provides an example on how to configure CORS rules for a specific bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\CorsConfig;
use OSS\Model\CorsRule;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$corsConfig = new CorsConfig();
$rule = new CorsRule();
// AllowedHeaders and ExposeHeaders do not support wildcard characters.
$rule->addAllowedHeader("x-oss-header");
// AllowedOlowedMethods allows only one asterisk (*) wildcard character. The asterisk (*) wildcard character specifies that all origins or operations
are allowed.
$rule->addAllowedOrigin("http://www.b.com");
$rule->addAllowedMethod("POST");
$rule->setMaxAgeSeconds(10);
// Up to 10 rules can be configured for each bucket.
$corsConfig->addRule($rule);
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // The existing rules are overwritten.
    $ossClient->putBucketCors($bucket, $corsConfig);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query CORS rules

The following code provides an example on how to query CORS rules that are configured for a specific bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$corsConfig = null;
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $corsConfig = $ossClient->getBucketCors($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print($corsConfig->serializeToXml() . "\n");
```

Delete CORS rules

The following code provides an example on how to delete the CORS rules that are configured for a specific bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteBucketCors($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.6. Upload objects

1.4.3.6.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for PHP provides a variety of methods to upload objects.

You can upload objects to OSS by using one of the following methods:

- **Simple upload:** uploads an object up to 5 GB in size.
- **Append upload:** uploads an object up to 5 GB in size.
- **Multipart upload:** uploads an object up to 48.8 TB in size. Use multipart upload for large objects.

1.4.3.6.2. Simple upload

Simple upload allows you to upload strings or local files. This topic describes how to upload strings and local files.

Upload a string

The following code provides an example on how to upload a string to an OSS object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// Specify the name of the object.
$object = "<yourObjectName>";
// Configure the object content.
$content = "Hello OSS";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putObject($bucket, $object, $content);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
// You can configure headers in upload requests when you upload the string. For example, you can set the object ACL to private and configure the user
metadata of the object.
$options = array(
    OssClient::OSS_HEADERS => array(
        'x-oss-object-acl' => 'private',
        'x-oss-meta-info' => 'your info'
    ),
);
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putObject($bucket, $object, $content, $options);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Upload a local file

The following code provides an example on how to upload a local file to OSS:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// Specify the name of the object.
$object = "<yourObjectName>";
// <yourLocalFile> consists of the local file path and the file name with an extension. Example: /users/local/myfile.txt.
$filePath = "<yourLocalFile>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->uploadFile($bucket, $object, $filePath);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.6.3. Append upload

You can append strings and local files when you use append upload. The CopyObject operation cannot be performed on append objects.

Append strings

The following code provides an example on how to upload strings to OSS by using append upload:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the bucket name.
$bucket = "<yourBucketName>";
// Specify the object name.
$object = "<yourObjectName>";
// Query the content of the object to upload.
$content_array = array('Hello OSS', 'Hi OSS', 'OSS OK');
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // Start the first append. // If the object is appended for the first time, the append position is 0. The return value is the position for the next
    // append. The position to start the next append is the total length of appended bytes and the append object.
    $position = $ossClient->appendObject($bucket, $object, $content_array[0], 0);
    $position = $ossClient->appendObject($bucket, $object, $content_array[1], $position);
    $position = $ossClient->appendObject($bucket, $object, $content_array[2], $position);
} catch (OssException $e) {
    printf(_FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(_FUNCTION__ . ": OK" . "\n");
```

Append local files

The following code provides an example on how to upload local files to OSS by using append upload:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Specify the bucket name.
$bucket = "<yourBucketName>";
// Specify the object name.
$object = "<yourObjectName>";
// Obtain Local File 1.
$filePath = "<yourLocalFile1>";
// Obtain Local File 2.
$filePath1 = "<yourLocalFile2>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $position = $ossClient->appendFile($bucket, $object, $filePath, 0);
    $position = $ossClient->appendFile($bucket, $object, $filePath1, $position);
} catch (OssException $e) {
    printf(_FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(_FUNCTION__ . ": OK" . "\n");
```

1.4.3.6.4. Multipart upload

Object Storage Service (OSS) allows you to use multipart upload to split an object into several parts and upload them separately. After all parts are uploaded, you can combine the uploaded parts into a complete object to finish the upload.

To implement multipart upload, perform the following operations:

For the complete code that is used to perform multipart upload, visit [GitHub](#).

The following code provides an example on how to perform multipart upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
```

```

require_once __DIR__ . '/../vendor/autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Core\OssUtil;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$uploadFile = "<yourLocalFile>";
/**
 * Step 1: Initiate a multipart upload task and obtain the upload ID.
 */
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // An upload ID is returned. It is the unique identifier for a multipart upload task. You can initiate related operations such as canceling or qu
    erting a multipart upload task based on the upload ID.
    $uploadId = $ossClient->initiateMultipartUpload($bucket, $object);
} catch (OssException $e) {
    printf(_FUNCTION_ . ": initiateMultipartUpload FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(_FUNCTION_ . ": initiateMultipartUpload OK" . "\n");
/*
 * Step 2: Upload parts.
 */
$partSize = 10 * 1024 * 1024;
$uploadFileSize = filesize($uploadFile);
$pieces = $ossClient->generateMultiuploadParts($uploadFileSize, $partSize);
$responseUploadPart = array();
$uploadPosition = 0;
$isCheckMd5 = true;
foreach ($pieces as $i => $piece) {
    $fromPos = $uploadPosition + (integer)$piece[$ossClient::OSS_SEEK_TO];
    $toPos = (integer)$piece[$ossClient::OSS_LENGTH] + $fromPos - 1;
    $uploadOptions = array(
        $ossClient::OSS_FILE_UPLOAD => $uploadFile,
        $ossClient::OSS_PART_NUM => ($i + 1),
        $ossClient::OSS_SEEK_TO => $fromPos,
        $ossClient::OSS_LENGTH => $toPos - $fromPos + 1,
        $ossClient::OSS_CHECK_MD5 => $isCheckMd5,
    );
    // Perform the MD5 verification.
    if ($isCheckMd5) {
        $contentMd5 = OssUtil::getMd5SumForFile($uploadFile, $fromPos, $toPos);
        $uploadOptions[$ossClient::OSS_CONTENT_MD5] = $contentMd5;
    }
    try {
        // Upload parts.
        $responseUploadPart[] = $ossClient->uploadPart($bucket, $object, $uploadId, $uploadOptions);
    } catch (OssException $e) {
        printf(_FUNCTION_ . ": initiateMultipartUpload, uploadPart - part#{ $i } FAILED\n");
        printf($e->getMessage() . "\n");
        return;
    }
    printf(_FUNCTION_ . ": initiateMultipartUpload, uploadPart - part#{ $i } OK\n");
}
// $uploadParts is an array that consists of the ETag and PartNumber of each part.
$uploadParts = array();
foreach ($responseUploadPart as $i => $eTag) {
    $uploadParts[] = array(
        'PartNumber' => ($i + 1),
        'ETag' => $eTag,
    );
}
/**
 * Step 3: Complete the multipart upload task.
 */
try {
    // You must provide all valid $uploadParts when you perform this operation. After OSS receives the $uploadParts, OSS verifies all parts one by on
    e. After all parts are verified, OSS combines these parts into a complete object.
    $ossClient->completeMultipartUpload($bucket, $object, $uploadId, $uploadParts);
} catch (OssException $e) {
    printf(_FUNCTION_ . ": completeMultipartUpload FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
printf(_FUNCTION_ . ": completeMultipartUpload OK\n");

```

The following table describes the parameters contained in \$options involved in the preceding example.

Parameter	Description
\$ossClient::OSS_FILE_UPLOAD	Uploads the object.
OssClient::OSS_PART_NUM	The part number.
OssClient::OSS_SEEK_TO	The start position.
OssClient::OSS_LENGTH	The size of the object.
OssClient::OSS_PART_SIZE	Specifies the part size.
OssClient::OSS_CHECK_MD5	Specifies whether to enable MD5 verification. A value of true indicates that the MD5 verification is enabled.

Cancel a multipart upload task

You can call \$ossClient->abortMultipartUpload to cancel a multipart upload task. If you cancel a multipart upload task, you cannot use the upload ID to upload any parts. The uploaded parts are deleted.

The following code provides an example on how to cancel a multipart upload task:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$upload_id = "<yourUploadId>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->abortMultipartUpload($bucket, $object, $upload_id);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Upload a local file by using multipart upload

The following code provides an example on how to upload a local file to OSS by using multipart upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$file = "<yourLocalFile>";
$options = array(
    OssClient::OSS_CHECK_MD5 => true,
    OssClient::OSS_PART_SIZE => 1,
);
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->multiuploadFile($bucket, $object, $file, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Upload a local directory by using multipart upload

The following code provides an example on how to upload a local directory including all files in the directory to OSS by using multipart upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}

if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$localDirectory = ".";
$prefix = "samples/codes";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->uploadDir($bucket, $prefix, $localDirectory);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

List uploaded parts

The following code provides an example on how to list uploaded parts:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}

if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$uploadId = "<yourUploadId>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $listPartsInfo = $ossClient->listParts($bucket, $object, $uploadId);
    foreach ($listPartsInfo->getListPart() as $partInfo) {
        print($partInfo->getPartNumber() . "\t" . $partInfo->getSize() . "\t" . $partInfo->getETag() . "\t" . $partInfo->getLastModified() . "\n");
    }
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

List multipart upload tasks

The following code provides an example on how to list multipart upload tasks:


```

<?php
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$options = array(
    'delimiter' => '/',
    'max-uploads' => 100,
    'key-marker' => '',
    'prefix' => '',
    'upload-id-marker' => ''
);
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $listMultipartUploadInfo = $ossClient->listMultipartUploads($bucket, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": listMultipartUploads FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
printf(__FUNCTION__ . ": listMultipartUploads OK\n");
$listUploadInfo = $listMultipartUploadInfo->getUploads();
var_dump($listUploadInfo);

```

The following table describes the parameters contained in \$options from the preceding sample code.

Parameter	Description
delimiter	The character used to group object names. Objects whose names contain the same string from the prefix and the next occurrence of the delimiter are grouped as a single result element.
key-marker	Specifies the name of the object after which the listing of multipart upload tasks begins. All multipart upload tasks with objects whose names are alphabetically greater than the keyMarker value are listed. You can use key-marker and upload-id-marker together to specify the position after which the next list begins.
max-uploads	The maximum number of multipart upload tasks to return this time. Default value: 1000. Maximum value: 1000.
prefix	The prefix that the names of returned objects must contain. If you use a prefix for a query, the returned object names contain the prefix.
upload-id-marker	The upload ID of the multipart upload task after which the list begins. This parameter is used together with the key-marker parameter. If key-marker is not configured, upload-id-marker is invalid. If key-marker is configured, the query result includes: - All objects whose names are alphabetically greater than the value of keyMarker. - All objects whose names start with the letter the same as the value of key-marker. The value of uploadid is greater than that of upload-id-marker.

1.4.3.7. Download objects

1.4.3.7.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for PHP provides a variety of methods to download objects.

You can use one of the following methods to download objects:

- [Download to local files](#)
- [Download objects to local memory](#)
- [Range download](#)
- [Conditional download](#)

1.4.3.7.2. Download to local files

This topic describes how to use OSS SDK for PHP to download specific objects to local files.

The following code provides an example on how to download a specific object to a local file:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// <yourObjectName> indicates the complete path of the object you want to download from OSS. The path must include the extension of the object name.
Example: abc/efg/123.jpg.
$object = "<yourObjectName>";
// <yourLocalFile> consists of a local file path and a file name with an extension. Example: /users/local/myfile.txt.
$localfile = "<yourLocalFile>";
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $localfile
);
// Use try catch to catch exceptions. If an exception is returned, the download fails. If no exceptions are returned, the object is downloaded.
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->getObject($bucket, $object, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK, please check localfile: 'upload-test-object-name.txt'" . "\n");
```

1.4.3.7.3. Download objects to local memory

This topic describes how to use OSS SDK for PHP to download specified objects to local memory.

The following code provides an example on how to download a specified object to local memory and display the content of the object:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Specify the object name.
$object = "<yourObjectName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $content = $ossClient->getObject($bucket, $object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print($content);
print(__FUNCTION__ . ": OK" . "\n");
```

 **Note** If a power outage occurs, data stored in the memory is lost. For more information about how to store the downloaded object in a local disk, see [Download to local files](#).

1.4.3.7.4. Range download

If you need only part of the data in an object, you can use range download to download data within the specified range.

The following code provides an example on how to download data within the range from byte 0 to byte 4 in an object to local memory:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
// Query the data within the range from byte 0 to byte 4, including a total of 5 bytes of data. If the specified range is invalid, for example, the s
pecified range includes a negative number, or the specified value is larger than the object size, the entire object is downloaded.
$options = array(OssClient::OSS_RANGE => '0-4');
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $content = $ossClient->getObject($bucket, $object, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print ($content);
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.7.5. Conditional download

OSS allows you to configure one or more conditions for object download. If the specified conditions are met, the object is downloaded. If the specified conditions are not met, an error is returned and the object is not downloaded.

The following table describes the conditions that you can configure for object download.

Parameter	Description	Configuration method
If-Modified-Since	If the specified time is earlier than the time the object is modified, the object is downloaded. Otherwise, 304 Not modified is returned.	OssClient::OSS_IF_MODIFIED_SINCE
If-Unmodified-Since	If the specified time is later than or equal to the time the object is modified, the object is downloaded. Otherwise, 412 Precondition failed is returned.	OssClient::OSS_IF_UNMODIFIED_SINCE
If-Match	If the specified ETag matches that of an object, the object is downloaded. Otherwise, 412 Precondition failed is returned.	OssClient::OSS_IF_MATCH
If-None-Match	If the specified ETag does not match that of an object, the object is downloaded. Otherwise, 304 Not modified is returned.	OssClient::OSS_IF_NONE_MATCH

Note

- You can use If-Modified-Since and If-Unmodified-Since as object download conditions at the same time. You can use If-Match and If-None-Match as object download conditions at the same time.
- You can use `$ossClient->getObjectMeta` to obtain the ETag.

You can use conditional download to download objects to the local file or local memory. The following code provides an example on how to perform conditional download:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
try{
    $options = array(
        OssClient::OSS_HEADERS => array(
            OssClient::OSS_IF_MODIFIED_SINCE => "Fri, 13 Nov 2015 14:47:53 GMT"),
    );
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $content = $ossClient->getObject($bucket, $object, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print ($content);
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.8. Manage objects

1.4.3.8.1. Overview

You can manage objects in a bucket by using a variety of operations including SetObjectAcl, GetObjectAcl, ListObjects, DeleteObject, and CopyObject.

By using the preceding operations, you can perform the following operations:

- [Determine whether objects exist](#)
- [Manage object ACLs](#)
- [Manage object metadata](#)
- [List objects](#)
- [Delete objects](#)
- [Copy an object](#)

1.4.3.8.2. Determine whether an object exists

This topic describes how to determine whether an object exists.

The following code provides an example on how to determine whether an object exists:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $exist = $ossClient->doesObjectExist($bucket, $object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
var_dump($exist);
```

1.4.3.8.3. Manage the ACL of an object

This topic describes how to manage the access control list (ACL) of an object.

The following table describes the ACLs that you can configure for an object.

ACL	Description	Value
Inherited from the bucket	The ACL of the object is the same as the ACL of the bucket in which the object is stored.	default
Private	Only the object owner and authorized users are granted the read and write permissions on the object.	private
Public read	Only the object owner and authorized users are granted the read and write permissions on the object. Other users are granted only the read permissions on the object. Exercise caution when you set the ACL of the object to this value.	public-read
Public read/write	All users are granted the read and write permissions on the object. Exercise caution when you set the ACL of the object to this value.	public-read-write

Configure the ACL of an object

The following code provides an example on how to configure the ACL of an object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
// Set the ACL of the object to public read. If you do not specify the ACL of the object, the ACL of the object inherits the ACL of the bucket in which the object is stored.
$acl = "public-read";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putObjectAcl($bucket, $object, $acl);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query the ACL of an object

The following code provides an example on how to query the ACL of an object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $objectAcl = $ossClient->getObjectAcl($bucket, $object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
var_dump($objectAcl);
```

1.4.3.8.4. Manage object metadata

This topic describes how to manage object metadata.

Configure object metadata

The following code provides an example on how to configure object metadata:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}

if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$content = file_get_contents(__FILE__);
$options = array(
    OssClient::OSS_HEADERS => array(
        'Expires' => '2012-10-01 08:00:00',
        'Content-Disposition' => 'attachment; filename="xxxxxx"',
        'x-oss-meta-self-define-title' => 'user define meta info',
    )
);
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->putObject($bucket, $object, $content, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Modify object metadata

The following code provides an example on how to modify object metadata:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}

if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$fromBucket = "<yourFromBucketName>";
$fromObject = "<yourFromObjectName>";
$toBucket = "<yourToBucketName>";
$toObject = "<yourToObjectName>";
$options = array(
    OssClient::OSS_HEADERS => array(
        'Expires' => '2018-10-01 08:00:00',
        'Content-Disposition' => 'attachment; filename="xxxxxx"',
        'x-oss-meta-location' => 'location',
    )
);
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->copyObject($fromBucket, $fromObject, $toBucket, $toObject, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query object metadata

```

<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $objectMeta = $ossClient->getObjectMeta($bucket, $object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
if (isset($objectMeta[strtolower('Content-Disposition')]) &&
    'attachment; filename="xxxxxx"' === $objectMeta[strtolower('Content-Disposition')])
) {
    print(__FUNCTION__ . ": ObjectMeta checked OK" . "\n");
} else {
    print(__FUNCTION__ . ": ObjectMeta checked FAILED" . "\n");
}

```

1.4.3.8.5. List objects

This topic describes how to list all objects or objects that meet specific conditions in a specified bucket.

List all objects

The following code provides an example on how to list all the objects in a specified bucket:

```

<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
$nextMarker = '';
while (true) {
    try {
        $options = array(
            'delimiter' => '',
            'marker' => $nextMarker,
        );
        $listObjectInfo = $ossClient->listObjects($bucket, $options);
    } catch (OssException $e) {
        printf(__FUNCTION__ . ": FAILED\n");
        printf($e->getMessage() . "\n");
        return;
    }
    // Obtain nextMarker to list the remaining objects. The reading starts from the object next to the last object previously obtained through the li
    stObjects operation.
    $nextMarker = $listObjectInfo->getNextMarker();
    $listObject = $listObjectInfo->getObjectList();
    $listPrefix = $listObjectInfo->getPrefixList();
    if (! empty($listObject)) {
        print("objectList:\n");
        foreach ($listObject as $objectInfo) {
            print($objectInfo->getKey() . "\n");
        }
    }
    if (! empty($listPrefix)) {
        print("prefixList: \n");
        foreach ($listPrefix as $prefixInfo) {
            print($prefixInfo->getPrefix() . "\n");
        }
    }
    if ($listObjectInfo->getIsTruncated() !== "true") {
        break;
    }
}
}

```

Note

- In the preceding sample code, \$options is optional and can be used to list all objects in the entire bucket or in a folder.
- The listObjects method is used to obtain a list of objects. For information about how to obtain attributes such as the name and size of an object, see [Manage object metadata](#).

List the objects that meet specific conditions

The following code provides an example on how to list objects that meet specific conditions:


```

<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket= "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
$prefix = 'dir/';
$delimiter = '/';
$nextMarker = '';
$maxkeys = 10;
$options = array(
    'delimiter' => $delimiter,
    'prefix' => $prefix,
    'max-keys' => $maxkeys,
    'marker' => $nextMarker,
);
try {
    $listObjectInfo = $ossClient->listObjects($bucket, $options);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
$objectList = $listObjectInfo->getObjectList(); // object list
$prefixList = $listObjectInfo->getPrefixList(); // directory list
if (! empty($objectList)) {
    print("objectList:\n");
    foreach ($objectList as $objectInfo) {
        print($objectInfo->getKey() . "\n");
    }
}
if (! empty($prefixList)) {
    print("prefixList: \n");
    foreach ($prefixList as $prefixInfo) {
        print($prefixInfo->getPrefix() . "\n");
    }
}
}

```

The following table describes the parameters contained in \$options from the preceding sample code.

Parameter	Description	Required
delimiter	Specifies the character used to group objects by name. CommonPrefixes specifies a set of substrings of objects whose names share a specific prefix and end with the next occurrence of the specified delimiter.	No
prefix	Specifies the prefix that the returned object names must contain.	No
max-keys	Specifies the maximum number of objects that can be listed each time. The default value is 100. The maximum value is 1000.	No
marker	Specifies the name of the object after which the list begins.	No

1.4.3.8.6. Delete objects

This topic describes how to delete objects.

 **Warning** Deleted objects cannot be recovered. Exercise caution when you delete objects.

Delete a single object

The following code provides an example on how to delete a single object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteObject($bucket, $object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Delete multiple objects

The following code provides an example on how to delete multiple objects:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$objects = array();
$objects[] = "<yourObjectName1>";
$objects[] = "<yourObjectName2>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteObjects($bucket, $objects);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

 **Note** If an exception is thrown when you use `try{}catch{} to catch errors, no objects are deleted. In this case, you can use $e->getMessage to obtain the error message and analyze the causes of the error.`

1.4.3.8.7. Copy objects

This topic describes how to copy an object.

Take note of the following items when you copy an object:

- You must have read permissions on the source object and write permissions on the destination bucket.
- The source bucket and the destination bucket must be in the same region.

Copy a small object

You can call `$ossClient->copyObject` to copy an object less than or equal to 1 GB in size from the source bucket to the destination bucket within the same region.

The following code provides an example on how to copy a small object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$from_bucket = "<yourFromBucketName>";
$from_object = "<yourFromObjectName>";
$to_bucket = $bucket;
$to_object = $from_object . '.copy';
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->copyObject($from_bucket, $from_object, $to_bucket, $to_object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Copy a large object

To copy an object whose size is greater than 1 GB, you must split the object into parts and sequentially copy the parts by using UploadPartCopy. To implement multipart copy, perform the following steps:

1. Call `$ossClient->initiateMultipartUpload` to initiate a multipart copy task.
2. Call `$ossClient->uploadPartCopy` to perform multipart copy. All parts must be larger than 100 KB, except for the very last part.
3. Call `$ossClient->completeMultipartUpload` to complete the multipart copy task.

The following code provides an example on how to use multipart copy to copy large objects:

```

<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$src_bucket = "<yourSourceBucketName>";
$src_object = "<yourSourceObjectName>";
$dst_bucket = "<yourDestinationBucketName>";
$dst_object = "<yourDestinationObjectName>";
// Set the part size.
$part_size = 256*1024*1024;
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $objectMeta = $ossClient->getObjectMeta($src_bucket, $src_object);
    $length = $objectMeta['content-length'] + 0;
    // Initiate a multipart copy task.
    $upload_id = $ossClient->initiateMultipartUpload($dst_bucket, $dst_object);
    // Copy each part sequentially.
    $pieces = $ossClient->generateMultiuploadParts($length, $part_size);
    $response_upload_part = array();
    $copyId = 1;
    $upload_position = 0;
    foreach ($pieces as $i => $piece) {
        $from_pos = $upload_position + (integer)$piece['seekTo'];
        $to_pos = (integer)$piece['length'] + $from_pos - 1;
        $up_options = array(
            'start' => $from_pos,
            'end' => $to_pos,
        );
        $response_upload_part[] = $ossClient->uploadPartCopy($src_bucket, $src_object, $dst_bucket, $dst_object, $copyId, $upload_id, $up_options);
        $copyId = $copyId + 1;
    }
    // Complete the multipart copy task.
    $upload_parts = array();
    foreach ($response_upload_part as $i => $etag) {
        $upload_parts[] = array(
            'PartNumber' => ($i + 1),
            'ETag' => $etag,
        );
    }
    $result = $ossClient->completeMultipartUpload($dst_bucket, $dst_object, $upload_id, $upload_parts);
} catch (OssException $e) {
    printf(_FUNCTION_ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(_FUNCTION_ . ": OK" . "\n");

```

1.4.3.9. Versioning

1.4.3.9.1. Manage versioning

The versioning state of a bucket takes effect on all objects in the bucket. If you enable versioning for a bucket, you can recover any previous version of an object in the bucket when you accidentally overwrite or delete the object.

A bucket can be in one of the following versioning states: unversioned (default), versioning-enabled, or versioning-suspended.



Notice If versioning is enabled for a bucket, the versioning state of the bucket cannot be set back to unversioned. However, you can set the versioning state of the bucket to suspended.

Configure versioning for a bucket

The following code provides an example on how to configure the versioning state of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    // Set the versioning state of the bucket to enabled or suspended.
    $ossClient->putBucketVersioning($bucket, "Enabled");
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query the versioning state of a bucket

The following code provides an example on how to query the versioning state of a bucket:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    // Query the versioning state of the bucket.
    $status = $ossClient->getBucketVersioning($bucket);
    print($status);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

List the versions of all objects in a bucket

The following code provides an example on how to list the versions of all objects including delete markers in a specified bucket:

```

<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    $maxKey = 100;
    $nextKeyMarker = '';
    $nextVersionIdMarker = '';
    while (true) {
        $options = array(
            'delimiter' => '',
            'key-marker' => $nextKeyMarker,
            'max-keys' => $maxKey,
            'version-id-marker' => $nextVersionIdMarker,
        );
        $result = $ossClient->listObjectVersions($bucket, $options);
        $nextKeyMarker = $result->getNextKeyMarker();
        $nextVersionIdMarker = $result->getNextVersionIdMarker();
        $objectList = $result->getObjectVersionList();
        $deleteMarkerList = $result->getDeleteMarkerList();
        // Display the versions of all objects in the bucket.
        if (!empty($objectList)) {
            print("objectList:\n");
            foreach ($objectList as $objectInfo) {
                print($objectInfo->getKey() . ",");
                print($objectInfo->getVersionId() . ",");
                print($objectInfo->getLastModified() . ",");
                print($objectInfo->getETag() . ",");
                print($objectInfo->getSize() . ",");
                print($objectInfo->getIsLatest() . "\n");
            }
        }
        // Display the version information about all delete markers in the bucket.
        if (!empty($deleteMarkerList)) {
            print("deleteMarkerList: \n");
            foreach ($deleteMarkerList as $deleteMarkerInfo) {
                print($deleteMarkerInfo->getKey() . ",");
                print($deleteMarkerInfo->getVersionId() . ",");
                print($deleteMarkerInfo->getLastModified() . ",");
                print($deleteMarkerInfo->getIsLatest() . "\n");
            }
        }
        if ($result->getIsTruncated() !== "true") {
            break;
        }
    }
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");

```

1.4.3.9.2. Upload objects

This topic describes how to upload objects to a versioned bucket.

Simple upload

When an object is uploaded to a bucket for which versioning is enabled, OSS generates a random string as the globally unique version ID of the object and includes the version ID in the response as the `x-oss-version-id` header.

When an object is uploaded to a bucket for which versioning is suspended, OSS generates a version ID of null for the uploaded object. In this case, if you upload an object that has the same name as the existing object to the bucket, the existing object is overwritten.

The following code provides an example on how to upload an object to a versioned bucket by using simple upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Specify the full path of the object. The full path cannot contain the bucket name. Example: example/test.txt.
$object = "<yourObjectName>";
$content = "hello world";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    // Upload the object to the versioned bucket.
    $ret = $ossClient->putObject($bucket, $object, $content);
    // Display the versions of the listed objects.
    print("versionId:" . $ret[OssClient::OSS_HEADER_VERSION_ID]);
} catch (OssException $e) {
    printf(__FUNCTION__ . " : FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . " : OK" . "\n");
```

Append upload

When you upload an object to a versioned bucket by using append upload, take note of the following items:

- You can perform the AppendObject operation only on an object whose current version is an appendable object.
- The AppendObject operation cannot be performed on an object whose current version is not an appendable object, such as a normal object or a delete marker.
- When you perform the AppendObject operation on an object whose current version is an appendable object, OSS does not generate a previous version for the appendable object.
- When you perform the PutObject or DeleteObject operation on an object whose current version is an appendable object, OSS stores the appendable object as a previous version and prevents the object from being further appended.

The following code provides an example on how to perform append upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
// Specify that the object content obtained after the first, second, and third append upload is Hello OSS, Hi OSS, and OSS OK.
$content_array = array('Hello OSS', 'Hi OSS', 'OSS OK');
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // Perform the first append upload. The position for the first append upload is 0, and the position for the next append upload is included in the response. The position from which the next append upload starts is the current length of the object.
    $position = $ossClient->appendObject($bucket, $object, $content_array[0], 0);
    $position = $ossClient->appendObject($bucket, $object, $content_array[1], $position);
    $position = $ossClient->appendObject($bucket, $object, $content_array[2], $position);
} catch (OssException $e) {
    printf(__FUNCTION__ . " : FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . " : OK" . "\n");
```

Multipart upload

If you call the CompleteMultipartUpload operation to complete the multipart upload task for an object in a versioned bucket, OSS generates a unique version ID for the uploaded object and returns the version ID in the response as the `x-oss-version-id` header.

The following code provides an example on how to perform multipart upload:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Core\OssUtil;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
// Specify the full path of the local file.
$uploadFile = "<yourLocalFile>";
/**
 * Step 1. Initiate a multipart upload task and obtain the upload ID.
 */
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // Obtain the upload ID. The upload ID uniquely identifies the multipart upload task. You can use the upload ID to cancel or query the multipart upload task.
    $uploadId = $ossClient->initiateMultipartUpload($bucket, $object);
} catch(OssException $e) {
    printf(_FUNCTION_ . ": initiateMultipartUpload FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(_FUNCTION_ . ": initiateMultipartUpload OK" . "\n");
/*
 * Step 2: Upload parts.
 */
$partSize = 10 * 1024 * 1024;
$uploadFileSize = sprintf('%u',filesize($uploadFile));
$pieces = $ossClient->generateMultiuploadParts($uploadFileSize, $partSize);
$responseUploadPart = array();
$uploadPosition = 0;
$isCheckMd5 = true;
foreach ($pieces as $i => $piece) {
    $fromPos = $uploadPosition + (integer)$piece[$ossClient::OSS_SEEK_TO];
    $toPos = (integer)$piece[$ossClient::OSS_LENGTH] + $fromPos - 1;
    $uploadOptions = array(
        // Upload the local file.
        $ossClient::OSS_FILE_UPLOAD => $uploadFile,
        // Specify part numbers.
        $ossClient::OSS_PART_NUM => ($i + 1),
        // Specify the position from which the multipart upload task starts.
        $ossClient::OSS_SEEK_TO => $fromPos,
        // Specify the object size.
        $ossClient::OSS_LENGTH => $toPos - $fromPos + 1,
        // Specify whether to enable MD5 verification. A value of true indicates MD5 verification is enabled. A value of false indicates MD5 verification is disabled.
        $ossClient::OSS_CHECK_MD5 => $isCheckMd5,
    );
    // Enable MD5 verification.
    if ($isCheckMd5) {
        $contentMd5 = OssUtil::getMd5SumForFile($uploadFile, $fromPos, $toPos);
        $uploadOptions[$ossClient::OSS_CONTENT_MD5] = $contentMd5;
    }
    try {
        // Upload parts.
        $responseUploadPart[] = $ossClient->uploadPart($bucket, $object, $uploadId, $uploadOptions);
    } catch(OssException $e) {
        printf(_FUNCTION_ . ": initiateMultipartUpload, uploadPart - part#{ $i } FAILED\n");
        printf($e->getMessage() . "\n");
        return;
    }
    printf(_FUNCTION_ . ": initiateMultipartUpload, uploadPart - part#{ $i } OK\n");
}
// $uploadParts is an array that consists of the ETag and PartNumber of each part.
$uploadParts = array();
foreach ($responseUploadPart as $i => $eTag) {
    $uploadParts[] = array(
        'PartNumber' => ($i + 1),
        'ETag' => $eTag,
    );
}
/**
 * Step 3: Complete the multipart upload task.
 */
}
```



```

*/
try {
    // All valid $uploadParts are required for the CompleteMultipartUpload operation. After OSS receives the $uploadParts, OSS verifies all parts one
    by one. After all parts are verified, OSS combines these parts into a complete object.
    $ret = $ossClient->completeMultipartUpload($bucket, $object, $uploadId, $uploadParts);
    // Display the version ID of the uploaded object.
    print("versionId:" . $ret[OssClient::OSS_HEADER_VERSION_ID]);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": completeMultipartUpload FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
printf(__FUNCTION__ . ": completeMultipartUpload OK\n");

```

1.4.3.9.3. Download objects

By default, when you call the `GetObject` operation to download an object from a versioned bucket, only the current version of the object is returned.

When you perform the `GetObject` operation on an object in a bucket, you may obtain one of the following results:

- If the current version of the object is a delete marker, OSS returns 404 Not Found.
- If the version ID of the object is specified in the request, OSS returns the specified version of the object. If the version ID is set to null in the request, OSS returns the version whose version ID is null.
- If the version ID is set to a delete marker, OSS returns 405 Method Not Allowed.

The following code provides an example on how to download an object:

```

<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;

// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS i
s a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM
console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Specify the full path of the object. The full path cannot contain the bucket name. Example: example/test.txt.
$object = "<yourObjectName>";
// Specify the version ID of the version you want to download.
$versionId = "<yourObjectVersionId>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);

try{
    // Download the version with the specified version ID.
    $content = $ossClient->getObject($bucket, $object, array(OssClient::OSS_VERSION_ID => $versionId));
    print($content);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");

```

1.4.3.9.4. Copy objects

This topic describes how to copy objects in a versioned bucket. You can call `CopyObject` to copy an object that is up to 1 GB in size and call `UploadPartCopy` to copy an object that is larger than 1 GB.

Copy a small object

You can call `CopyObject` to copy an object that is up to 1 GB in size from a source bucket to a destination bucket in the same region.

- By default, the `x-oss-copy-source` header in a `CopyObject` request specifies the current version of the object to copy. If the current version of the object is a delete marker, OSS returns the 404 status code, which indicates that the object does not exist. You can add a version ID in the `x-oss-copy-source` header to copy a specified object version. Delete markers cannot be copied.
- You can copy a previous version of an object to the same bucket. The copied previous version becomes the new version of the object. This way, the previous version of the object is recovered.
- If versioning is enabled for the destination bucket, OSS generates a unique version ID for the new object. The version ID is returned as the `x-oss-version-id` header in the response. If versioning is disabled or suspended for the destination bucket, OSS generates a version whose version ID is null for the destination object and overwrites the original version whose version ID is null.
- Appendable objects in a bucket for which versioning is enabled or suspended cannot be copied.

The following code provides an example on how to copy a small object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Specify the full path of the object. The full path cannot contain the bucket name. Example: example/test.txt.
$object = "<yourObjectName>";
$versionId = "<yourObjectVersionId>";
$to_bucket = $bucket;
$to_object = $object . '.copy';
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try{
    // Copy the object with the specified version ID.
    $ret = $ossClient->copyObject($bucket, $object, $to_bucket, $to_object, array(OssClient::OSS_VERSION_ID => $versionId));
    print("versionId:" . $ret['x-oss-version-id']);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Copy a large object

To copy an object that is larger than 1 GB in size, you must split the object into parts and copy the parts by calling UploadPartCopy.

By default, the UploadPartCopy operation uploads a part by copying data from the current version of an existing object. You can add a version ID to the `x-oss-copy-source` request header as the condition to upload a part by copying data from the specified version of an existing object. Example: `x-oss-copy-source : /SourceBucketName/SourceObjectName?versionId=111111`.

Note The value of SourceObjectName must be URL-encoded. The version ID of the copied object is returned as the `x-oss-copy-source-version-id` header value in the response.

If `versionId` is not specified in the request and the current version of the source object is a delete marker, OSS returns 404 Not Found. If `versionId` is specified in the request and the specified version of the source object is a delete marker, OSS returns 400 Bad Request.

The following code provides an example on how to use multipart copy to copy a large object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$versionId = "<yourObjectVersionId>";
$to_bucket = $bucket;
$to_object = $object . '.copy';
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
/**
 * Step 1. Initiate a multipart upload task and obtain the upload ID.
 */
try {
    // Obtain the upload ID. The upload ID is the unique identifier for a multipart upload task. You can perform related operations such as cancel or query the multipart upload task based on the upload ID.
    $uploadId = $ossClient->initiateMultipartUpload($to_bucket, $to_object);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": initiateMultipartUpload FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": initiateMultipartUpload OK" . "\n");
```

```

/*
 * Step 2: Upload parts.
 */
$partSize = 10 * 1024 * 1024;
try {
    $meta = $ossClient->getSimplifiedObjectMeta($bucket, $object);
    $uploadSize = $meta[strtolower('Content-Length')];
} catch (OssException $e) {
    printf(__FUNCTION__ . ": getSimplifiedObjectMeta FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}

$pieces = $ossClient->generateMultiuploadParts($uploadSize, $partSize);
$responseUploadPart = array();
$uploadPosition = 0;
foreach ($pieces as $i => $piece) {
    $fromPos = $uploadPosition + (integer)$piece[$ossClient::OSS_SEEK_TO];
    $toPos = (integer)$piece[$ossClient::OSS_LENGTH] + $fromPos - 1;
    $uploadOptions = array(
        // Specify the position of the source object from which the multipart copy task starts .
        'start' => $fromPos,
        // Specify the position of the source object at which the multipart copy task ends.
        'end' => $toPos,
        // Specify the version ID of the source object that you want to copy.
        OssClient::OSS_VERSION_ID => $versionId
    );
    try {
        // Upload parts.
        $responseUploadPart[] = $ossClient->uploadPartCopy($bucket, $object, $to_bucket, $to_object, ($i + 1), $uploadId, $uploadOptions);
    } catch (OssException $e) {
        printf(__FUNCTION__ . ": initiateMultipartUpload, uploadPartCopy - part#{$i} FAILED\n");
        printf($e->getMessage() . "\n");
        return;
    }
    printf(__FUNCTION__ . ": initiateMultipartUpload, uploadPartCopy - part#{$i} OK\n");
}
// $uploadParts is an array that consists of the ETag and PartNumber of each part.
$uploadParts = array();
foreach ($responseUploadPart as $i => $eTag) {
    $uploadParts[] = array(
        'PartNumber' => ($i + 1),
        'ETag' => $eTag,
    );
}
/**
 * Step 3: Complete multipart upload task.
 */
try {
    // All valid $uploadParts are required for the CompleteMultipartUpload operation. After OSS receives the $uploadParts, OSS verifies all parts one
    by one. After all parts are verified, OSS combines these parts into a complete object.
    $ret = $ossClient->completeMultipartUpload($to_bucket, $to_object, $uploadId, $uploadParts);
    // Display the versions of the listed objects.
    print("versionId:" . $ret[OssClient::OSS_HEADER_VERSION_ID]);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": completeMultipartUpload FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
printf(__FUNCTION__ . ": completeMultipartUpload OK\n");

```

1.4.3.9.5. Delete objects

This topic describes how to delete a single object, multiple objects, or objects whose names contain a specific prefix from a versioned bucket.

Delete operations on a versioned bucket

- Delete an object without specifying a version ID (temporary deletion)

By default, if you do not specify the version ID of the object that you want to delete in the request, OSS does not delete the current version of the object but adds a delete marker to the object as the new version. If you perform the `GetObject` operation on the object, OSS identifies the current version of the object as a delete marker and returns `404 Not Found`. In addition, the `x-oss-delete-marker` header and the `x-oss-version-id` header that indicates the version ID of the created delete marker are included in the response.

- Delete an object by specifying a version ID (permanent deletion)

If you specify the version ID of the object that you want to delete in the request, OSS permanently deletes the version specified by the `versionId` field in the `params` parameter. To delete a version whose ID is null, add `params['versionId'] = "null"` in `params` in the request. OSS identifies the string "null" as the ID of the version to delete and deletes the version whose ID is null.

Delete a single object

The following examples describe how to permanently or temporarily delete a single object from a versioned bucket.

- Permanently delete an object from a versioned bucket

The following code provides an example on how to permanently delete an object from a versioned bucket by specifying the version ID of the object in the request:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS
is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
// Specify the full path of the object. The full path cannot contain the bucket name. Example: example/test.txt.
$object = "<yourObjectName>";
$versionId = "<yourObjectVersionId>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try{
    // Delete the object with the specified version ID.
    $ossClient->deleteObject($bucket, $object, array(OssClient::OSS_VERSION_ID => $versionId));
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

- Temporarily delete an object from a versioned bucket

The following code provides an example on how to temporarily delete an object from a versioned bucket by sending a request in which no version ID is specified:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS
is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try{
    // Temporarily delete an object without specifying its version ID. A delete marker is added to the object.
    $ret = $ossClient->deleteObject($bucket, $object);
    print("delete marker:" . $ret['x-oss-delete-marker'] . "\n");
    print("version id:" . $ret['x-oss-version-id']);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Delete multiple objects

The following examples demonstrate how to permanently or temporarily delete multiple objects from a versioned bucket.

- Permanently delete multiple objects from a versioned bucket

The following code provides an example on how to permanently delete multiple objects with specified version IDs from a versioned bucket:

```

<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\DeleteObjectInfo;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS
is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the R
AM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try{
    // Delete the objects with the specified version IDs.
    $objects = array();
    $objects[] = new DeleteObjectInfo("<yourObject1Name>", "<object1VersionId>");
    $objects[] = new DeleteObjectInfo("<yourObject2Name>", "<object2VersionId>");
    $deletedObjectList = $ossClient->deleteObjectVersions($bucket, $objects);
    // Display the deleted objects.
    if (!empty($deletedObjectList)) {
        print("deletedObjectList:\n");
        foreach ($deletedObjectList as $deletedObjectInfo) {
            print($deletedObjectInfo->getKey() . ",");
            print($deletedObjectInfo->getVersionId() . ",");
            print($deletedObjectInfo->getDeleteMarker() . ",");
            print($deletedObjectInfo->getDeleteMarkerVersionId() . "\n");
        }
    }
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");

```

- Temporarily delete multiple objects from a versioned bucket

The following code provides an example on how to delete multiple objects from a versioned bucket by sending a request in which no version IDs are specified:

```

<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Model\DeleteObjectInfo;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS
is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the R
AM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try{
    // Temporarily delete multiple object without specifying their version IDs. Delete markers are added to the objects.
    $objects = array();
    $objects[] = new DeleteObjectInfo("<yourObject1Name>");
    $objects[] = new DeleteObjectInfo("<yourObject2Name>");
    $deletedObjectList = $ossClient->deleteObjectVersions($bucket, $objects);
    // Display the deleted objects.
    if (!empty($deletedObjectList)) {
        print("deletedObjectList:\n");
        foreach ($deletedObjectList as $deletedObjectInfo) {
            print($deletedObjectInfo->getKey() . ",");
            print($deletedObjectInfo->getVersionId() . ",");
            print($deletedObjectInfo->getDeleteMarker() . ",");
            print($deletedObjectInfo->getDeleteMarkerVersionId() . "\n");
        }
    }
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");

```

1.4.3.9.6. Query object metadata

By default, when you perform the `HeadObject` operation on an object in a versioned bucket, only the metadata of the current version of the object is returned.

Note If the current version of the object is a delete marker, OSS returns 404 Not Found. If you specify the version ID in the request, OSS returns the metadata of the specified version of the object.

The following code provides an example on how to query object metadata:

```

<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS i
s a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM
console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$versionId = "<yourObjectVersionId>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    // Query all metadata of the object with the specified version ID.
    $objectMeta = $ossClient->getObjectMeta($bucket, $object, array(OssClient::OSS_VERSION_ID => $versionId));
    print_r($objectMeta);
    // Query part of metadata of the object with the specified version ID.
    $objectMeta = $ossClient->getSimplifiedObjectMeta($bucket, $object, array(OssClient::OSS_VERSION_ID => $versionId));
    print_r($objectMeta);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");

```

1.4.3.9.7. Manage the ACL of an object

OSS provides the following four ACLs for objects to control access to objects: public read/write, public read, private read/write, and default permission. This topic describes how to manage the ACLs of objects in a versioned bucket.

Object ACLs

The following table describes the four ACLs of objects.

ACL	Name	Access control
public-read-write	Public read/write	All users, including anonymous users, can read and write the object.
public-read	Public read	Only the owner or authorized users of the object can write the object. Other users, including anonymous users, can only read the object.
private	Private read/write	Only the owner or authorized users of the object can read and write the object.
default	Default permission	The ACL of an object is the same as that of the bucket in which the object is stored.

The ACL of an object takes precedence over the ACL of the bucket in which the object is stored. For example, if the ACL of a bucket is public read and the ACL of an object in the bucket is private, only the owner or authorized users of the object can read and write the object.

Configure the ACL for an object.

By default, the `putObjectAcl` method is used to configure the ACL of the current version of an object. If the current version of the object is a delete marker, OSS returns 404 Not Found. You can specify a version ID in the request to configure the ACL of the specified version of an object.

The following code provides an example on how to configure the ACL of an object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}

if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// Specify the name of the object. The name is the full path of the object and cannot contain the bucket name. Example: example/test.txt.
$object = "<yourObjectName>";
// Specify the ID of the version for which you want to configure ACL.
$versionId = "<yourObjectVersionId>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    // Set the ACL of the specified version of the object to public read.
    $ossClient->putObjectAcl($bucket, $object, 'public-read', array(OssClient::OSS_VERSION_ID => $versionId));
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query object ACLs

By default, the `getObjectAcl` method is used to query the ACL of the current version of an object. If the current version of the object is a delete marker, OSS returns 404 Not Found. You can specify a version ID in the request to query the ACL of the specified version of an object.

The following code provides an example on how to query the ACL of an object:

```
<?php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
// The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
$endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// Specify the name of the bucket.
$bucket = "<yourBucketName>";
// Specify the name of the object. The name is the full path of the object and cannot contain the bucket name. Example: example/test.txt.
$object = "<yourObjectName>";
// Specify the ID of the version for which you want to query ACL.
$versionId = "<yourObjectVersionId>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
try {
    // Query the ACL of the object.
    $acl = $ossClient->getObjectAcl($bucket, $object, array(OssClient::OSS_VERSION_ID => $versionId));
    printf($acl . "\n");
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

1.4.3.10. Authorized access

This topic describes how to use Security Token Service (STS) and a signed URL to authorize access.

Use STS to authorize temporary access

Alibaba Cloud STS is a web service that provides a temporary access token to a cloud computing user. You can use STS to grant a third-party application or your RAM user an access credential with a customized validity period and permissions.

STS has the following benefits:

- You need only to generate an access token and send the access token to a third-party application, instead of exposing your long-term AccessKey pair to the third-party application. You can customize the access permissions and validity period of this token.
- The access token automatically expires when the validity period ends.

The following code provides an example on how to create a signature request by using STS:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$securityToken = "<yourSecurityToken>";
$content = "Hi, OSS.";
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false, $securityToken);
    $ossClient->putObject($bucket, $object, $content);
} catch (OssException $e) {
    print $e->getMessage();
}
```

Use a signed URL to authorize temporary access

You can generate a signed URL and provide the URL to a visitor for temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of time during which the visitor can access OSS.

For the complete code of authorized access, visit [GitHub](#).

- Generate a signed URL to download an object

The following code provides an example on how to generate a signed URL to download an object:


```

<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Http\RequestCore;
use OSS\Http\ResponseCore;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$securityToken = "<yourSecurityToken>";
// Set the validity period of a URL to 3,600 seconds.
$timeout = 3600;
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false, $securityToken);
    // Generate the signed URL for GetObject.
    $signedUrl = $ossClient->signUrl($bucket, $object, $timeout);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": signedUrl: " . $signedUrl . "\n");
// You can run code to access the signed URL, or access the signed URL in a browser.
$request = new RequestCore($signedUrl);
// Set GET as the default method to access the signed URL.
$request->set_method('GET');
$request->add_header('Content-Type', '');
$request->send_request();
$res = new ResponseCore($request->get_response_header(), $request->get_response_body(), $request->get_response_code());
if ($res->isOk()) {
    print(__FUNCTION__ . ": OK" . "\n");
} else {
    print(__FUNCTION__ . ": FAILED" . "\n");
};

```

- Generate a signed URL to upload an object

The following code provides an example on how to generate a signed URL to upload an object:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
use OSS\Core\OssException;
use OSS\Http\RequestCore;
use OSS\Http\ResponseCore;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$securityToken = "<yourSecurityToken>";
$timeout = 3600;
try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false, $securityToken);
    // Generate the signed URL for PutObject.
    $signedUrl = $ossClient->signUrl($bucket, $object, $timeout, "PUT");
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": signedUrl: " . $signedUrl . "\n");
$content = "Hello OSS.";
$request = new RequestCore($signedUrl);
// Set PUT as the default method to access the signed URL.
$request->set_method('PUT');
$request->add_header('Content-Type', '');
$request->add_header('Content-Length', strlen($content));
$request->set_body($content);
$request->send_request();
$res = new ResponseCore($request->get_response_header(),
    $request->get_response_body(), $request->get_response_code());
if ($res->isOk()) {
    print(__FUNCTION__ . ": OK" . "\n");
} else {
    print(__FUNCTION__ . ": FAILED" . "\n");
};
```

1.4.3.11. IMG

OSS Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service that can process large amounts of data. After you upload source images to OSS, you can call RESTful API operations to process the images anytime, anywhere, and on any Internet device.

For the complete IMG code, visit [GitHub](#).

Basic features

OSS IMG provides the following features:

- Query image information.
- Convert image formats.
- Resize, crop, and rotate images.
- Apply image effect.
- Add image, text, and text-and-image watermarks to images.
- Customize image processing styles. Log on to the OSS console. Click a bucket name to go to the bucket details page. Click the **Image Processing (IMG)** tab and click **Create Rule** to customize an image processing style.
- Use cascade processing to call multiple IMG features.

Usage

You can use standard `HTTP GET` requests to access IMG. All processing parameters are encoded in the query string of a URL.

If the ACL of an image object is public-read, as described in the following table, anonymous users can access the image object.

Bucket ACL	Object ACL
public-read or public-read-write	default
Any ACL	public-read or public-read-write

You can anonymously access IMG by using a third-level domain in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=image/action,param_value
```

Parameter	Description
bucket	The name of the bucket.
endpoint	The domain name used to access the region.
object	The name of the image object.
image	The identifier reserved by IMG.
action	The operations performed on images, such as scaling, cropping, and rotating.
param	The parameters corresponding to the operations performed on the images.

- Basic operations

For example, you can scale an image to a width of 100 and adjust the height based on the ratio:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customize an image style

You can anonymously access IMG by using a third-level domain in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- style: the identifier reserved by IMG for the custom style.
- yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when you create the custom style in the console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade processing

You can use cascade processing to perform multiple operations in sequence on an image by using a URL in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Access by using HTTPS

IMG supports access by using HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Authorized access

If the ACL of an image object is private, you must have permissions on the image object before you can perform operations on the image object.

Bucket ACL	Object ACL
private	default
Any ACL	private

The following code provides an example on how to generate a signed URL to access IMG:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
// Generate a signed URL that you can directly access by using a browser. The validity period of the URL is 3,600 seconds.
$timeout = 3600;
$options = array(
    OssClient::OSS_PROCESS => "image/resize,m_lfit,h_100,w_100" );
$signedUrl = $ossClient->signUrl($bucket, $object, $timeout, "GET", $options);
print("tmp url: \n" . $signedUrl);
```

- Access by using SDKs

You can use OSS SDKs to access and process the image objects of any ACL.

- Basic operations

The following code provides an example on how to perform basic IMG operations including obtaining image information, converting image formats, resizing, cropping, rotating, watermarking, and applying effects:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$download_file = "download.jpg";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
// Upload a sample image.
$ossClient->uploadFile($bucket, $object, "example.jpg");
// Resize the image.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/resize,m_fixed,h_100,w_100" );
$ossClient->getObject($bucket, $object, $options);
// Crop the image.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/crop,w_100,h_100,x_100,y_100,r_1" );
$ossClient->getObject($bucket, $object, $options);
// Rotate the image.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/rotate,90" );
$ossClient->getObject($bucket, $object, $options);
// Sharpen the image.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/sharpen,100" );
$ossClient->getObject($bucket, $object, $options);
// Add watermarks to the image.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ" );
$ossClient->getObject($bucket, $object, $options);
// Convert the image format.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/format,png" );
$ossClient->getObject($bucket, $object, $options);
// Obtain the image information.
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => "image/info" );
$ossClient->getObject($bucket, $object, $options);
// Delete the sample image.
$ossClient->deleteObject($bucket, $object);
```

- Custom image styles

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
}

use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$download_file = "download.jpg";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
// Upload a sample image.
$ossClient->uploadFile($bucket, $object, "example.jpg");
// Customize the image style.
$style = "style/oss-pic-style-w-300";
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => $style);
$ossClient->getObject($bucket, $object, $options);
// Delete the sample image.
$ossClient->deleteObject($bucket, $object);
```

- Cascade processing

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
}

use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket = "<yourBucketName>";
$object = "<yourObjectName>";
$download_file = "download.jpg";
$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint, false);
// Upload a sample image.
$ossClient->uploadFile($bucket, $object, "example.jpg");
// Perform cascade processing.
$style = "image/resize,m_fixed,w_100,h_100/rotate,90";
$options = array(
    OssClient::OSS_FILE_DOWNLOAD => $download_file,
    OssClient::OSS_PROCESS => $style);
$ossClient->getObject($bucket, $object, $options);
// Delete the sample image.
$ossClient->deleteObject($bucket, $object);
```

Persistent storage of processed images

The following code provides an example on how to save processed images:

```
<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}
use OSS\OssClient;
$accessKeyId = "<yourAccessKeyId>";
$accessKeySecret = "<yourAccessKeySecret>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
$endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
$bucket= "<yourBucketName>";
// Specify the object name.
$object = "<yourObjectName>";
// Specify the object name for the processed image.
$targetObject = "<yourTargetObjectName>";
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    // The following code provides an example on how to convert the image format of a processed image. To crop, rotate, or watermark an image, modify the
    $options parameters.
    $options = array(
        OssClient::OSS_PROCESS => "image/format,png"
    );
    $content = $ossClient->getObject($bucket, $object, $options);
    $ossClient->putObject($bucket, $targetObject, $content);
} catch (OssException $e) {
    printf(__FUNCTION__ . " : FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print($content);
printf(__FUNCTION__ . " : OK" . "\n");
```

1.4.3.12. Error handling

OSS SDK for PHP exceptions (OssException) include errors caused by invalid parameters and objects that do not exist. You can use getMessage to obtain an error message.

For more information about OssException, visit [GitHub](#).

Error handling example

The following code provides an example on how to rectify an error and display the error information when you create a bucket that already exists:

```
try {
    $ossClient->createBucket($bucket);
} catch (OssException $e) {
    print ("Exception:" . $e->getMessage() . "\n");
}
```

The following table describes the error messages you can obtain.

Parameter	Description
HTTPStatus	The HTTP status code that can be obtained by using getHTTPStatus.
ErrorCode	The error code returned by OSS. The error code can be obtained by using getErrorCode.
ErrorMessage	The error information returned by OSS. The error information can be obtained by using getErrorMessage.
RequestId	The UUID used to uniquely identify the request. If the problem persists, you can provide the request ID of a problem to the OSS development engineers for help. You can obtain the UUID by using getRequestId.
Details	The error information details returned by OSS. You can obtain the error message details by using getDetails.

OSS error codes

Error code	Error message	HTTP status code
AccessDenied	The error message returned because the access is denied.	403
BucketAlreadyExists	The error message returned when the bucket already exists.	409
BucketNotEmpty	The error message returned when the bucket is not empty.	409
EntityTooLarge	The error message returned because the entity exceeds the maximum allowed size.	400

Error code	Error message	HTTP status code
EntityTooSmall	The error message returned because the entity is smaller than the minimum allowed size.	400
FileGroupTooLarge	The error message returned because the object group exceeds the maximum allowed size.	400
FilePartNotExist	The error message returned because the specified part does not exist.	400
FilePartStale	The error message returned because the part has expired.	400
InvalidArgument	The error message returned because the parameter format is invalid.	400
InvalidAccessKeyId	The error message returned because the specified AccessKey ID does not exist.	403
InvalidBucketName	The error message returned because the bucket name is invalid.	400
InvalidDigest	The error message returned because the digest is invalid.	400
InvalidObjectName	The error message returned because the object name is invalid.	400
InvalidPart	The error message returned because the part is invalid.	400
InvalidPartOrder	The error message returned because the part order is invalid.	400
InvalidTargetBucketForLogging	The error message returned because the target bucket specified for logging is invalid.	400
InternalError	The error message returned because an internal OSS error occurs.	500
MalformedXML	The error message returned because the XML format is invalid.	400
MethodNotAllowed	The error message returned because the method is not supported.	405
MissingArgument	The error message returned because a parameter is missing.	411
MissingContentLength	The error message returned because the content length is not specified.	411
NoSuchBucket	The error message returned because the specified bucket does not exist.	404
NoSuchKey	The error message returned because the specified object does not exist.	404
NoSuchUpload	The error message returned because the specified multipart upload ID does not exist.	404
NotImplemented	The error message returned because the method cannot be implemented.	501
PreconditionFailed	The error message returned because a precondition error occurs.	412
RequestTimeTooSkewed	The error message returned because the difference between the time on the client when the request is sent and the time on the server when the request is received is greater than 15 minutes.	403
RequestTimeout	The error message returned because the request times out.	400
SignatureDoesNotMatch	The error message returned because a signature error occurs.	403
InvalidEncryptionAlgorithmError	The error message returned because the specified encryption algorithm based on entropy encoding is invalid.	400

1.4.4. C-SDK

1.4.4.1. Preface

This topic describes the information about OSS SDK for C.

Compatibility

- For OSS SDK for C V3.*.*: compatible.
- For OSS SDK for C V2.*.*:
 - Compatible with Windows
 - Compatible with Linux API operations and incompatible with the following linked list (aos_list_t) traversal API operations:
 - aos_list_for_each_entry
 - aos_list_for_each_entry_reverse
 - aos_list_for_each_entry_safe
 - aos_list_for_each_entry_safe_reverse
- For OSS SDK for C V1.*.*: compatible except the following structures and API operations:
 - oss_request_options_t
 - oss_get_object_to_buffer
 - oss_get_object_to_file
 - oss_get_object_to_buffer_by_url
 - oss_get_object_to_file_by_url
 - oss_init_multipart_upload
 - oss_complete_multipart_upload
- For OSS SDK for C V0.0.*: incompatible.

SDK source code

For the source code of OSS SDK for C, visit [GitHub](#).

Sample codes

The OSS SDK for C provides a variety of sample codes for your reference or use. The following table describes the sample codes provided by OSS SDK for C.

Document	Description
oss_put_object_sample	Shows you how to upload objects, including upload objects from the memory, upload objects by using MD5 verification, upload objects with object metadata, and upload objects based on a signed URL.
oss_get_object_sample.c	Shows you how to download objects, including download objects to the memory or local files, range download, and object download based on a signed URL.
oss_append_object_sample.c	Shows you how to perform append upload, including append upload from the memory and append upload from local files.
oss_multipart_upload_sample.c	Shows you how to perform multipart upload, including multipart upload from the memory, multipart upload from local files, and multipart upload cancellation.
oss_resumable_sample.c	Shows you how to perform resumable upload.
oss_head_object_sample.c	Shows you how to query object metadata.
oss_delete_object_sample.c	Shows you how to delete objects, including the deletion of a single object and multiple objects.
oss_callback_sample.c	Shows you how to perform upload callback, including object upload callback and multipart upload callback.
oss_progress_sample.c	Shows you how to use a progress bar during object upload, append upload, multipart upload, and download.
oss_crc_sample.c	Shows you how to perform CRC in object upload and download.
oss_image_sample.c	Shows you how to use Image Processing (IMG).

1.4.4.2. Installation

This topic describes how to install OSS SDK for C.

Install OSS SDK for C on Linux

- Install CMake and third-party libraries

When you install OSS SDK for C on Linux, you must install CMake and the following third-party libraries: cURL, APR, APR-util, and Mini-XML.

The following table describes parameters required to install CMake and the third-party libraries.

Parameter	Description	Version
CMake	Manages the build process of software by using a compiler-independent method.	2.6.0 or later
curl	Resolves network issues.	7.32.0 or later
apr-util	Resolves the issues of memory and different platforms.	1.5.2 or later
minixml	Parses the XML data returned by a request.	2.9

Install CMake and the third-party libraries based on the following systems:

- Ubuntu and Debian

- Install CMake

```
sudo apt-get install cmake
```

- Install third-party libraries

```
sudo apt-get install libcurl4-openssl-dev libapr1-dev libaprutil1-dev libmxml-dev
```

- RedHat, Aliyun, and CentOS

- Install CMake

```
sudo yum install cmake
```

- Install third-party libraries

```
sudo yum install curl-devel apr-devel apr-util-devel mxml mxml-devel
```

If the YUM source does not contain the MXML installation package, use RPM to install MXML. Download the RPM package based on your operating system:

- 64-bit: [MXML-2.9-1.x86_64.rpm](#)
- 32-bit: [MXML-2.9-1.i386.rpm](#)

Use the RPM package to install MXML:

```
rpm -ivh mxml-2.9-1.x86_64.rpm
```

- SuSE

- Install CMake

```
zypper install cmake
```

- Install third-party libraries

```
zypper install libcurl-devel libapr1-devel libapr-util-devel mxml-devel
```

- Other Linux-based systems

- Install CMake

[Download CMake](#), and run the following commands to install CMake:

```
./configure
make
make install
```

 **Note** When you run `./configure`, the default installation path is `/usr/local/`. To specify the installation path, add the `-prefix` option when you run `./configure`.

- Install libcurl

[Download libcurl](#), and run the following commands to install libcurl:

```
./configure
make
make install
```

- Install APR

[Download APR](#), and run the following commands to install APR:

```
./configure
make
make install
```

- Install APR-util

[Download APR-util](#), and run the following commands to install APR-util:

```
// You must specify the --with-APR option when you install APR-util.
./configure --with-apr=/your/apr/install/path
make
make install
```

- Install Mini-XML

 **Note** Use Mini-XML 2.x for installation.

[Download Mini-XML](#), and run the following commands to install Mini-XML:

```
./configure
make
sudo make install
```

- Download the SDK

- [Download directly](#)
- [Download from GitHub](#)
- [Download previous versions](#)

- Install the SDK

- If third-party libraries cURL, APR, APR-util, and MXML are installed in the default path, run the following commands to install OSS SDK for C:

```
cmake .
make
make install
```

- You can specify a path to which third-party libraries cURL, APR, APR-util, and MXML are installed. Run the following commands to install OSS SDK for C:

```
cmake -f CMakeLists.txt
// Set the build type to Release. Typical build types include Debug, Release, RelWithDebInfo, and MinSizeRel. By default, Debug is used.
-DCMAKE_BUILD_TYPE=Release
// Set the directory to install the SDK.
-DCMAKE_INSTALL_PREFIX=/usr/local/
// Specify the path where third-party libraries cURL, APR, APR-util, and XML are installed.
-DCURL_INCLUDE_DIR=/usr/include/curl
-DCURL_LIBRARY=/usr/lib64/libcurl.so
-DAPR_INCLUDE_DIR=/usr/include/apr-1
-DAPR_LIBRARY=/usr/lib64/libapr-1.so
-DAPR_UTIL_INCLUDE_DIR=/usr/include/apr-1
-DAPR_UTIL_LIBRARY=/usr/lib64/libaprutil-1.so
-DMINIXML_INCLUDE_DIR=/usr/include
-DMINIXML_LIBRARY=/usr/lib64/libmxml.so
// If the 'Could not find APR-config/APR-1-config' error is reported during compilation, one possible reason is that the APR-1-config file does not exist in the default path. Add the following options:
-DAPR_CONFIG_BIN=/path/to/bin/apr-1-config
// If the 'Could not find apu-config/apu-1-config' error is reported, add the following options:
-DAPU_CONFIG_BIN=/path/to/bin/apu-1-config
```

Install OSS SDK for C on Windows

- Download the SDK

- [Download directly](#)
- [Download from GitHub](#)
- [Download previous versions](#)

- Install the SDK

For more information about how to use Visual Studio to compile OSS SDK for C, visit [Use Alibaba Cloud OSS SDK for C on Windows](#).

Note If you use Visual Studio 2012 or later, you are prompted to choose whether to upgrade the project version to adapt to the latest versions of the compiler and libraries. We recommend that you use versions that match your project. If the project uses the compiler and libraries of the latest versions, upgrade the project. Otherwise, you do not need to upgrade the project.

1.4.4.3. Initialization

You must initialize `oss_request_options_t` and specify an endpoint to use OSS SDK for C.

You must specify an endpoint when you initialize OSS SDK for C. For more information about endpoints, see [Obtain an endpoint](#).

You must specify an AccessKey pair when you initialize OSS SDK for C. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

Use the OSS domain name to initialize request options

The following code provides an example on how to use the OSS domain name to initialize request options:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters. The second parameter in this function indicates the ownership of ctl. By default, the value is 0. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main() {
    aos_pool_t *p;
    oss_request_options_t *options;
    /* Initialize global variables, which are run before the program starts. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        return -1;
    }
    /* Initialize the memory pool and options. */
    aos_pool_create(&p, NULL);
    options = oss_request_options_create(p);
    init_options(options);
    /* The code for logic operations is omitted. */
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(p);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

 **Note** For more information about how to set the request options, see [How to set parameters for cURL-based communication in OSS SDK for C](#).

Use the custom domain name to initialize request options

The following code provides an example on how to use custom domain name to initialize request options:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourCustomEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
void init_options(oss_request_options_t *options) {
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Enable the CNAME and bind the custom domain name to the bucket. */
    options->config->is_cname = 1;
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main() {
    aos_pool_t *p;
    oss_request_options_t *options;
    /* Initialize global variables, which are run before the program starts. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        return -1;
    }
    /* Initialize the memory pool and options. */
    aos_pool_create(&p, NULL);
    options = oss_request_options_create(p);
    init_options(options);
    /* The code for logic operations is omitted. */
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(p);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

Use STS to initialize request options

The following code provides an example on how to use STS to initialize request options:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *sts_token = "<yourSecurityToken>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Configure STS. */
    aos_str_set(&options->config->sts_token, sts_token);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main() {
    aos_pool_t *p;
    oss_request_options_t *options;
    /* Initialize global variables, which are run before the program starts. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        return -1;
    }
    /* Initialize the memory pool and options. */
    aos_pool_create(&p, NULL);
    options = oss_request_options_create(p);
    init_options(options);
    /* The code for logic operations is omitted. */
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(p);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

Configure a timeout period

The following code provides an example on how to configure a timeout period:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters. The second parameter in this function indicates the ownership of ctl. By default, the value is 0. */
    options->ctl = aos_http_controller_create(options->pool, 0);
    /* Set the connection timeout period. Default value: 10 seconds. */
    options->ctl->options->connect_timeout = 10;
    /* Set the DNS cache timeout period. Default value: 60 seconds. */
    options->ctl->options->dns_cache_timeout = 60;
    /*
    Configure the request timeout period:
    Set the value of speed_limit to specify the minimum acceptable speed. Default value: 1 KB/s.
    Set the value of speed_time to specify the minimum acceptable speed. Default value: 15 seconds.
    The default values indicate that if the transmission speed is continuously smaller than 1 KB/s for 15 consecutive seconds, the request times out.
    */
    options->ctl->options->speed_limit = 1024;
    options->ctl->options->speed_time = 15;
}
```

1.4.4.4. Getting Started

This topic describes how to use Object Storage Service (OSS) SDK for C to complete routine operations such as bucket creation, object upload, and object download.

Sample projects

- Sample projects for Linux
 - i. [Download a sample project](#) and extract it.
 - ii. When you install oss-c-sdk-demo-specified-installation, you must specify the installation directory: `**/home/your/oss/csdk**`. You do not need to specify installation directories for other sample projects because other sample projects are automatically installed based on OSS SDK for C and other dependent third-party libraries.

- iii. Replace `OSS_ENDPOINT`, `ACCESS_KEY_ID`, `ACCESS_KEY_SECRET`, and `BUCKET_NAME` in the sample project with valid values. If OSS SDK for C and the dynamic libraries of the dependent libraries are not in the system directory, use `LD_LIBRARY_PATH` to specify the directory of OSS SDK for C and the dynamic libraries of the dependent libraries.
- iv. Go to the `oss-c-sdk-demo-xxx` directory. Run `make` to compile the sample project. Run `./main` to run the executable program. To re-compile the project, run `make clean`.

For more information about the sample projects, see [Use Alibaba Cloud OSS SDK for C in Linux](#).

- Sample projects for Windows

- i. [Download a sample project](#) and extract it.

For more sample projects, visit [GitHub](#).

- ii. Use Visual Studio to open the sample project, and then set the following parameters in the `oss_config.c` file to valid values: `OSS_ENDPOINT`, `ACCESS_KEY_ID`, `ACCESS_KEY_SECRET`, `BUCKET_NAME`, `OBJECT_NAME`, `MULTIPART_UPLOAD_FILE_PATH`, and `DIR_NAME`. Compile and run the project. You can develop your own project based on the sample project.

 **Note** For more information about how to use Visual Studio, see [Use Alibaba Cloud OSS SDK for C in Windows](#).

Create a bucket

The following code provides an example on how to create a bucket:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Create the bucket. */
    aos_status_t *resp_status;
    aos_table_t *resp_headers;
    oss_acl_t oss_acl = OSS_ACL_PRIVATE;
    aos_string_t bucket;
    /* Assign the char* data to the bucket. */
    aos_str_set(&bucket, bucket_name);
    resp_status = oss_create_bucket(oss_client_options, &bucket, oss_acl, &resp_headers);
    /* Determine whether the bucket is created. */
    if (aos_status_is_ok(resp_status)) {
        printf("create bucket succeeded\n");
    } else {
        printf("create bucket failed\n");
    }

    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

Upload an object

The following code provides an example on how to upload objects to OSS by using stream upload:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_status_t *resp_status;
    aos_table_t *headers;
    aos_table_t *resp_headers;
    aos_list_t buffer;
    aos_buf_t *content = NULL;
    const char *data = "More than just cloud.";
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    headers = aos_table_make(pool, 0);
    /* Convert char* data to aos_list_t data. */
    aos_list_init(&buffer);
    content = aos_buf_pack(oss_client_options->pool, data, strlen(data));
    aos_list_add_tail(&content->node, &buffer);
    /* Upload the object. */
    resp_status = oss_put_object_from_buffer(oss_client_options, &bucket, &object, &buffer, headers, &resp_headers);
    /* Determine whether the object is uploaded. */
    if (aos_status_is_ok(resp_status)) {
        printf("put file succeeded\n");
    } else {
        printf("put file failed\n");
    }

    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Download an object

The following code provides an example on how to download a specified object to a local computer:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *download_filename = "<yourDownloadedFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_status_t *resp_status;
    aos_table_t *headers = NULL;
    aos_table_t *params = NULL;
    aos_table_t *resp_headers = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, download_filename);
    headers = aos_table_make(pool, 0);
    /* Download the object. */
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    /* Determine whether the object is downloaded. */
    if (aos_status_is_ok(resp_status)) {
        printf("get object to local file succeeded\n");
    } else {
        printf("get object to local file failed\n");
    }

    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

List objects

The following code provides an example on how to list objects in a specified bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool = NULL;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options = NULL;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    /* List objects. */
    resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
    /* Determine whether the objects are listed. */
    if (!aos_status_is_ok(resp_status))
    {
        printf("list object failed\n");
    }
    else{
        /* Display the objects. */
        printf("Object\tSize\tLastModified\n");
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_pstrprintf(pool, "%s\t%s\t%s\n", content->key.len, content->key.data,
                content->size.len, content->size.data,
                content->last_modified.len, content->last_modified.data);
            printf("%s", line);
        }
        printf("Total %d\n", size);
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete an object

The following code provides an example on how to delete an object:


```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool = NULL;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options = NULL;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_status_t *resp_status = NULL;
    aos_table_t *resp_headers = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Delete the object. */
    resp_status = oss_delete_object(oss_client_options, &bucket, &object, &resp_headers);
    /* Determine whether the object is deleted. */
    if (aos_status_is_ok(resp_status)) {
        printf("delete object succeeded\n");
    } else {
        printf("delete object failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.5. Buckets

1.4.4.5.1. Create buckets

A bucket is a container used to store objects in Object Storage Service (OSS). Each object is contained in a bucket. This topic describes how to create a bucket.

The following code provides an example on how to create a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    oss_acl_t oss_acl = OSS_ACL_PRIVATE;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    /* Assign char* data to a bucket of the aos_string_t type. */
    aos_str_set(&bucket, bucket_name);
    /* Create the bucket. */
    resp_status = oss_create_bucket(oss_client_options, &bucket, oss_acl, &resp_headers);
    /* Determine whether the bucket is created. */
    if (aos_status_is_ok(resp_status)) {
        printf("create bucket succeeded\n");
    } else {
        printf("create bucket failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

The following code provides an example on how to specify the access control list (ACL) of a bucket when you create the bucket:

```

/* Create a bucket and set the ACL of the bucket to public read. The default bucket ACL is private. */
resp_status = oss_create_bucket(oss_client_options, &bucket, OSS_ACL_PUBLIC_READ, &resp_headers);
if (aos_status_is_ok(resp_status)) {
    printf("create bucket succeeded\n");
} else {
    printf("create bucket failed\n");
}

```

1.4.4.5.2. List buckets

A bucket is a container that is used to store objects in Object Storage Service (OSS). All objects in OSS are contained in buckets. Bucket names are listed in alphabetical order. You can list buckets that belong to the current Alibaba Cloud account in all regions and meet the specific conditions.

The following code provides an example on how to list buckets that belong to the current Alibaba Cloud account in all regions:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t *pool is equivalent to apr_pool_t. The code to create a memory pool is included in the apr library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_list_buckets_params_t *params = NULL;
    oss_list_bucket_content_t *content = NULL;
    int size = 0;
    params = oss_create_list_buckets_params(pool);
    /* List the buckets. */
    resp_status = oss_list_bucket(oss_client_options, params, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("list buckets succeeded\n");
    } else {
        printf("list buckets failed\n");
    }
    /* Display the list of buckets. */
    aos_list_for_each_entry(oss_list_bucket_content_t, content, &params->bucket_list, node) {
        printf("BucketName: %s\n", content->name.data);
        ++size;
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

1.4.4.5.3. Manage bucket ACLs

A bucket is a container used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. This topic describes how to configure and query the access control list (ACL) of a bucket.

Configure ACL for a bucket

The following table describes the bucket ACLs.

ACL	Description	Value
Private	Only the owner or authorized users of the bucket can read and write the object.	OSS_ACL_PRIVATE
Public read	Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object. Exercise caution when you use this ACL.	OSS_ACL_PUBLIC_READ
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	OSS_ACL_PUBLIC_READ_WRITE

The following code provides an example on how to configure the ACL of a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    /* Assign char* data to the aos_string_t bucket. */
    aos_str_set(&bucket, bucket_name);
    /* Set the ACL of the bucket to public read (OSS_ACL_PUBLIC_READ). */
    resp_status = oss_put_bucket_acl(oss_client_options, &bucket, OSS_ACL_PUBLIC_READ, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("set bucket acl succeeded\n");
    } else {
        printf("set bucket acl failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Query the ACL of a bucket

The following code provides an example on how to query the ACL of a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information, such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t oss_acl;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    /* Assign char* data to the aos_string_t bucket. */
    aos_str_set(&bucket, bucket_name);
    /* Query the ACL of the bucket. */
    resp_status = oss_get_bucket_acl(oss_client_options, &bucket, &oss_acl, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket acl succeeded : %s \n", oss_acl.data);
    } else {
        printf("get bucket acl failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.5.4. Delete buckets

If you no longer use a bucket, delete the bucket to stop unnecessary charges.

Before you delete a bucket, ensure that all objects, LiveChannels, and parts that are generated from multipart upload in the bucket are deleted. The following code provides an example on how to delete a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t *pool is equivalent to apr_pool_t. The code to create a memory pool is included in the apr library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    /* Assign the char* data to the aos_string_t bucket. */
    aos_str_set(&bucket, bucket_name);
    /* Delete the bucket. */
    resp_status = oss_delete_bucket(oss_client_options, &bucket, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("delete bucket succeeded\n");
    } else {
        printf("delete bucket failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.5.5. Manage lifecycle rules

Object Storage Service (OSS) allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to Infrequent Access (IA) or Archive to reduce your storage costs. This topic describes how to manage lifecycle rules for buckets.

Background

Each lifecycle rule contains the following information:

- Policy: specifies the mode to match objects and parts.
 - Match by prefix: matches objects and parts by prefix. You can create multiple rules to configure different prefixes. Each prefix must be unique.
 - Match by bucket: matches all objects and parts contained in the bucket. You can create only one rule if you select this method.
- Object lifecycle policy: specifies the validity period or expiration data and the operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired objects are deleted.
 - Expiration date: specifies a date. Objects that are last modified before this date expire and are deleted.
- Part lifecycle policy: specifies the validity period or expiration time and the delete operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted.
 - Expiration date: specifies a date. Parts that are last modified before this date expire and are deleted.

Lifecycle rules also apply to the parts uploaded by using uploadPart. In this case, the last modified time of an object is the time the multipart upload task is initiated.

Configure lifecycle rules

The following code provides an example on how to configure lifecycle rules for a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_list_t lifecycle_rule_list;
    /* Create lifecycle rules for the bucket. */
    aos_str_set(&bucket, bucket_name);
    aos_list_init(&lifecycle_rule_list);
    /* Specify the validity period. */
    oss_lifecycle_rule_content_t *rule_content_days = oss_create_lifecycle_rule_content(pool);
    aos_str_set(&rule_content_days->id, "rule-1");
    aos_str_set(&rule_content_days->prefix, "obsoleted");
    aos_str_set(&rule_content_days->status, "Enabled");
    rule_content_days->days = 3;
    aos_list_add_tail(&rule_content_days->node, &lifecycle_rule_list);
    /* Specify the expiration date. */
    oss_lifecycle_rule_content_t *rule_content_date = oss_create_lifecycle_rule_content(pool);
    aos_str_set(&rule_content_date->id, "rule-2");
    aos_str_set(&rule_content_date->prefix, "delete");
    aos_str_set(&rule_content_date->status, "Enabled");
    aos_str_set(&rule_content_date->date, "2022-10-11T00:00:00.000Z");
    aos_list_add_tail(&rule_content_date->node, &lifecycle_rule_list);
    /* Configure the lifecycle rules. */
    resp_status = oss_put_bucket_lifecycle(oss_client_options, &bucket, &lifecycle_rule_list, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put bucket lifecycle succeeded\n");
    } else {
        printf("put bucket lifecycle failed, code:%d, error_code:%s, error_msg:%s, request_id:%s\n",
            resp_status->code, resp_status->error_code, resp_status->error_msg, resp_status->req_id);
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Query lifecycle rules

The following code provides an example on how to query lifecycle rules of a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_list_t lifecycle_rule_list;
    oss_lifecycle_rule_content_t *rule_content;
    char *rule_id;
    char *prefix;
    char *status;
    int days = INT_MAX;
    char *date = "";
    aos_str_set(&bucket, bucket_name);
    /* Query and display the lifecycle rules configured for the bucket. */
    aos_list_init(&lifecycle_rule_list);
    resp_status = oss_get_bucket_lifecycle(oss_client_options, &bucket, &lifecycle_rule_list, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket lifecycle succeeded\n");
        aos_list_for_each_entry(oss_lifecycle_rule_content_t, rule_content, &lifecycle_rule_list, node) {
            rule_id = apr_psprintf(pool, "%.s", rule_content->id.len, rule_content->id.data);
            prefix = apr_psprintf(pool, "%.s", rule_content->prefix.len, rule_content->prefix.data);
            status = apr_psprintf(pool, "%.s", rule_content->status.len, rule_content->status.data);
            date = apr_psprintf(pool, "%.s", rule_content->date.len, rule_content->date.data);
            days = rule_content->days;
        }
        printf("rule_id: %s \n", rule_id);
        printf("prefix: %s \n", prefix);
        printf("status: %s \n", status);
        printf("date: %s \n", date);
        printf("days: %d \n", days);
    } else {
        printf("get bucket lifecycle failed, code:%d, error_code:%s, error_msg:%s, request_id:%s\n",
            resp_status->code, resp_status->error_code, resp_status->error_msg, resp_status->req_id);
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete lifecycle rules

The following code provides an example on how to delete lifecycle rules from a bucket:


```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    /* Delete the lifecycle rules configured for the bucket. */
    aos_str_set(&bucket, bucket_name);
    resp_status = oss_delete_bucket_lifecycle(oss_client_options, &bucket, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("delete bucket lifecycle succeeded\n");
    } else {
        printf("delete bucket lifecycle failed, code:%d, error_code:%s, error_msg:%s, request_id:%s\n",
            resp_status->code, resp_status->error_code, resp_status->error_msg, resp_status->req_id);
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.5.6. CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to different regions. OSS provides CORS operations to facilitate cross-origin access control.

Configure CORS rules

Run the following code to configure CORS rules for a specified bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_list_t cors_rule_list;
    oss_cors_rule_t *cors_rule1 = NULL, *cors_rule2 = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_list_init(&cors_rule_list);
    cors_rule1 = oss_create_cors_rule(pool);
    aos_list_add_tail(&cors_rule1->node, &cors_rule_list);
    oss_create_sub_cors_rule(pool, &cors_rule1->allowed_origin_list, "allowed_origin_1_1");
    oss_create_sub_cors_rule(pool, &cors_rule1->allowed_origin_list, "allowed_origin_1_1");
    oss_create_sub_cors_rule(pool, &cors_rule1->allowed_method_list, "PUT");
    oss_create_sub_cors_rule(pool, &cors_rule1->allowed_method_list, "GET");
    oss_create_sub_cors_rule(pool, &cors_rule1->allowed_head_list, "Authorization");
    oss_create_sub_cors_rule(pool, &cors_rule1->expose_head_list, "expose_head_1_1");
    oss_create_sub_cors_rule(pool, &cors_rule1->expose_head_list, "expose_head_1_1");
    cors_rule2 = oss_create_cors_rule(pool);
    aos_list_add_tail(&cors_rule2->node, &cors_rule_list);
    oss_create_sub_cors_rule(pool, &cors_rule2->allowed_origin_list, "allowed_origin_2_1");
    oss_create_sub_cors_rule(pool, &cors_rule2->allowed_origin_list, "allowed_origin_2_2");
    oss_create_sub_cors_rule(pool, &cors_rule2->allowed_method_list, "PUT");
    oss_create_sub_cors_rule(pool, &cors_rule2->allowed_method_list, "GET");
    oss_create_sub_cors_rule(pool, &cors_rule2->allowed_head_list, "Authorization");
    oss_create_sub_cors_rule(pool, &cors_rule2->expose_head_list, "expose_head_2_1");
    oss_create_sub_cors_rule(pool, &cors_rule2->expose_head_list, "expose_head_2_2");
    /* Configure the CORS rules. */
    resp_status = oss_put_bucket_cors(oss_client_options, &bucket, &cors_rule_list, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put bucket cors succeeded\n");
    } else {
        printf("put bucket cors failed\n");
    }
}
/* Release the memory pool, which releases memory resources allocated for the request. */
aos_pool_destroy(pool);
/* Release the allocated global resources. */
aos_http_io_deinitialize();
return 0;
}

```

Query CORS rules

Run the following code to obtain CORS rules:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_list_t cors_rule_list;
    oss_cors_rule_t *cors_rule = NULL;
    oss_sub_cors_rule_t *sub_cors_rule = NULL;
    aos_str_set(&bucket, bucket_name);
    /* Query the CORS rules. */
    aos_list_init(&cors_rule_list);
    resp_status = oss_get_bucket_cors(oss_client_options, &bucket, &cors_rule_list, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket cors succeeded\n");
        aos_list_for_each_entry(oss_cors_rule_t, cors_rule, &cors_rule_list, node) {
            printf("max_age_seconds: %d\n", cors_rule->max_age_seconds);
            aos_list_for_each_entry(oss_sub_cors_rule_t, sub_cors_rule, &cors_rule->allowed_origin_list, node) {
                printf("allowed_origin_list: %s\n", sub_cors_rule->rule.data);
            }
            aos_list_for_each_entry(oss_sub_cors_rule_t, sub_cors_rule, &cors_rule->allowed_method_list, node) {
                printf("allowed_method_list: %s\n", sub_cors_rule->rule.data);
            }
            aos_list_for_each_entry(oss_sub_cors_rule_t, sub_cors_rule, &cors_rule->allowed_head_list, node) {
                printf("allowed_head_list: %s\n", sub_cors_rule->rule.data);
            }
            aos_list_for_each_entry(oss_sub_cors_rule_t, sub_cors_rule, &cors_rule->expose_head_list, node) {
                printf("expose_head_list: %s\n", sub_cors_rule->rule.data);
            }
        }
    } else {
        printf("get bucket cors failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete CORS rules

Run the following code to delete all CORS rules for the specified bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    /* Delete the CORS rules. */
    resp_status = oss_delete_bucket_cors(oss_client_options, &bucket, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket cors succeeded\n");
    } else {
        printf("get bucket cors failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.5.7. Configure hotlink protection

OSS provides hotlink protection to prevent unauthorized domain names from accessing your OSS data. You can configure the Referer field in the HTTP header.

To configure hotlink protection, you must configure the following parameters:

- **Referer Whitelist:** Only specified domain names are allowed to access OSS resources.
- **Allow Empty Referer:** If you do not allow empty Refer fields, a request is allowed to access OSS resources only if the request includes the Referer field configured in the HTTP or HTTPS header.

Configure hotlink protection

Run the following code to configure the referer whitelist:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_referer_config_t referer_config;
    aos_str_set(&bucket, bucket_name);
    aos_list_init(&referer_config.referer_list);
    oss_create_and_add_refer(pool, &referer_config, "http://www.aliyun.com");
    oss_create_and_add_refer(pool, &referer_config, "https://www.aliyun.com");
    referer_config.allow_empty_referer = 0;
    /* Add a Referer whitelist. You can use an asterisk (*) or a question mark (?) as a wildcard character to set the Referer parameter. */
    resp_status = oss_put_bucket_referer(oss_client_options, &bucket, &referer_config, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put bucket referer succeeded\n");
    } else {
        printf("put bucket referer failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Query hotlink protection information

Run the following code to obtain a referer whitelist:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_referer_config_t referer_config;
    oss_referer_t *referer;
    aos_str_set(&bucket, bucket_name);
    aos_list_init(&referer_config.referer_list);
    /* Query the Referer whitelist of the bucket. */
    resp_status = oss_get_bucket_referer(oss_client_options, &bucket, &referer_config, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket referer succeeded\n");
        aos_list_for_each_entry(oss_referer_t, referer, &referer_config.referer_list, node) {
            printf("get referer %s\n", referer->referer.data);
        }
    } else {
        printf("get bucket referer failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete hotlink protection configurations

Run the following code to clear a referer whitelist:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_referer_config_t referer_config;
    aos_str_set(&bucket, bucket_name);
    aos_list_init(&referer_config.referer_list);
    referer_config.allow_empty_referer = 1;
    /* You cannot clear hotlink protection configurations directly. To delete hotlink protection configurations, you must create a rule that allows an empty Referer field to replace the existing rule. */
    resp_status = oss_put_bucket_referer(oss_client_options, &bucket, &referer_config, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("delete bucket referer succeeded\n");
    } else {
        printf("delete bucket referer failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.5.8. Configure static website hosting

You can set the static website hosting mode for your bucket. After configurations take effect, you can use the bucket domain name to access this static website and be redirected to a specified index page or error page.

Configure static website hosting

Run the following code to configure static website hosting:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_website_config_t website_config;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&website_config.suffix_str, "index.html");
    aos_str_set(&website_config.key_str, "error.html");
    /* Configure static web hosting: The index page is index.html and the error page is error.html. */
    resp_status = oss_put_bucket_website(oss_client_options, &bucket, &website_config, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put bucket website succeeded\n");
    } else {
        printf("put bucket website failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

View static website hosting configurations

Run the following code to view static website hosting configurations:


```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_website_config_t website_config;
    aos_str_set(&bucket, bucket_name);
    /* View the static website hosting configurations. */
    resp_status = oss_get_bucket_website(oss_client_options, &bucket, &website_config, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket website succeeded\n");
        printf("website_config: %s %s\n", website_config.suffix_str.data, website_config.key_str.data);
    } else {
        printf("get bucket website failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    /* Delete the static website hosting configurations. */
    resp_status = oss_delete_bucket_website(oss_client_options, &bucket, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("delete bucket website succeeded\n");
    } else {
        printf("delete bucket website failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

1.4.4.6. Upload objects

1.4.4.6.1. Overview

Objects are the basic units for data operations in OSS. OSS SDK for C provides a variety of methods to upload objects.

You can upload an object to OSS by using one of the following methods:

- **Simple upload:** includes stream upload and object upload. You can use this method to upload an object up to 5 GB in size.
- **Append upload:** uploads an object up to 5 GB in size.
- **Resumable upload:** supports concurrent upload and resumable upload. You can define the size of each part. This method is suitable for uploading large objects. You can use this method to upload an object up to 48.8 TB in size.
- **Multipart upload:** uploads an object up to 48.8 TB in size. This method is suitable for uploading large objects.

During object upload, you can configure object metadata and view upload progress by using a progress bar. For more information, see [Use progress bars](#).

1.4.4.6.2. Simple upload

This topic describes how to use simple upload.

For the complete code that is used to perform simple upload, visit [GitHub](#).

Upload data from memory

The following code provides an example on how to upload data from memory:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *object_content = "More than just cloud.";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_list_t buffer;
    aos_buf_t *content = NULL;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_list_init(&buffer);
    content = aos_buf_pack(oss_client_options->pool, object_content, strlen(object_content));
    aos_list_add_tail(&content->node, &buffer);
    /* Upload the object. */
    resp_status = oss_put_object_from_buffer(oss_client_options, &bucket, &object, &buffer, headers, &resp_headers);
    /* Determine whether the object is uploaded. */
    if (aos_status_is_ok(resp_status)) {
        printf("put object from buffer succeeded\n");
    } else {
        printf("put object from buffer failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Upload a local file

The following code provides an example on how to upload a local file to OSS:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    /* Upload the object. */
    resp_status = oss_put_object_from_file(oss_client_options, &bucket, &object, &file, headers, &resp_headers);
    /* Determine whether the object is uploaded. */
    if (aos_status_is_ok(resp_status)) {
        printf("put object from file succeeded\n");
    } else {
        printf("put object from file failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

FAQ: How can I upload objects whose names contain Chinese characters?

On Windows systems, you can use UTF-8 to encode the object names that contain Chinese characters and upload the object. The following code provides an example on how to upload objects whose names contain Chinese characters:

```

#include "oss_api.h"
#include "aos_http_io.h"
#include <wchar.h>
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename_ch = "c:\\测试.txt";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
}

```

```

/* Configure network parameters such as timeout periods. */
options->ctl = aos_http_controller_create(options->pool, 0);
}

char* multibyte_to_utf8(const char *ch, char *str, int size)
{
    if (!ch || !str || size <= 0)
        return NULL;
    int chlen = strlen(ch);
    int multi_byte_cnt = 0;
    for (int i = 0; i < chlen - 1; i++) {
        if ((ch[i] & 0x80) && (ch[i + 1] & 0x80)) {
            i++;
            multi_byte_cnt++;
        }
    }
    if (multi_byte_cnt == 0)
        return ch;
    int len = MultiByteToWideChar(CP_ACP, 0, ch, -1, NULL, 0);
    if (len <= 0)
        return NULL;
    wchar_t *wch = malloc(sizeof(wchar_t) * (len + 1));
    if (!wch)
        return NULL;
    wmemset(wch, 0, len + 1);
    MultiByteToWideChar(CP_ACP, 0, ch, -1, wch, len);
    len = WideCharToMultiByte(CP_UTF8, 0, wch, -1, NULL, 0, NULL, NULL);
    if ((len <= 0) || (size < len + 1)) {
        free(wch);
        return NULL;
    }
    memset(str, 0, len + 1);
    WideCharToMultiByte(CP_UTF8, 0, wch, -1, str, len, NULL, NULL);
    free(wch);
    return str;
}

int main(int argc, char argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, i
s_cname, and curl. */
    oss_request_options_t oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Use UTF-8 to encode the object names that contain multibyte Chinese characters */
    char local_filename_buf[1024];
    char *local_filename = multibyte_to_utf8(local_filename_ch, local_filename_buf, 1024);
    aos_str_set(&file, local_filename);
    /* Upload the object. */
    resp_status = oss_put_object_from_file(oss_client_options, &bucket, &object, &file, headers, &resp_headers);
    /* Determine whether the object is uploaded. */
    if (aos_status_is_ok(resp_status)) {
        printf("put object from file succeeded\n");
    } else {
        printf("put object from file failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.6.3. Append upload

You can append strings and files during append upload. You cannot perform CopyObject for append objects.

 **Note** You cannot perform CopyObject for append objects.

Append data from memory

The following code provides an example on how to append data from memory:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *object_content = "More than just cloud.";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_list_t buffer;
    int64_t position = 0;
    char *next_append_position = NULL;
    aos_buf_t *content = NULL;
    aos_table_t *headers1 = NULL;
    aos_table_t *headers2 = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    headers1 = aos_table_make(pool, 0);
    /* Obtain the position where the object is appended for the first time. */
    resp_status = oss_head_object(oss_client_options, &bucket, &object, headers1, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        next_append_position = (char*)(apr_table_get(resp_headers, "x-oss-next-append-position"));
        position = atoi(next_append_position);
    }

    /* Append the object. */
    headers2 = aos_table_make(pool, 0);
    aos_list_init(&buffer);
    content = aos_buf_pack(pool, object_content, strlen(object_content));
    aos_list_add_tail(&content->node, &buffer);
    resp_status = oss_append_object_from_buffer(oss_client_options, &bucket, &object, position, &buffer, headers2, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("append object from buffer succeeded\n");
    } else {
        printf("append object from buffer failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

Append data from a local file

The following code provides an example on how to append data from a local file:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    int64_t position = 0;
    char *next_append_position = NULL;
    aos_table_t *headers1 = NULL;
    aos_table_t *headers2 = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    /* Obtain the position where the object is appended for the first time. */
    resp_status = oss_head_object(oss_client_options, &bucket, &object, headers1, &resp_headers);
    if(aos_status_is_ok(resp_status)) {
        next_append_position = (char*)(apr_table_get(resp_headers, "x-oss-next-append-position"));
        position = atoi(next_append_position);
    }
    /* Append the object. */
    resp_status = oss_append_object_from_file(oss_client_options, &bucket, &object, position, &file, headers2, &resp_headers);
    /* Determine whether the append upload is successful. */
    if (aos_status_is_ok(resp_status)) {
        printf("append object from file succeeded\n");
    } else {
        printf("append object from file failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

1.4.4.6.4. Resumable upload

If the system stops responding during multipart upload, you can resume the upload by using the ListMultipartUploads and ListParts operations to retrieve all multipart upload tasks on an object and list the uploaded parts in each task. This method allows an upload task to be resumed from the last uploaded part.

The progress of the current upload is recorded in the checkpoint file during the upload process. If the upload of a part fails during the process, the next upload attempt starts from the recorded position in the checkpoint file. The checkpoint file is deleted after the resumable upload task is completed.

You can use `oss_resumable_cli_params_t` to perform resumable upload. The following table describes common parameters used in this method:

Parameter	Description
thread_num	The number of concurrent threads. Default value: 1.

Parameter	Description
enable_checkpoint	Specifies whether to enable resumable upload. This feature is disabled by default.
partSize	The size of each part. The part size ranges from 100 KB to 5 GB.
checkpoint_path	The path of the checkpoint file. Default value: {upload_file_path}.cp.

The following code provides an example on how to perform resumable upload:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_list_t resp_body;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_resumable_clt_params_t *clt_params;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    aos_list_init(&resp_body);
    /* Start resumable upload. */
    clt_params = oss_create_resumable_clt_params_content(pool, 1024 * 100, 3, AOS_TRUE, NULL);
    resp_status = oss_resumable_upload_file(oss_client_options, &bucket, &object, &file, headers, NULL, clt_params, NULL, &resp_headers, &resp_body);
    if (aos_status_is_ok(resp_status)) {
        printf("resumable upload succeeded\n");
    } else {
        printf("resumable upload failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

1.4.4.6.5. Multipart upload

OSS allows you to use multipart upload to split an object into several parts and upload them separately. After all parts are uploaded, you can combine the uploaded parts into a complete object to complete the upload.

To implement multipart upload, perform the following operations:

The following code provides an example on how to perform multipart upload:


```

#include "oss_api.h"
#include "aos_http_io.h"
#include <sys/stat.h>
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int64_t get_file_size(const char *file_path)
{
    int64_t filesize = -1;
    struct stat statbuff;
    if(stat(file_path, &statbuff) < 0){
        return filesize;
    } else {
        filesize = statbuff.st_size;
    }
    return filesize;
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    oss_upload_file_t *upload_file = NULL;
    aos_string_t upload_id;
    aos_table_t *headers = NULL;
    aos_table_t *complete_headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_null(&upload_id);
    headers = aos_table_make(pool, 1);
    complete_headers = aos_table_make(pool, 1);
    int part_num = 1;
    /* Initiate a multipart upload task and obtain an upload ID. */
    resp_status = oss_init_multipart_upload(oss_client_options, &bucket, &object, &upload_id, headers, &resp_headers);
    /* Determine whether the multipart upload task is initiated. */
    if (aos_status_is_ok(resp_status)) {
        printf("Init multipart upload succeeded, upload_id: %s\n",
            upload_id.len, upload_id.data);
    } else {
        printf("Init multipart upload failed, upload_id: %s\n",
            upload_id.len, upload_id.data);
    }
    /* Upload the parts. */
    int64_t file_length = 0;
    int64_t pos = 0;
    file_length = get_file_size(local_filename);
    while(pos < file_length) {
        upload_file = oss_create_upload_file(pool);
        aos_str_set(&upload_file->filename, local_filename);
        upload_file->file_pos = pos;
        pos += 100 * 1024;
        upload_file->file_last = pos < file_length ? pos : file_length;
        resp_status = oss_upload_part_from_file(oss_client_options, &bucket, &object, &upload_id, part_num++, upload_file, &resp_headers);
        if (aos_status_is_ok(resp_status)) {

```

```

    if (aos_status_is_ok(resp_status)) {
        printf("Multipart upload part from file succeeded\n");
    } else {
        printf("Multipart upload part from file failed\n");
    }
}

oss_list_upload_part_params_t *params = NULL;
aos_list_t complete_part_list;
/* Query the uploaded parts. */
params = oss_create_list_upload_part_params(pool);
params->max_ret = 1000;
aos_list_init(&complete_part_list);
resp_status = oss_list_upload_part(oss_client_options, &bucket, &object, &upload_id, params, &resp_headers);
/* Determine whether all parts are listed. */
if (aos_status_is_ok(resp_status)) {
    printf("List multipart succeeded\n");
} else {
    printf("List multipart failed\n");
}

oss_complete_part_content_t *complete_part_content = NULL;
oss_list_part_content_t *part_content = NULL;
aos_list_for_each_entry(oss_list_part_content_t, part_content, &params->part_list, node) {
    complete_part_content = oss_create_complete_part_content(pool);
    aos_str_set(&complete_part_content->part_number, part_content->part_number.data);
    aos_str_set(&complete_part_content->etag, part_content->etag.data);
    aos_list_add_tail(&complete_part_content->node, &complete_part_list);
}
/* Complete the multipart upload task. */
resp_status = oss_complete_multipart_upload(oss_client_options, &bucket, &object, &upload_id,
    &complete_part_list, complete_headers, &resp_headers);
/* Determine whether the multipart upload task is complete. */
if (aos_status_is_ok(resp_status)) {
    printf("Complete multipart upload from file succeeded, upload_id: %s\n",
        upload_id.data);
} else {
    printf("Complete multipart upload from file failed\n");
}

/* Release the memory pool, which releases memory resources allocated for the request. */
aos_pool_destroy(pool);
/* Release the allocated global resources. */
aos_http_io_deinitialize();
return 0;
}

```

Note Query the uploaded parts. You must provide the ETag value of each part to call the `oss_complete_multipart_upload` operation. You can use one of the following methods to obtain the ETag values: First, the response of a part upload contains the ETag value of the part. You can save the values for future use. Second, you can call the `oss_list_upload_part` operation to query the ETag values of the uploaded parts. In this example, the second method is used.

Cancel a multipart upload task

The following code provides an example on how to cancel a multipart upload task:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t upload_id;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_null(&upload_id);
    /* Initiate a multipart upload task and obtain an upload ID. */
    resp_status = oss_init_multipart_upload(oss_client_options, &bucket, &object, &upload_id, headers, &resp_headers);
    /* Determine whether the multipart upload task is initiated. */
    if (aos_status_is_ok(resp_status)) {
        printf("Init multipart upload succeeded, upload_id: %s\n",
            upload_id.len, upload_id.data);
    } else {
        printf("Init multipart upload failed, upload_id: %s\n",
            upload_id.len, upload_id.data);
    }
    /* Cancel the multipart upload task. */
    resp_status = oss_abort_multipart_upload(oss_client_options, &bucket, &object, &upload_id, &resp_headers);
    /* Determine whether the multipart upload task is canceled. */
    if (aos_status_is_ok(resp_status)) {
        printf("Abort multipart upload succeeded, upload_id: %s\n",
            upload_id.len, upload_id.data);
    } else {
        printf("Abort multipart upload failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

 **Note** If a multipart upload task is canceled, the upload ID can no longer be used to upload any parts. The parts uploaded by using that upload ID are also deleted.

1.4.4.6.6. Use progress bars

Progress bars are used to indicate the progress of an upload or download.

For the complete code of upload progress bars, visit [GitHub](#).

The following code uses PutObject as an example to show you how to view progress information:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
void percentage(int64_t consumed_bytes, int64_t total_bytes)
{
    assert(total_bytes >= consumed_bytes);
    printf("%%% APR_INT64_T_FMT "\n", consumed_bytes * 100 / total_bytes);
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_list_t resp_body;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    /* View the upload progress information when you upload an object. */
    resp_status = oss_do_put_object_from_file(oss_client_options, &bucket, &object, &file, NULL, NULL, percentage, &resp_headers, &resp_body);
    if (aos_status_is_ok(resp_status)) {
        printf("put object from file succeeded\n");
    } else {
        printf("put object from file failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

You cannot use progress bars when you use `oss_put_object_from_file` and `oss_put_object_from_buffer`.

1.4.4.7. Download objects

1.4.4.7.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for C provides a variety of methods to upload objects.

For the complete object download code, visit [GitHub](#).

You can use one of the following methods to download a single object:

- [Download to local files](#)
- [Download to local memory](#)
- [Range download](#)
- [Resumable download](#)

You can view the download progress by using the progress bar. For more information, see [Use progress bars](#).

1.4.4.7.2. Download to local files

This topic describes how to download an object to a local file.

For the complete code of object download, visit [GitHub](#).

The following code provides an example on how to download a specified object to a local file:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *params;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    params = aos_table_make(pool, 0);
    /* Download the object to your local file. If the name of the object is the same as that of the local file, the object replaces the local file. If the name of the object is different from that of the local file, the object is downloaded. */
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("Get object from file succeeded\n");
    } else {
        printf("Get object from file failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

 **Note** Compared with OSS SDK for C V1.0.0, this version adds the params parameter to the oss_get_object_to_file operation, and the values of the headers and params parameters can be NULL.

1.4.4.7.3. Download to local memory

This topic describes how to download an object to local memory.

For the complete code that is used to download objects, visit [GitHub](#).

The following code provides an example on how to download a specified object to local memory:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_list_t buffer;
    aos_buf_t *content = NULL;
    aos_table_t *params = NULL;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    char *buf = NULL;
    int64_t len = 0;
    int64_t size = 0;
    int64_t pos = 0;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_list_init(&buffer);
    /* Download the object to local memory. */
    resp_status = oss_get_object_to_buffer(oss_client_options, &bucket, &object,
                                         headers, params, &buffer, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to buffer succeeded\n");
        /* Copy the downloaded content to the buffer. */
        len = aos_buf_list_len(&buffer);
        buf = aos_palloc(pool, len + 1);
        buf[len] = '\0';
        aos_list_for_each_entry(aos_buf_t, content, &buffer, node) {
            size = aos_buf_size(content);
            memcpy(buf + pos, content->pos, size);
            pos += size;
        }
    }
    else {
        printf("get object to buffer failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.7.4. Range download

If you need only part of the data in an object, you can use range download to download data within the specified range.

The following code provides an example on how to perform range download:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_list_t buffer;
    aos_buf_t *content = NULL;
    aos_table_t *params = NULL;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    char *buf = NULL;
    int64_t len = 0;
    int64_t size = 0;
    int64_t pos = 0;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_list_init(&buffer);
    headers = aos_table_make(pool, 1);
    /* Set the range to byte 20 to byte 100. */
    apr_table_set(headers, "Range", "bytes=20-100");
    /* Download the object to local memory. */
    resp_status = oss_get_object_to_buffer(oss_client_options, &bucket, &object, headers, params, &buffer, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to buffer succeeded\n");
        /* Obtain the buffer length. */
        aos_list_for_each_entry(aos_buf_t, content, &buffer, node) {
            len += aos_buf_size(content);
        }
        buf = aos_palloc(pool, (apr_size_t)(len + 1));
        buf[len] = '\0';
        /* Copy the content in the buffer to the memory. */
        aos_list_for_each_entry(aos_buf_t, content, &buffer, node) {
            size = aos_buf_size(content);
            memcpy(buf + pos, content->pos, (size_t)size);
            pos += size;
        }
    }
    else {
        printf("get object to buffer failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.7.5. Resumable download

You may fail to download a large object if the network is unstable or other exceptions occur. In some cases, you may still fail to download the object even after multiple attempts. To solve this problem, OSS provides the resumable download feature.

You can use the `oss_resumable_clt_params_t` method to implement resumable download. The following table describes the parameters included in `oss_resumable_clt_params_t`.

Parameter	Description
<code>thread_num</code>	The number of concurrent threads. Default value: 1.
<code>enable_checkpoint</code>	Specifies whether to enable resumable upload. By default, resumable upload is disabled.
<code>partSize</code>	The size of each part. Valid values: 1 byte to 5 GB.
<code>checkpoint_path</code>	The path of the checkpoint file. Default value: <code>{upload_file_path}.cp</code> .

The `checkpoint` file stores information about the download progress. If you fail to download a part, the download continues based on the progress recorded in the `checkpoint` file. The `checkpoint` file is deleted after resumable download is complete.

Run the following code for resumable download:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_resumable_clt_params_t *clt_params;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    /* Use resumable download to download the object. */
    clt_params = oss_create_resumable_clt_params_content(pool, 1024 * 100, 3, AOS_TRUE, NULL);
    resp_status = oss_resumable_download_file(oss_client_options, &bucket, &object, &file, headers, NULL, clt_params, NULL, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("download succeeded\n");
    } else {
        printf("download failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```


1.4.4.7.6. Use progress bars

Progress bars are used to indicate the progress of an upload or download.

For the complete code of download progress bars, visit [GitHub](#).

The following code provides an example on how to view the progress information:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
const char *download_filename = "<yourDownloadFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

void percentage(int64_t consumed_bytes, int64_t total_bytes)
{
    assert(total_bytes >= consumed_bytes);
    printf("%%%" APR_INT64_T_FMT "\n", consumed_bytes * 100 / total_bytes);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *resp_headers = NULL;
    aos_list_t resp_body;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    /* View the upload progress while you upload an object. */
    resp_status = oss_do_put_object_from_file(oss_client_options, &bucket, &object, &file, NULL, NULL, percentage, &resp_headers, &resp_body);
    if (aos_status_is_ok(resp_status)) {
        printf("put object from file succeeded\n");
    } else {
        printf("put object from file failed\n");
    }

    /* View the download progress while you download an object. */
    aos_pool_create(&pool, NULL);
    oss_client_options = oss_request_options_create(pool);
    init_options(oss_client_options);
    aos_str_set(&file, download_filename);
    resp_status = oss_do_get_object_to_file(oss_client_options, &bucket, &object, NULL, NULL, &file, percentage, NULL);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

`oss_get_object_from_file` and `oss_get_object_from_buffer` does not support the progress bar feature.

1.4.4.8. Manage objects

1.4.4.8.1. Overview

You can manage objects in a bucket by using a variety of operations including `SetObjectAcl`, `GetObjectAcl`, `ListObjects`, `DeleteObject`, and `CopyObject`.

By using the preceding operations, you can perform the following operations:

- [Manage object ACLs](#)
- [Manage object metadata](#)
- [List objects](#)
- [Delete objects](#)
- [Copy objects](#)

1.4.4.8.2. Manage object ACLs

This topic describes how to manage the access control list (ACL) of an object.

The following table describes the ACLs that can be configured for an object.

ACL	Description	Value
Inherited from the bucket	The ACL of an object is the same as that of the bucket in which the object is stored.	OSS_ACL_DEFAULT
Private	Only the owner or authorized users of the bucket can read and write the object.	OSS_ACL_PRIVATE
Public read	Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object. Exercise caution when you use this ACL.	OSS_ACL_PUBLIC_READ
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	OSS_ACL_PUBLIC_READ_WRITE

The ACL of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

You can run the following code to configure an ACL for an object:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    oss_acl_e oss_acl = OSS_ACL_PUBLIC_READ;
    /* Configure the object ACL. */
    resp_status = oss_put_object_acl(oss_client_options, &bucket, &object, oss_acl, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put object acl success! \n");
    } else {
        printf("put object acl failed! \n");
    }

    /* Query the ACL of the object. */
    aos_string_t oss_acl_string;
    resp_status = oss_get_object_acl(oss_client_options, &bucket, &object, &oss_acl_string, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object acl success! \n");
        printf("acl: %s \n", oss_acl_string.data);
    } else {
        printf("get object acl failed! \n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.8.3. Manage object metadata

After you upload an object to OSS or before you read an object from OSS, you can use the `oss_get_object_meta` operation to query object metadata such as the content length and object type.

The following code provides an example on how to configure and query object metadata:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t *headers;
    aos_list_t buffer;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_buf_t *content = NULL;
    char *content_length_str = NULL;
    char *object_type = NULL;
    char *object_author = NULL;
    int64_t content_length = 0;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    headers = aos_table_make(pool, 2);
    /* Configure user metadata. */
    apr_table_set(headers, "Expires", "Fri, 28 Feb 2032 05:38:42 GMT");
    apr_table_set(headers, "x-oss-meta-author", "oss");
    aos_list_init(&buffer);
    content = aos_buf_pack(oss_client_options->pool, object_content, strlen(object_content));
    aos_list_add_tail(&content->node, &buffer);
    /* Configure the object ACL. Upload object from the buffer. */
    resp_status = oss_put_object_from_buffer(oss_client_options, &bucket, &object,
        &buffer, headers, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put object from buffer with md5 succeeded\n");
    } else {
        printf("put object from buffer with md5 failed\n");
    }

    /* Query the ACL of the object. */
    resp_status = oss_get_object_meta(oss_client_options, &bucket, &object, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        content_length_str = (char*)apr_table_get(resp_headers, OSS_CONTENT_LENGTH);
        if (content_length_str != NULL) {
            content_length = atol(content_length_str);
        }
        object_author = (char*)apr_table_get(resp_headers, OSS_AUTHORIZATION);
        object_type = (char*)apr_table_get(resp_headers, OSS_OBJECT_TYPE);
        printf("get object meta succeeded, object author:%s, object type:%s, content length:%ld\n", object_author, object_type, content_length);
    } else {
        printf("req:%s, get object meta failed\n", resp_status->req_id);
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

 **Note** For more information about HTTP headers, see [RFC 2616](#).

1.4.4.8.4. List objects

This topic describes how to list the objects stored in a bucket.

For the complete code used to list objects, visit [GitHub](#).

Objects are listed in alphabetical order. You can use `oss_list_object` to list objects in a bucket. The following table describes common parameters.

Parameter	Description
delimiter	Groups objects by name. Objects whose names contain the same string from the prefix and the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes.
prefix	Specifies the prefix that returned object names must contain.
max_ret	Specifies the maximum number of objects that can be listed at a time. The default value is 100. The maximum value is 1000.
marker	Specifies the name of the object after which the listing begins.

List all objects

The following code provides an example on how to list all objects in a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the apr library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    char *prefix = "";
    char *nextMarker = "";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    params->max_ret = 100;
    aos_str_set(&params->prefix, prefix);
    aos_str_set(&params->marker, nextMarker);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    do {
        resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
        if (!aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_psprintf(pool, "%.s\t%.s\t%.s\n", content->key.len, content->key.data,
                                content->size.len, content->size.data,
                                content->last_modified.len, content->last_modified.data);
            printf("%s", line);
        }
        nextMarker = apr_psprintf(pool, "%.s", params->next_marker.len, params->next_marker.data);
        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    printf("Total %d\n", size);
    /* Release the memory pool. The memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

② **Note** If a bucket contains more than 1,000 objects, only the first 1,000 objects are returned, and the value of the truncated parameter in the response is true. Additionally, the next_marker parameter is returned as the start point for the next read. To list more than 1,000 objects at a time, you can modify the max_ret parameter. You can also use the marker parameter to perform multiple read operations.

List a specified number of objects

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. The code used to create a memory pool is included in the APR library. aos_pool_t is equivalent to apr_pool_t. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* The memory consumed by options is allocated by the memory pool and is released with the pool. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    char *prefix = "";
    char *nextMarker = "";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    /* Set the maximum number of returned objects to 200. */
    params->max_ret = 200;
    aos_str_set(&params->prefix, prefix);
    aos_str_set(&params->marker, nextMarker);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
    if (!aos_status_is_ok(resp_status))
    {
        printf("list object failed\n");
    }
    aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
        ++size;
        line = apr_pstrprintf(pool, "%.s\t%.s\t%.s\n", content->key.len, content->key.data,
            content->size.len, content->size.data,
            content->last_modified.len, content->last_modified.data);
        printf("%s", line);
    }
    printf("Total %d\n", size);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Before you close the program, call aos_http_io_deinitialize to release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

List objects whose names contain the specified prefix

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOS_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. The code used to create a memory pool is included in the APR library. aos_pool_t is equivalent to apr_pool_t. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* The memory consumed by options is allocated by the memory pool and is released with the pool. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    /* List the objects whose names contain the specified prefix. */
    char *prefix = "<yourObjectPefix>";
    char *nextMarker = "";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    params->max_ret = 100;
    aos_str_set(&params->prefix, prefix);
    aos_str_set(&params->marker, nextMarker);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    do {
        resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
        if (! aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_psprintf(pool, "%.s\t%.s\t%.s\n", content->key.len, content->key.data,
                content->size.len, content->size.data,
                content->last_modified.len, content->last_modified.data);
            printf("%s", line);
        }
        nextMarker = apr_psprintf(pool, "%.s", params->next_marker.len, params->next_marker.data);
        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    printf("Total %d\n", size);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Before you close the program, call aos_http_io_deinitialize to release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

List objects that come after a specified marker


```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOS_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. The code used to create a memory pool is included in the APR library. aos_pool_t is equivalent to apr_pool_t. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* The memory consumed by options is allocated by the memory pool and is released with the pool. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    char *prefix = "";
    /* List objects that come after the specified marker. */
    char *nextMarker = "<yourObjectMarker>";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    params->max_ret = 100;
    aos_str_set(&params->prefix, prefix);
    aos_str_set(&params->marker, nextMarker);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    do {
        resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
        if (!aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }

        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_psprintf(pool, "%.s\t%.s\t%.s\n", content->key.data,
                content->size.data,
                content->last_modified.data);
            printf("%s", line);
        }
        nextMarker = apr_psprintf(pool, "%.s", params->next_marker.data);
        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    printf("Total %d\n", size);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Before you close the program, call aos_http_io_deinitialize to release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Folder simulation

OSS does not use folders. All elements are stored as objects. To simulate a folder on the OSS console, you actually create a 0 MB object whose name ends with a forward slash (/). This object can be uploaded and downloaded. By default, the OSS console displays an object whose name ends with a forward slash (/) as a folder.

The delimiter and prefix parameters can be used to simulate folder functions.

- **Prefix:** specifies the prefix as the name of a folder. The folder is used to list all files (in the folder) and subfolders (directories in the folder) that start with this prefix. These files and subfolders are included in the Objects list.
- **Delimiter:** If the delimiter is specified as a forward slash (/), only the files and subfolders (directories) are displayed. Subfolders (directories) are included in the CommonPrefixes list, while the files and folders in subfolders are not displayed.

For the complete code used to create a folder, visit [GitHub](#).

Assume that the following objects are stored in a bucket: oss.jpg, fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. Forward slashes (/) are used as delimiters for folders. The subsequent examples show how to simulate folder functions.

- List all objects in a bucket

Run the following code to list all objects in a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the apr
    library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret,
    is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    char *nextMarker = "";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    params->max_ret = 100;
    aos_str_set(&params->marker, nextMarker);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    do {
        resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
        if (! aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_pstrprintf(pool, "%s\t%s\t%s\n", content->key.len, content->key.data,
                content->size.len, content->size.data,
                content->last_modified.len, content->last_modified.data);
            printf("%s", line);
        }
        nextMarker = apr_pstrprintf(pool, "%s", params->next_marker.len, params->next_marker.data);
        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    printf("Total %d\n", size);
    /* Release the memory pool. The memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

The returned results are as follows:

```

Objects:
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
oss.jpg
CommonPrefixes:

```

- List all objects in a specified folder

Run the following code to list all objects in a specified folder:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOS_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. The code used to create a memory pool is included in the APR library. aos_pool_t is equivalent to apr_pool_t. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* The memory consumed by options is allocated by the memory pool and is released with the pool. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    int size = 0;
    char *line = NULL;
    /* List the objects whose names contain the specified prefix. */
    char *prefix = "fun/";
    char *nextMarker = "";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    params->max_ret = 100;
    aos_str_set(&params->prefix, prefix);
    aos_str_set(&params->marker, nextMarker);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    do {
        resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
        if (! aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_pstrprintf(pool, "%.s\t%.s\t%.s\n", content->key.len, content->key.data,
                content->size.len, content->size.data,
                content->last_modified.len, content->last_modified.data);
            printf("%s", line);
        }
        nextMarker = apr_pstrprintf(pool, "%.s", params->next_marker.len, params->next_marker.data);
        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    printf("Total %d\n", size);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Before you close the program, call aos_http_io_deinitialize to release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

The returned results are as follows:

```
Objects:
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
CommonPrefixes:
```

- List the objects and subfolders in a folder

Use the following code to list objects and subfolders in a folder:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. The code used to create a memory pool is included in the APR library. aos_pool_t is equivalent to apr
    _pool_t. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret,
    is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* The memory consumed by options is allocated by the memory pool and is released with the pool. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_status_t *resp_status = NULL;
    oss_list_object_params_t *params = NULL;
    oss_list_object_content_t *content = NULL;
    oss_list_object_common_prefix_t *commonPrefix = NULL;
    int size = 0;
    char *line = NULL;
    /* List the objects whose names contain the specified prefix. */
    char *prefix = "fun/";
    char *nextMarker = "";
    /* Set the delimiter to forward slashes (/). */
    char *delimiter = "/";
    aos_str_set(&bucket, bucket_name);
    params = oss_create_list_object_params(pool);
    params->max_ret = 100;
    aos_str_set(&params->prefix, prefix);
    aos_str_set(&params->marker, nextMarker);
    aos_str_set(&params->delimiter, delimiter);
    printf("Object\tSize\tLastModified\n");
    /* List all objects. */
    do {
        resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
        if (!aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            ++size;
            line = apr_pstrprintf(pool, "%.s\t%.s\t%.s\n", content->key.len, content->key.data,
                content->size.len, content->size.data,
                content->last_modified.len, content->last_modified.data);
            printf("%s", line);
        }
        aos_list_for_each_entry(oss_list_object_common_prefix_t, commonPrefix, &params->common_prefix_list, node) {
            line = apr_pstrprintf(pool, "%.s", commonPrefix->prefix.len, commonPrefix->prefix.data);
            printf("%s", line);
        }
        nextMarker = apr_pstrprintf(pool, "%.s", params->next_marker.len, params->next_marker.data);
```

```

        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    printf("Total %d\n", size);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Before you close the program, call aos_http_io_deinitialize to release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

The returned results are as follows:

```

Objects:
fun/test.jpg
CommonPrefixes:
fun/movie/

```

- List the sizes of objects in a specified folder

The following code provides an example on how to query the sizes of objects in a specified folder:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int64_t calculateFolderLength (aos_string_t bucketName, char* folder)
{
    int64_t size = 0;
    oss_list_object_params_t *params = NULL;
    char *nextMarker = "";
    aos_status_t *resp_status = NULL;
    aos_pool_t *p = NULL;
    oss_request_options_t *options = NULL;
    oss_list_object_content_t* content = NULL;
    aos_pool_create(&p, NULL);
    options = oss_request_options_create(p);
    params = oss_create_list_object_params(p);
    params->max_ret = 10;
    aos_str_set(&params->prefix, folder);
    aos_str_set(&params->marker, nextMarker);
    /* List all objects. */
    do {
        resp_status = oss_list_object(options, &bucketName, params, NULL);
        if (!aos_status_is_ok(resp_status))
        {
            printf("list object failed\n");
            break;
        }
        aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
            size += content->size.len;
        }
        nextMarker = apr_pstrprintf(p, "%.s", params->next_marker.len, params->next_marker.data);
        aos_str_set(&params->marker, nextMarker);
        aos_list_init(&params->object_list);
        aos_list_init(&params->common_prefix_list);
    } while (params->truncated == AOS_TRUE);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
    aos_pool_destroy(p);
    return size;
}
int main(int argc, char *argv[])
{
    /* Call aos_http_io_initialize in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOS_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. The code used to create a memory pool is included in the APR library. aos_pool_t is equivalent to apr_pool_t. */
    aos_pool_t *pool;

```

```

/* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
aos_pool_create(&pool, NULL);
/* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret,
is_cname, and curl. */
oss_request_options_t *oss_client_options;
/* The memory consumed by options is allocated by the memory pool and is released with the pool. */
oss_client_options = oss_request_options_create(pool);
/* Call oss_client_options to initialize the client options. */
init_options(oss_client_options);
/* Initialize the parameters. */
aos_string_t bucket;
aos_status_t *resp_status = NULL;
oss_list_object_params_t *params = NULL;
oss_list_object_content_t *content = NULL;
oss_list_object_common_prefix_t *commonPrefix = NULL;
int size = 0;
char *line = NULL;
/* List the objects whose names contain the specified prefix. */
char *prefix = "fun/";
char *nextMarker = "";
/* Set the delimiter to forward slashes (/). */
char *delimiter = "/";
aos_str_set(&bucket, bucket_name);
params = oss_create_list_object_params(pool);
params->max_ret = 100;
aos_str_set(&params->prefix, prefix);
aos_str_set(&params->marker, nextMarker);
aos_str_set(&params->delimiter, delimiter);
printf("Object\tSize\tLastModified\n");
/* List all objects. */
do {
    resp_status = oss_list_object(oss_client_options, &bucket, params, NULL);
    if (!aos_status_is_ok(resp_status))
    {
        printf("list object failed\n");
        break;
    }
    aos_list_for_each_entry(oss_list_object_content_t, content, &params->object_list, node) {
        ++size;
        line = apr_psprintf(pool, "%s\t%s\t%s\n", content->key.len, content->key.data,
            content->size.len, content->size.data,
            content->last_modified.len, content->last_modified.data);
        printf("%s", line);
    }
    aos_list_for_each_entry(oss_list_object_common_prefix_t, commonPrefix, &params->common_prefix_list, node) {
        line = apr_psprintf(pool, "%s", commonPrefix->prefix.len, commonPrefix->prefix.data);
        int64_t size = calculateFolderLength(bucket, commonPrefix->prefix.data);
        printf("Total %d\n", size);
    }
    nextMarker = apr_psprintf(pool, "%s", params->next_marker.len, params->next_marker.data);
    aos_str_set(&params->marker, nextMarker);
    aos_list_init(&params->object_list);
    aos_list_init(&params->common_prefix_list);
} while (params->truncated == AOS_TRUE);
printf("Total %d\n", size);
/* Release the memory pool after the request is complete. In this case, the memory allocated to various resources used for the request is released. */
aos_pool_destroy(pool);
/* Before you close the program, call aos_http_io_deinitialize to release the allocated global resources. */
aos_http_io_deinitialize();
return 0;
}

```

1.4.4.8.5. Delete objects

This topic describes how to delete objects.

 **Warning** Deleted objects cannot be recovered. Exercise caution when you delete objects.

For the complete code that is used to delete an object, visit [GitHub](#).

Delete a single object

The following code provides an example on how to delete a single object:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    /* Assign char* data to a bucket of the aos_string_t type. */
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Delete the object. */
    resp_status = oss_delete_object(oss_client_options, &bucket, &object, &resp_headers);
    /* Determine whether the object is deleted. */
    if (aos_status_is_ok(resp_status)) {
        printf("delete object succeed\n");
    } else {
        printf("delete object failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete multiple objects

The following code provides an example on how to delete multiple objects:


```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name1 = "<yourObjectName1>";
const char *object_name2 = "<yourObjectName2>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object1;
    aos_string_t object2;
    int is_quiet = 1;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object1, object_name1);
    aos_str_set(&object2, object_name2);
    /* Create a list of objects that you want to delete. */
    aos_list_t object_list;
    aos_list_t deleted_object_list;
    oss_object_key_t *content1;
    oss_object_key_t *content2;
    aos_list_init(&object_list);
    aos_list_init(&deleted_object_list);
    content1 = oss_create_oss_object_key(pool);
    aos_str_set(&content1->key, object_name1);
    aos_list_add_tail(&content1->node, &object_list);
    content2 = oss_create_oss_object_key(pool);
    aos_str_set(&content2->key, object_name2);
    aos_list_add_tail(&content2->node, &object_list);
    /* Delete the objects in the list. 'is_quiet' specifies whether to return the deletion result. */
    resp_status = oss_delete_objects(oss_client_options, &bucket, &object_list, is_quiet, &resp_headers, &deleted_object_list);
    if (aos_status_is_ok(resp_status)) {
        printf("delete objects succeeded\n");
    } else {
        printf("delete objects failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Delete multiple objects whose names contain a specified prefix

You can delete multiple objects whose names contain a specified prefix to delete a folder. For example, if you delete objects whose names contain `dir/`, the `dir` folder is deleted. The following code provides an example on how to delete multiple objects whose names contain a specified prefix:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_prefix = "<yourObjectNamePrefix>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t prefix;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&prefix, object_prefix);
    /* Delete the objects whose names contain the specified prefix. */
    resp_status = oss_delete_objects_by_prefix(oss_client_options, &bucket, &prefix);
    if (aos_status_is_ok(resp_status)) {
        printf("delete objects by prefix succeeded\n");
    } else {
        printf("delete objects by prefix failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

 **Warning** If you set the prefix to an empty string or NULL, all objects in the bucket are deleted. Exercise caution when you use this operation.

1.4.4.8.6. Copy objects

This topic describes how to copy an object.

Limits on object copying:

- You must have read permissions on the source object and write permissions on the destination bucket.
- You cannot copy an object across regions.

Copy a small object

The following code provides an example on how to copy an object smaller than or equal to 1 GB in size:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *source_bucket_name = "<yourSourceBucketName>";
const char *source_object_name = "<yourSourceObjectName>";
const char *dest_bucket_name = "<yourDestBucketName>";
const char *dest_object_name = "<yourDestObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t source_bucket;
    aos_string_t source_object;
    aos_string_t dest_bucket;
    aos_string_t dest_object;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&source_bucket, source_bucket_name);
    aos_str_set(&source_object, source_object_name);
    aos_str_set(&dest_bucket, dest_bucket_name);
    aos_str_set(&dest_object, dest_object_name);
    headers = aos_table_make(pool, 0);
    /* Copy the object. */
    resp_status = oss_copy_object(oss_client_options, &source_bucket, &source_object, &dest_bucket, &dest_object, headers, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("copy object succeeded\n");
    } else {
        printf("copy object failed\n");
    }
    /* Release the memory pool. The memory allocated to various resources used for the request is released. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

Copy a large object

We recommend that you use the `oss_upload_part_copy` operation to copy an object larger than 1 GB in size. The following code provides an example on how to perform multipart copy:

```

#include "oss_api.h"
#include "aos_http_io.h"
#include <sys/stat.h>
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *source_bucket_name = "<yourSourceBucketName>";
const char *source_object_name = "<yourSourceObjectName>";
const char *dest_bucket_name = "<yourDestBucketName>";
const char *dest_object_name = "<yourDestObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{

```

```

options->config = oss_config_create(options->pool);
/* Use a char* string to initialize the aos_string_t data type. */
aos_str_set(&options->config->endpoint, endpoint);
aos_str_set(&options->config->access_key_id, access_key_id);
aos_str_set(&options->config->access_key_secret, access_key_secret);
/* Specify whether to use CNAME to access OSS. A value of 0 indicates that CNAME is not used. */
options->config->is_cname = 0;
/* Configure network parameters such as the timeout periods. */
options->ctl = aos_http_controller_create(options->pool, 0);
}

int64_t get_file_size(const char *file_path)
{
    int64_t filesize = -1;
    struct stat statbuff;
    if(stat(file_path, &statbuff) < 0){
        return filesize;
    } else {
        filesize = statbuff.st_size;
    }
    return filesize;
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate a memory chunk in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Call oss_client_options to initialize the client options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t dest_bucket;
    aos_string_t dest_object;
    aos_string_t upload_id;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    oss_list_upload_part_params_t *list_upload_part_params;
    oss_upload_part_copy_params_t *upload_part_copy_params1;
    oss_upload_part_copy_params_t *upload_part_copy_params2;
    oss_list_part_content_t *part_content;
    aos_list_t complete_part_list;
    oss_complete_part_content_t *complete_content;
    aos_table_t *list_part_resp_headers = NULL;
    aos_table_t *complete_resp_headers = NULL;
    int part1 = 1;
    int part2 = 2;
    int64_t range_start1 = 0;
    int64_t range_end1 = 6000000; //not less than 5MB
    int64_t range_start2 = 6000001;
    int64_t range_end2;
    int max_ret = 1000;
    headers = aos_table_make(pool, 0);
    aos_str_set(&dest_bucket, dest_bucket_name);
    aos_str_set(&dest_object, dest_object_name);
    resp_status = oss_init_multipart_upload(oss_client_options, &dest_bucket, &dest_object, &upload_id, headers, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("init multipart upload succeeded, upload_id is %s\n", upload_id.data);
    } else {
        printf("init multipart upload failed\n");
    }

    /* Copy the first part. */
    upload_part_copy_params1 = oss_create_upload_part_copy_params(pool);
    aos_str_set(&upload_part_copy_params1->source_bucket, source_bucket_name);
    aos_str_set(&upload_part_copy_params1->source_object, source_object_name);
    aos_str_set(&upload_part_copy_params1->dest_bucket, dest_bucket_name);
    aos_str_set(&upload_part_copy_params1->dest_object, dest_object_name);
    aos_str_set(&upload_part_copy_params1->upload_id, upload_id.data);
    upload_part_copy_params1->part_num = part1;
    upload_part_copy_params1->range_start = range_start1;
    upload_part_copy_params1->range_end = range_end1;
    headers = aos_table_make(pool, 0);
    resp_status = oss_upload_part_copy(oss_client_options, upload_part_copy_params1, headers, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("upload part 1 copy succeeded\n");
    } else {
        printf("upload part 1 copy failed\n");
    }
}

```

```

/* Copy the second part. */
range_end2 = get_file_size(local_filename) - 1;
upload_part_copy_params2 = oss_create_upload_part_copy_params(pool);
aos_str_set(&upload_part_copy_params2->source_bucket, source_bucket_name);
aos_str_set(&upload_part_copy_params2->source_object, source_object_name);
aos_str_set(&upload_part_copy_params2->dest_bucket, dest_bucket_name);
aos_str_set(&upload_part_copy_params2->dest_object, dest_object_name);
aos_str_set(&upload_part_copy_params2->upload_id, upload_id.data);
upload_part_copy_params2->part_num = part2;
upload_part_copy_params2->range_start = range_start2;
upload_part_copy_params2->range_end = range_end2;
headers = aos_table_make(pool, 0);
resp_status = oss_upload_part_copy(oss_client_options, upload_part_copy_params2, headers, &resp_headers);
if (aos_status_is_ok(resp_status)) {
    printf("upload part 2 copy succeeded\n");
} else {
    printf("upload part 2 copy failed\n");
}
/* List the copied parts. */
headers = aos_table_make(pool, 0);
list_upload_part_params = oss_create_list_upload_part_params(pool);
list_upload_part_params->max_ret = max_ret;
aos_list_init(&complete_part_list);
resp_status = oss_list_upload_part(oss_client_options, &dest_bucket, &dest_object, &upload_id, list_upload_part_params, &list_part_resp_headers);
aos_list_for_each_entry(oss_list_part_content_t, part_content, &list_upload_part_params->part_list, node) {
    complete_content = oss_create_complete_part_content(pool);
    aos_str_set(&complete_content->part_number, part_content->part_number.data);
    aos_str_set(&complete_content->etag, part_content->etag.data);
    aos_list_add_tail(&complete_content->node, &complete_part_list);
}
/* Complete the multipart copy task. */
resp_status = oss_complete_multipart_upload(oss_client_options, &dest_bucket, &dest_object, &upload_id, &complete_part_list, headers, &complete_resp_headers);
if (aos_status_is_ok(resp_status)) {
    printf("complete multipart upload succeeded\n");
} else {
    printf("complete multipart upload failed\n");
}
/* Release the memory pool. The memory allocated to various resources used for the request is released. */
aos_pool_destroy(pool);
/* Release the allocated global resources. */
aos_http_io_deinitialize();
return 0;
}

```

1.4.4.9. Authorized access

This topic describes how to authorize access to OSS by using STS or a signed URL.

Use STS for temporary access authorization

Alibaba Cloud STS is a web service that provides a temporary access token to a cloud computing user. You can use STS to grant a set of temporary access credentials that have a custom validity period and custom permissions to a third-party application or a RAM user managed by you.

STS provides the following benefits:

- You need only to generate an access token and send the access token to a third-party application. You do not need to expose your AccessKey pair to the third-party application. You can specify the access permissions and validity period of this token.
- The token automatically expires after the validity period. Therefore, you do not need to manually revoke the access permissions of a token.

The following sample code provides an example on how to upload a string to an object by using temporary access credentials obtained from STS.

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *sts_token = "<yourStsToken>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *object_content = "More than just cloud.";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    aos_str_set(&options->config->sts_token, sts_token);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_list_t buffer;
    aos_buf_t *content = NULL;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    /* Assign the char* data to the bucket. */
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_list_init(&buffer);
    content = aos_buf_pack(oss_client_options->pool, object_content, strlen(object_content));
    aos_list_add_tail(&content->nnode, &buffer);
    /* Upload the object. */
    resp_status = oss_put_object_from_buffer(oss_client_options, &bucket, &object, &buffer, headers, &resp_headers);
    /* Determine whether the object is uploaded. */
    if (aos_status_is_ok(resp_status)) {
        printf("put object from buffer succeeded\n");
    } else {
        printf("put object from buffer failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_deinitialize();
    return 0;
}

```

Use a signed URL for temporary access authorization

can generate a signed URL and provide the URL to a visitor for temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of time during which the visitor can access OSS.

- Use a signed URL to upload an object

The following code provides an example on how to use a signed URL to upload an object:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_http_request_t *req;
    apr_time_t now;
    char *url_str;
    aos_string_t url;
    int64_t expire_time;
    int one_hour = 3600;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    expire_time = now / 1000000 + one_hour;
    headers = aos_table_make(pool, 0);
    req = aos_http_request_create(pool);
    req->method = HTTP_PUT;
    now = apr_time_now(); /* Unit: microseconds */
    expire_time = now / 1000000 + one_hour;
    /* Generate a signed URL. */
    url_str = oss_gen_signed_url(oss_client_options, &bucket, &object, expire_time, req);
    aos_str_set(&url, url_str);
    printf("The signed URL used to upload the object: %s\n", url_str);
    /* Use the signed URL to upload the object. */
    resp_status = oss_put_object_from_file_by_url(oss_client_options, &url, &file, headers, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("put objects by signed url succeeded\n");
    } else {
        printf("put objects by signed url failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

- Use a signed URL to download an object

The following code provides an example on how to use a signed URL to download an object:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
const char *local_filename = "<yourLocalFilename>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize the aos_string_t data type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. The value 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as the timeout period. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code that is used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The value of the second parameter is NULL. This value indicates that the pool does not inherit other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *params = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_http_request_t *req;
    apr_time_t now;
    char *url_str;
    aos_string_t url;
    int64_t expire_time;
    int one_hour = 3600;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    aos_str_set(&file, local_filename);
    expire_time = now / 1000000 + one_hour;
    headers = aos_table_make(pool, 0);
    params = aos_table_make(pool, 0);
    req = aos_http_request_create(pool);
    req->method = HTTP_GET;
    now = apr_time_now(); /* Unit: microseconds. */
    expire_time = now / 1000000 + one_hour;
    /* Generate a signed URL. */
    url_str = oss_gen_signed_url(oss_client_options, &bucket, &object, expire_time, req);
    aos_str_set(&url, url_str);
    /* Use the signed URL to download the object. */
    resp_status = oss_get_object_to_file_by_url(oss_client_options, &url, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object succeeded\n");
    } else {
        printf("get object failed\n");
    }
    /* Release the memory pool. This operation releases the memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

1.4.4.10. IMG

OSS Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service that can process large amounts of data. After you upload source images to OSS, you can call RESTful API operations to process the images anytime, anywhere, and on any Internet device.

For the complete IMG code, visit [GitHub](#).

Basic features

OSS IMG provides the following features:

- Query image information.
- Convert image formats.
- Resize, crop, and rotate images.
- Apply image effect.
- Add image, text, and text-and-image watermarks to images.
- Customize image processing styles. Log on to the OSS console. Click a bucket name to go to the bucket details page. Click the **Image Processing (IMG)** tab and click **Create Rule** to customize an image processing style.
- Use cascade processing to call multiple IMG features.

Usage

You can use standard `HTTP GET` requests to access IMG. All processing parameters are encoded in the query string of a URL.

• Anonymous access

You can anonymously access a processed image by using a third-level domain in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction>,<yourParamValue>
```

Parameter	Description
bucket	The name of the bucket.
endpoint	The domain name used to access the region.
object	The name of the image object.
image	The identifier reserved by IMG.
action	The operations performed on images, such as scaling, cropping, and rotating.
param	The parameters corresponding to the operations performed on the images.

◦ Basic operations

For example, you can scale an image to a width of 100 and adjust the height based on the ratio:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

◦ Customize an image style

You can anonymously access IMG by using a third-level domain in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- style: the identifier reserved by IMG for the custom style.
- yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when you create the custom style in the console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

◦ Cascade processing

You can use cascade processing to perform multiple operations in sequence on an image by using a URL in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

◦ Access by using HTTPS

IMG supports access by using HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

• Authorized access

If the ACL of an image object is private, you must have permissions on the image object before you can perform operations on the image object.

Bucket ACL	Object ACL
private	default
Any ACL	private

The following code provides an example on how to generate a signed URL to access IMG:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This value indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t *params = NULL;
    aos_http_request_t *req;
    char *url_str;
    apr_time_t now;
    int64_t expire_time;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Process the image. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/resize,m_fixed,w_100,h_100");
    req = aos_http_request_create(pool);
    req->method = HTTP_GET;
    req->query_params = params;
    /* Specify the validity period (expire_time) in seconds. */
    now = apr_time_now();
    expire_time = now / 1000000 + 10 * 60;
    /* Generate a signed URL. */
    url_str = oss_gen_signed_url(oss_client_options, &bucket, &object, expire_time, req);
    printf("url: %s\n", url_str);
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}

```

- Access by using SDKs

You can use OSS SDKs to access and process image objects of any ACLs.

- Basic operations

The following code provides an example on how to perform basic IMG operations including obtaining image information, converting image formats, resizing, cropping, rotating, watermarking, and applying effects:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);

```

```

aos_str_set(&options->config->access_key_secret, access_key_secret);
/* Specify whether to use CNAME. A value of 0 indicates that CNAME is not used. */
options->config->is_cname = 0;
/* Configure network parameters such as timeout periods. */
options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This value indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *params = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Scale the image. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/resize,m_fixed,w_100,h_100");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-resize.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }

    /* Crop the image. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/crop,w_100,h_100,x_100,y_100,r_1");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-crop.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }

    /* Rotate the image. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/rotate,90");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-rotate.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }

    /* Sharpen the image. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/sharpen,100");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-sharpen.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }

    /* Add watermarks to the image. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/watermark,text_SGVsbG8g5Zu-54mh5pyN5YqhIQ");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-watermark.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {

```

```
        printf("get object to file failed\n");
    }
    /* Convert the image format. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/format,png");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-format.png");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }
    /* Obtain image information. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/info");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-info.txt");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

- Custom image styles

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This value indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *params = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Customize the image style. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "style/<yourCustomStyleName>");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-custom-style.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }

    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

- Cascade processing

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
const char *access_key_secret = "<yourAccessKeySecret>";
const char *bucket_name = "<yourBucketName>";
const char *object_name = "<yourObjectName>";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char* string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Specify whether to use CNAME. A value of 0 indicates that CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters such as timeout periods. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}

int main(int argc, char *argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources such as networks and memory. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }

    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used to create a memory pool is included in the APR library. */
    aos_pool_t *pool;
    /* Create a memory pool. The second parameter is NULL. This value indicates that the pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options;
    /* Allocate the memory resources in the memory pool to the options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize the parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_string_t file;
    aos_table_t *headers = NULL;
    aos_table_t *params = NULL;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    aos_str_set(&object, object_name);
    /* Perform cascade processing. */
    params = aos_table_make(pool, 1);
    apr_table_set(params, OSS_PROCESS, "image/resize,m_fixed,w_100,h_100/rotate,90");
    /* Download the processed image to a local file. */
    aos_str_set(&file, "example-cascade.jpg");
    resp_status = oss_get_object_to_file(oss_client_options, &bucket, &object, headers, params, &file, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object to file succeeded\n");
    } else {
        printf("get object to file failed\n");
    }
    /* Release the memory pool, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

1.4.4.11. Error handling

This topic describes how to handle errors when you use OSS SDK for C.

Exception handling

If an error occurs when you send a request by using OSS SDK for C, an corresponding error message is returned by `aos_status_s`. The following table describes the attributes of `aos_status_s`.

Error code	Description	Type
code	The HTTP status code of the error request.	Integer
error_code	The error code returned by OSS.	String
error_msg	The error message returned by OSS.	String
req_id	The UUID that identifies the request. You can provide req_id to OSS development engineers for help.	String

Timeout handling

If the code in the returned `aos_status_t` parameter is not 2xx and `error_code` indicates -992 or -995, the connection or request times out. You can send the request again.

You can use `aos_should_retry(aos_status_t *)` in the `aos_status.h` file to determine whether the error code returned requires a retry. If a value of 1 is returned, you can try again.

Common error codes

Error code	Error message	HTTP status code
AccessDenied	The error message returned because the access is denied.	403
BucketAlreadyExists	The error message returned because the bucket already exists.	409
BucketNotEmpty	The error message returned because the bucket is not empty.	409
EntityTooLarge	The error message returned because the entity exceeds the maximum allowed size.	400
EntityTooSmall	The error message returned because the entity is smaller than the minimum allowed size.	400
FileGroupTooLarge	The error message returned because the object group exceeds the maximum allowed size.	400
FilePartNotExist	The error message returned because the specified part does not exist.	400
FilePartStale	The error message returned because the part has expired.	400
InvalidArgument	The error message returned because the parameter format is invalid.	400
InvalidAccessKeyId	The error message returned because the specified Accesskey ID does not exist.	403
InvalidBucketName	The error message returned because the bucket name is invalid.	400
InvalidDigest	The error message returned because the digest is invalid.	400
InvalidObjectName	The error message returned because the object name is invalid.	400
InvalidPart	The error message returned because the part is invalid.	400
InvalidPartOrder	The error message returned because the part order is invalid.	400
InvalidTargetBucketForLogging	The error message returned because the target bucket specified for logging is invalid.	400
InternalError	The error message returned because an internal OSS error occurs.	500
MalformedXML	The error message returned because the XML format is invalid.	400
MethodNotAllowed	The error message returned because the method is not supported.	405
MissingArgument	The error message returned because the required parameters are not set.	411
MissingContentLength	The error message returned because the content length is not specified.	411
NoSuchBucket	The error message returned because the specified bucket does not exist.	404
NoSuchKey	The error message returned because the specified object does not exist.	404
NoSuchUpload	The error message returned because the specified multipart upload ID does not exist.	404
NotImplemented	The error message returned because the method cannot be implemented.	501
PreconditionFailed	The error message returned because a precondition error occurs.	412

Error code	Error message	HTTP status code
RequestTimeTooSkewed	The error message returned because the difference between the time on the client when the request is sent and the time on the server when the request is received is greater than 15 minutes.	403
RequestTimeout	The error message returned because the request times out.	400
SignatureDoesNotMatch	The error message returned because the signature is invalid.	403
TooManyBuckets	The error message returned because the number of the buckets managed by a user exceeds the limit.	400

1.4.5. Go-SDK

1.4.5.1. Preface

This topic describes the sample code of OSS SDK for Go in various use cases.

SDK source code and API documents

For the source code of OSS SDK for Go, visit [GitHub](#). For more information, see [OSS SDK for Go API documents](#).

Sample programs

OSS SDK for Go provides a variety of sample programs for your reference or use. The following table describes the sample programs provided.

Sample file	Description
new_bucket.go	Shows you how to initialize a client.
put_object.go	Shows you how to use simple upload and resumable upload.
append_object.go	Shows you how to use append upload.
get_object.go	Shows you how to use streaming download, range download, and resumable download.
delete_object.go	Shows you how to delete a single object and multiple objects.
copy_object.go	Shows you how to use object copy and resumable copy.
list_objects.go	Shows you how to list objects, including list by default or specified parameter.
object_meta.go	Shows you how to configure and query object metadata.
object_acl.go	Shows you how to configure and query object ACL.
cname_sample.go	Shows you how to use CNAME.
create_bucket.go	Shows you how to create a bucket.
list_buckets.go	Shows you how to list buckets, including list by default or specified parameter.
bucket_acl.go	Shows you how to configure and query bucket ACL.
bucket_referer.go	Shows you how to configure, query, and delete the Referer whitelist for a bucket.
bucket_logging.go	Shows you how to configure, query, and delete access logs for a bucket.
bucket_lifecycle.go	Shows you how to configure, query, and delete lifecycle rules for objects in a bucket.
bucket_cors.go	Shows you how to configure, query, and delete the cross-origin resource sharing (CORS) rules for a bucket.

1.4.5.2. Installation

This topic describes how to install OSS SDK for Go.

Prerequisites

- Environment

Go 1.6 or later is required.

For more information about how to download and install the environment for Go, visit [Installing Go from source](#). After Go is installed, create a new system variable GOPATH, and set the value to your code directory. For more information about GOPATH, run the `go help gopath` command.

- Check the Go version

Run the `go version` command to view the version of Go.

Download OSS SDK for Go

- [Download from GitHub](#)
- [Download previous versions](#)

Install OSS SDK for Go

Run the following command to install OSS SDK for Go:

```
go get github.com/aliyun/aliyun-oss-go-sdk/oss
```

 **Note** When you run the go get command to install the SDK, no prompts are displayed. If the installation times out, run the preceding command again.

View the version of OSS SDK for Go

Run the following code to view the version of OSS SDK for Go:

```
package main
import (
    "fmt"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    fmt.Println("OSS Go SDK Version: ", oss.Version)
}
```

1.4.5.3. Initialization

OSSClient serves as the client of OSS SDK for Go to manage OSS resources such as buckets and objects.

To create an OSSClient instance, you must specify an endpoint. For more information about endpoints, see [Obtain an endpoint](#).

To create an OSSClient instance, you must specify an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

The following table describes the parameters you can configure for this operation.

Parameter	Description	Default
UseCName	Specifies whether to use CNAME.	No
Timeout	The timeout period for request, including the connection timeout period and socket read/write timeout period. Unit: seconds.	The connection timeout period is 30 seconds. The read/write timeout period is 60 seconds.
SecurityToken	The temporary security token for users.	None
EnableMD5	Specifies whether to enable MD5 verification. We recommend that you enable CRC, which is more efficient than MD5 verification.	No
EnableCRC	Specifies whether to enable CRC.	Yes
Proxy	The proxy server. Example: <code>http://8.8.8.8:3128</code> .	None
AuthProxy	The proxy server that requires an account and a password to log on.	None

Create an OSSClient instance based on an OSS endpoint

The following code provides an example on how to create an OSSClient instance based on an OSS endpoint:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

Create an OSSClient instance based on a custom domain name

For the complete code of custom domain names, visit [GitHub](#).

The following code provides an example on how to create an OSSClient instance based on a custom domain name:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"
// oss.UseCName(true) indicates that CNAME is enabled. If CNAME is enabled, a custom domain name can be bound to a bucket.
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.UseCName(true))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

 **Notice** When you use CNAME, you cannot list buckets because a custom domain name is bound to a specified bucket.

Create an OSSClient instance based on STS

The following code provides an example on how to create an OSSClient instance based on Security Token Service (STS):

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.SecurityToken("<yourSecurityToken>"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

Configure the connection timeout period

The following code provides an example on how to configure the connection timeout period:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"
// oss.Timeout(10, 120) indicates that the HTTP connection timeout period is 10 seconds and the HTTP read/write timeout period is 120 seconds. The default HTTP connection timeout period is 30 seconds. The default HTTP read/write timeout period is 60 seconds. A value of 0 indicates that the connection never times out.
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.Timeout(10, 120))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

Disable CRC

By default, CRC is enabled to ensure data integrity when you upload or download objects. The following code provides an example on how to disable CRC:

 **Warning** We recommend that you do not disable CRC. If you disable CRC, data integrity may be affected during object upload and download.

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.EnableCRC(false))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

1.4.5.4. Quick start

This topic describes how to use OSS SDK for Go to perform routine operations, such as creating buckets, uploading objects, and downloading objects.

Create buckets

A bucket is a global namespace in OSS. It is similar to a data container that stores files. The following code provides an example on how to create a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func handleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId, accessKeySecret)
    if err != nil {
        handleError(err)
    }
    // Create the bucket.
    err = client.CreateBucket(bucketName)
    if err != nil {
        handleError(err)
    }
}
```

For more information about how to obtain an AccessKey ID and AccessKey secret, see [Obtain an AccessKey pair](#).

For more information about how to obtain an endpoint, see [Obtain an endpoint](#).

Upload objects

The following code provides an example on how to upload objects to OSS by using stream upload:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func handleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // <yourObjectName> indicates the full path of the object you want to upload to OSS, and must include the extension of the object name. Example:
    abc/efg/123.jpg.
    objectName := "<yourObjectName>"
    // <yourLocalFileName> consists of a local file path and a file name with extension. Example: /users/local/myfile.txt.
    localFileName := "<yourLocalFileName>"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId, accessKeySecret)
    if err != nil {
        handleError(err)
    }
    // Query the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        handleError(err)
    }
    // Upload the object.
    err = bucket.PutObjectFromFile(objectName, localFileName)
    if err != nil {
        handleError(err)
    }
}
```

Download objects

The following code provides an example on how to download an object to your local device:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func handleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // <yourObjectName> indicates the full path of the object that you want to download from OSS. <yourObjectName> must include the extension of the
    object name. Example: abc/efg/123.jpg.
    objectName := "<yourObjectName>"
    downloadedFileName := "<yourDownloadedFileName>"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId, accessKeySecret)
    if err != nil {
        handleError(err)
    }
    // Query the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        handleError(err)
    }
    // Download the object.
    err = bucket.GetObjectToFile(objectName, downloadedFileName)
    if err != nil {
        handleError(err)
    }
}
```

List objects

The following code provides an example on how to list objects in a specified bucket. By default, a maximum of 100 objects can be listed.

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    // Query the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // List objects.
    marker := ""
    for {
        lsRes, err := bucket.ListObjects(oss.Marker(marker))
        if err != nil {
            HandleError(err)
        }
        // Display the listed objects. By default, a maximum of 100 objects are returned at a time.
        for _, object := range lsRes.Objects {
            fmt.Println("Bucket: ", object.Key)
        }
        if lsRes.IsTruncated {
            marker = lsRes.NextMarker
        } else {
            break
        }
    }
}

```

Delete objects

The following code provides an example on how to delete an object:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func handleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // <yourObjectName> indicates the full path of the object that you want to delete from OSS. The path must include the extension of the object name. Example: abc/efg/123.jpg.
    objectName := "<yourObjectName>"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId, accessKeySecret)
    if err != nil {
        handleError(err)
    }
    // Query the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        handleError(err)
    }
    // Delete the object.
    err = bucket.DeleteObject(objectName)
    if err != nil {
        handleError(err)
    }
}

```

For more information about how to delete objects, see [Delete objects](#).

1.4.5.5. Buckets

1.4.5.5.1. Create buckets

A bucket is a container used to store objects in Object Storage Service (OSS). Each object is contained in a bucket. This topic describes how to create a bucket.

For the complete code that is used to create a bucket, visit [GitHub](#).

The following code provides an example on how to create a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Create the bucket. By default, the storage class of the bucket is Standard, and the ACL of the bucket is private.
    err = client.CreateBucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.5.2. List buckets

A bucket is a container that is used to store objects in Object Storage Service (OSS). All objects in OSS are contained in buckets. Bucket names are listed in alphabetical order. You can list buckets that belong to the current Alibaba Cloud account in all regions and meet the specific conditions.

For the complete code that is used to list buckets, visit [GitHub](#).

List all buckets

Buckets are listed in alphabetical order. You can list all buckets or the buckets that meet the specified conditions.

The following sample code provides an example on how to list all buckets that belong to an Apsara Stack tenant account:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the buckets.
    marker := ""
    for {
        lsRes, err := client.ListBuckets(oss.Marker(marker))
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        // By default, a maximum of 100 buckets are listed at a time.
        for _, bucket := range lsRes.Buckets {
            fmt.Println("Bucket: ", bucket.Name)
        }
        if lsRes.IsTruncated {
            marker = lsRes.NextMarker
        } else {
            break
        }
    }
}
```

List the buckets whose names contain a specified prefix

The following code provides an example on how to list the buckets whose names contain a specified prefix:

```
// List the buckets whose names contain the specified prefix.
lsRes, err = client.ListBuckets(oss.Prefix("<yourBucketPrefix>"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
// Display the list of buckets.
fmt.Println("Buckets with prefix: ", lsRes.Buckets)
for _, bucket := range lsRes.Buckets {
    fmt.Println("Bucket with prefix: ", bucket.Name)
}
```

List the buckets whose names are alphabetically after the bucket specified by marker

The following code provides an example on how to list the buckets whose names are alphabetically after the bucket specified by the marker parameter:

```
// List the buckets whose names are alphabetically after the bucket specified by marker.
lsRes, err = client.ListBuckets(oss.Marker("<yourBucketMarker>"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
// Display the list of buckets.
fmt.Println("My buckets with marker :", lsRes.Buckets)
for _, bucket := range lsRes.Buckets {
    fmt.Println("Bucket with marker: ", bucket.Name)
}
```

List a specified number of buckets

The following code provides an example on how to list up to 500 buckets:

```
// Set the maximum number of buckets to list to 500. Maximum value: 1000. Default value: 100.
lsRes, err = client.ListBuckets(oss.MaxKeys(500))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
// Display the list of buckets.
fmt.Println("My buckets max num:", lsRes.Buckets)
for _, bucket := range lsRes.Buckets {
    fmt.Println("Bucket with maxKeys: ", bucket.Name)
}
```

1.4.5.5.3. Manage bucket ACLs

A bucket is a container used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. This topic describes how to configure and query the access control list (ACL) of a bucket.

For the complete code used to manage the ACL of a bucket, visit [GitHub](#).

Configure ACL for a bucket

The following table describes the bucket ACLs.

ACL	Description	Value
Private	Only the owner or authorized users of the bucket can read and write the object.	oss.ACLPrivate
Public read	Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object. Exercise caution when you use this ACL.	oss.ACLPublicRead
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	oss.ACLPublicReadWrite

The following code provides an example on how to configure the ACL of a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Set the ACL of the bucket to public read.
    err = client.SetBucketACL("<yourBucketName>", oss.ACLPublicRead)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Query the ACL of a bucket

The following code provides an example on how to query the ACL of a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the ACL of the bucket.
    aclRes, err := client.GetBucketACL("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Bucket ACL:", aclRes.ACL)
}
```

1.4.5.5.4. Configure bucket tagging

You can configure bucket tagging to classify and manage buckets in OSS. For example, you can configure that only buckets with specific tags are listed in when you call the ListBucket operation.

② Note

- Only the bucket owner or authorized Resource Access Management (RAM) users can configure tagging for the bucket. Otherwise, 403 Forbidden is returned with the error code AccessDenied.
- You can configure up to 20 tags for a bucket. A tag is a key-value pair.
- The key of a tag can be up to 64 bytes in length and cannot start with `http://`, `https://`, or `Aliyun`. The key of a tag cannot be empty.
- The tag value is optional. The tag value can be up to 128 bytes in length. You can leave the parameter empty.
- The key and value of a tag must be UTF-8-encoded.
- When you use PutBucketTags to configure a tag for a bucket, the new tag overwrites the existing tag.

Configure tags for a bucket

The following code provides an example on how to configure tags for a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Initialize tagging configurations.
    tag1 := oss.Tag{
        Key:   "key1",
        Value: "value1",
    }
    tag2 := oss.Tag{
        Key:   "key2",
        Value: "value2",
    }
    tagging := oss.Tagging{
        Tags: []oss.Tag{tag1, tag2},
    }
    // Configure tags for the bucket.
    err = client.SetBucketTagging("yourBucketName", tagging)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Query tags of a bucket

The following code provides an example on how to query the tags of a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the tags of the bucket.
    ret, err := client.GetBucketTagging("yourBucketName")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the number of returned tags.
    fmt.Println("Tag length: ", len(ret.Tags))
}
```

List the buckets that have a specified tag

The following code provides an example on how to list buckets that have a specified tag:


```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query buckets by specifying the TagKey parameter.
    ret, err := client.ListBuckets(oss.TagKey("yourTaggingKey"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display information about the buckets.
    for _, bucket := range ret.Buckets{
        fmt.Println("bucket:", bucket)
    }
    // Query buckets by specifying TagKey and TagValue.
    // The TagValue parameter must be configured together with the TagKey parameter and can be unspecified. If you do not specify the TagValue parameter, buckets are listed based on only the TagKey parameter.
    res, err := client.ListBuckets(oss.TagKey("yourTaggingKey"), oss.TagValue("yourTaggingValue"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display information about the buckets.
    for _, b := range res.Buckets{
        fmt.Println("bucket:", b)
    }
}

```

Delete the tags of a bucket

The following code provides an example on how to delete the tags of a bucket:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Delete the tags of the object.
    err = client.DeleteBucketTagging("yourBucketName")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

1.4.5.5.5. Delete buckets

A bucket is a container for objects stored in OSS. Every object is contained in a bucket. This topic describes how to delete a bucket.

Before you delete a bucket, ensure that all objects, LiveChannels, and parts that are generated from multipart upload in the bucket are deleted. The following code provides an example on how to delete a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Delete the bucket.
    err = client.DeleteBucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.5.6. Manage lifecycle rules

Object Storage Service (OSS) allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to Infrequent Access (IA) or Archive to reduce your storage costs. This topic describes how to manage lifecycle rules for buckets.

Background

Each lifecycle rule contains the following information:

- Policy: specifies the mode to match objects and parts.
 - Match by prefix: matches objects and parts by prefix. You can create multiple rules to configure different prefixes. Each prefix must be unique.
 - Match by bucket: matches all objects and parts contained in the bucket. You can create only one rule if you select this method.
- Object lifecycle policy: specifies the validity period or expiration date and the operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired objects are deleted.
 - Expiration date: specifies a date. Objects that are last modified before this date expire and are deleted.
- Part lifecycle policy: specifies the validity period or expiration time and the delete operation to perform on these objects after they expire.
 - Validity period: specifies the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted.
 - Expiration date: specifies a date. Parts that are last modified before this date expire and are deleted.

Lifecycle rules also apply to the parts uploaded by using `uploadPart`. In this case, the last modified time of an object is the time the multipart upload task is initiated.

Configure lifecycle rules

The following code provides an example on how to configure lifecycle rules:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // rule1 (id:"rule1", enable:true, prefix:"foo/", expiry:Days 3) indicates that objects whose names contain the prefix foo/ expire three days after they are last modified.
    rule1 := oss.BuildLifecycleRuleByDays("rule1", "foo/", true, 3)
    rules := []oss.LifecycleRule{rule1}
    // Configure the lifecycle rules.
    bucketName := "<yourBucketName>"
    err = client.SetBucketLifecycle(bucketName, rules)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Query lifecycle rules

The following code provides an example on how to query lifecycle rules of a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Query the lifecycle rules configured for the bucket.
    lcRes, err := client.GetBucketLifecycle(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Lifecycle Rules:", lcRes.Rules)
}
```

Delete lifecycle rules

The following code provides an example on how to delete lifecycle rules from a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Delete the lifecycle rules.
    err = client.DeleteBucketLifecycle(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.5.7. Hotlink protection

OSS provides hotlink protection to prevent unauthorized domain names from accessing your OSS data. You can configure the Referer field in the HTTP header.

To configure hotlink protection, you must configure the following parameters:

- **Referer Whitelist:** Only specified domain names are allowed to access OSS resources.
- **Allow Empty Referer:** If you do not allow empty Refer fields, a request is allowed to access OSS resources only if the request includes the Referer field configured in the HTTP or HTTPS header.

Configure hotlink protection

Run the following code to configure the referer whitelist:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Add a Referer whitelist. Specify that an empty Referer field is not allowed. You can use an asterisk (*) or a question mark (?) as a wildcard
    // to set the Referer parameter.
    referers := []string{"http://www.aliyun.com",
        "http://www.????.aliyuncs.com",
        "http://www. *.com"}
    err = client.SetBucketReferer(bucketName, referers, false)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

Query hotlink protection configurations

Run the following code to obtain a referer whitelist:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Query hotlink protection information.
    refRes, err := client.GetBucketReferer(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Referers: ", refRes.RefererList)
    fmt.Println("AllowEmptyReferer: ", refRes.AllowEmptyReferer)
}

```

Delete hotlink protection configurations

Run the following code to clear a referer whitelist:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Delete hotlink protection configurations. // You cannot delete hotlink protection configurations directly. To delete hotlink protection configurations, you must create a rule that allows an empty Referer field to replace the existing rule.
    err = client.SetBucketReferer(bucketName, []string{}, true)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

1.4.5.5.8. Configure logging

You can enable logging to record access to a bucket in log objects. The log objects are stored in a specified bucket.

The log file format is as follows:

```
<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString
```

For the complete code of logging, visit [GitHub](#).

Enable logging

Run the following code to enable bucket access logging:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    targetBucketName := "<yourTargetBucketName>"
    targetPrefix := "<yourTargetPrefix>"
    // Enable logging for the bucket. targetBucketName specifies the bucket that stores log objects. targetPrefix specifies the prefix of the stored
    log objects. The log objects are stored in the targetPrefix folder.
    err = client.SetBucketLogging(bucketName, targetBucketName, targetPrefix, true)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

View logging configurations

Run the following code to view the access logging configurations for a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // View logging configurations of the bucket.
    logRes, err := client.GetBucketLogging(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Target Bucket: ", logRes.LoggingEnabled.TargetBucket)
    fmt.Println("Target Prefix: ", logRes.LoggingEnabled.TargetPrefix)
}
```

Disable logging

Run the following code to disable access logging for a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    // client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Disable logging.
    err = client.DeleteBucketLogging(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.5.9. Static website hosting

You can set the static website hosting mode for your bucket. After the configurations take effect, you can use the bucket domain to access this static website and be redirected to a specified index page or error page.

Configure static website hosting

Run the following code to configure static website hosting:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Configure static web hosting. Set the index page to index.html and the error page to error.html.
    err = client.SetBucketWebsite(bucketName, "index.html", "error.html")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // View the static website hosting configurations.
    wsRes, err := client.GetBucketWebsite(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("indexWebsite: ", wsRes.IndexDocument.Suffix)
    fmt.Println("errorWebsite: ", wsRes.ErrorDocument.Key)
}
```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Delete the static website hosting configurations.
    err = client.DeleteBucketWebsite(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.5.10. CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to different regions. OSS provides CORS operations to facilitate cross-origin access control.

For the complete CORS rule code, visit [GitHub](#).

Configure CORS rules

Run the following code to configure CORS rules for the specified bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    rule1 := oss.CORSRule{
        AllowedOrigin: []string{"*"},
        AllowedMethod: []string{"PUT", "GET"},
        AllowedHeader: []string{},
        ExposeHeader:  []string{},
        MaxAgeSeconds: 200,
    }
    rule2 := oss.CORSRule{
        AllowedOrigin: []string{"http://www.a.com", "http://www.b.com"},
        AllowedMethod: []string{"POST"},
        AllowedHeader: []string{"Authorization"},
        ExposeHeader:  []string{"x-oss-test", "x-oss-test1"},
        MaxAgeSeconds: 100,
    }
    // Configure the CORS rules.
    err = client.SetBucketCORS(bucketName, []oss.CORSRule{rule1, rule2})
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Query CORS rules

Run the following code to obtain CORS rules:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Query the CORS rules.
    corsRes, err := client.GetBucketCORS(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Bucket CORS:", corsRes.CORSRules)
}
```

Delete CORS rules

Run the following code to delete all CORS rules for the specified bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Delete the CORS rules.
    err = client.DeleteBucketCORS(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.6. Upload objects

1.4.5.6.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for PHP provides the following object upload methods:

You can upload objects to OSS by using one of the following methods:

- **Simple upload:** uploads an object up to 5 GB in size.
- **Append upload:** uploads an object up to 5 GB in size.
- **Resumable upload:** supports concurrent upload and resumable upload. You can define the size of each part. This method is suitable for the upload of large objects. You can use this method to upload an object up to 48.8 TB in size.
- **Multipart upload:** uploads an object up to 48.8 TB in size. This method is suitable for the upload of large objects.

During object upload, you can configure object metadata and view upload progress by using progress bars. For more information, see [Use a progress bar](#).

1.4.5.6.2. Simple upload

This topic describes how to use simple upload.

For the complete code that is used to perform simple upload, visit [GitHub](#).

Upload a string

The following code provides an example on how to upload a string to an OSS object:


```
package main
import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Set the storage class to Standard. The default storage class is Standard.
    storageType := oss.ObjectStorageClass(oss.StorageStandard)
    // Set the object ACL to public read. By default, the object inherits the bucket ACL.
    objectAcl := oss.ObjectACL(oss.ACLPublicRead)
    // Upload the string.
    err = bucket.PutObject("<yourObjectName>", strings.NewReader("yourObjectValue"), storageType, objectAcl)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Upload a byte array

The following code provides an example on how to upload a byte array:

```
package main
import (
    "fmt"
    "os"
    "bytes"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Upload the byte array.
    err = bucket.PutObject("<yourObjectName>", bytes.NewReader([]byte("yourObjectValueByteArray")))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Upload a file stream

The following code provides an example on how to upload a file stream:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Read the on-premises file.
    fd, err := os.Open("<yourLocalFile>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer fd.Close()
    // Upload the file stream.
    err = bucket.PutObject("<yourObjectName>", fd)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

Upload an on-premises file

The following code provides an example on how to upload an on-premises file to OSS:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Upload an on-premises file.
    err = bucket.PutObjectFromFile("<yourObjectName>", "<yourLocalFile>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

1.4.5.6.3. Append upload

This topic describes how to perform append upload.

You cannot perform the copyObject operation on appendable objects. If the object to append does not exist, you can call `AppendObject` to create an appendable object. If the object to append exists, you can call `AppendObject` to append content to the end of the object.

The following code provides an example on how to perform append upload:

```

package main
import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    var nextPos int64 = 0
    // The position for the first append upload is 0, and the position for the next append upload is included in the response. The position from which the next append upload starts is the current length of the object.
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectAppendValue1"), nextPos)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Perform the second append operation.
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectAppendValue2"), nextPos)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // You can append content to an object multiple times.
}

```

You can specify object metadata when you perform the first append operation from the position of 0. Object metadata cannot be specified during subsequent append operations.

```

// Specify the metadata of a new object when you perform the first append operation.
nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectValue"), 0, oss.Meta("MyProp", "MyPropVal"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

```

1.4.5.6.4. Resumable upload

If the system stops responding during multipart upload, you can resume the upload by using the `ListMultipartUploads` and `ListParts` operations to retrieve all multipart upload tasks on an object and list the uploaded parts in each task. This method allows an upload task to be resumed from the last uploaded part.

The progress of the current upload is recorded in the checkpoint file during the upload process. If the upload of a part fails during the process, the next upload attempt starts from the recorded position in the checkpoint file. The checkpoint file is deleted after the resumable upload task is completed.

② Note

- OSS SDK for Go records the upload progress in the checkpoint file. Make sure that you have write permissions on the checkpoint file. The checkpoint file contains a checksum. This file cannot be edited. If the checkpoint file is damaged, you must upload all parts again.
- If the local file is modified during the upload, you must upload all parts of the object again.

You can use `Bucket.UploadFile` to perform resumable upload. You can configure the following parameters.

Parameter	Description
<code>objectKey</code>	The name of the object that is uploaded to OSS.
<code>filePath</code>	The path of the local file.
<code>partSize</code>	The size of each part. The part size ranges from 100 KB to 5 GB.
<code>options</code>	Optional parameters: • <code>Routines</code>: specifies the number of concurrent upload threads. The default value is 1, which indicates that only one thread is used to upload the parts. • <code>Checkpoint</code>: specifies whether to enable resumable upload and the path of the checkpoint file. By default, resumable upload is disabled. For example, <code>oss.Checkpoint(true, "")</code> indicates that resumable upload is enabled and the checkpoint file is the <code>file.cp</code> file stored in the same directory as the local file. <code>file</code> is a variable that indicates the local file name. You can also use <code>oss.Checkpoint(true, "your-cp-file.cp")</code> to specify the checkpoint file. • For other object metadata, see Manage object metadata.

The following code provides an example on how to perform resumable upload:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Obtain the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // The part size is 100 KB. Start resumable upload. Upload the parts by using three concurrent threads.
    // <yourObjectName> is equivalent to objectKey and indicates the full path of the object that you want to upload to OSS. <yourObjectName> must include the file extension of the object. Example: abc/efg/123.jpg.
    // LocalFile is equivalent to filePath. 100*1024 is the partSize value.
    err = bucket.UploadFile("<yourObjectName>", "LocalFile", 100*1024, oss.Routines(3), oss.Checkpoint(true, ""))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.6.5. Multipart upload

OSS allows you to use multipart upload to split an object into several parts and upload them separately. After all parts are uploaded, you can combine the uploaded parts into a complete object to complete the upload.

To implement multipart upload, perform the following operations:

The following code provides an example on how to perform multipart upload:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localFilename := "<yourLocalFilename>"
    // Specify the name of the bucket to which to upload the object.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    chunks, err := oss.SplitFileByPartNum(localFilename, 3)
    fd, err := os.Open(localFilename)
    defer fd.Close()
    // Set the storage class to Standard. The default storage class is Standard.
    storageType := oss.ObjectStorageClass(oss.StorageStandard)
    // Set the storage class to Archive.
    // storageType := oss.ObjectStorageClass(oss.StorageArchive)
    // Step 1: Initiate a multipart upload task. Set the storage class to Standard.
    imur, err := bucket.InitiateMultipartUpload(objectName, storageType)
    // Step 2: Upload the parts.
    var parts []oss.UploadPart
    for _, chunk := range chunks {
        fd.Seek(chunk.Offset, os.SEEK_SET)
        // Call the UploadPart method to upload each part.
        part, err := bucket.UploadPart(imur, fd, chunk.Size, chunk.Number)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        parts = append(parts, part)
    }
    // Set the object ACL to public read. By default, the object inherits the bucket ACL.
    objectAcl := oss.ObjectACL(oss.ACLPublicRead)
    // Step 3: Complete the multipart upload task. Set the object ACL to public read.
    cmur, err := bucket.CompleteMultipartUpload(imur, parts, objectAcl)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("cmur:", cmur)
}

```

Cancel a multipart upload task

You can call the `bucket.AbortMultipartUpload` method to cancel a multipart upload task. If you cancel a multipart upload task, you cannot use the upload ID to upload any parts. The uploaded parts are deleted.

The following code provides an example on how to cancel a multipart upload task:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket to which to upload the object.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Initiate a multipart upload task.
    imur, err := bucket.InitiateMultipartUpload("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Cancel the multipart upload task.
    err = bucket.AbortMultipartUpload(imur)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

List uploaded parts

The following code provides an example on how to list the parts that are uploaded during a specified multipart upload task:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localFilename := "<yourLocalFilename>"
    uploadID := "<yourUploadID>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Split the local file. In this example, the local file is split into three parts.
    chunks, err := oss.SplitFileByPartNum(localFilename, 3)
    fd, err := os.Open(localFilename)
    defer fd.Close()
    // Initiate a multipart upload task.
    imur, err := bucket.InitiateMultipartUpload(objectName)
    uploadID = imur.UploadID
    fmt.Println("InitiateMultipartUpload UploadID: ", uploadID)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Upload the parts.
    var parts []oss.UploadPart
    for _, chunk := range chunks {
        fd.Seek(chunk.Offset, os.SEEK_SET)
        // Call the UploadPart method to upload each part.
        part, err := bucket.UploadPart(imur, fd, chunk.Size, chunk.Number)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        fmt.Println("UploadPartNumber: ", part.PartNumber, ", ETag: ", part.ETag)
        parts = append(parts, part)
    }

    // List the uploaded parts based on the result obtained by using the InitiateMultipartUploadResult operation.
    lsRes, err := bucket.ListUploadedParts(imur)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Display the uploaded parts.
    fmt.Println("\nParts:", lsRes.UploadedParts)
    for _, upload := range lsRes.UploadedParts {
        fmt.Println("List PartNumber: ", upload.PartNumber, ", ETag: ", upload.ETag, ", LastModified: ", upload.LastModified)
    }

    // List all the uploaded parts based on the result that is obtained by using the InitiateMultipartUploadResult operation. The result is generated
    based on the specified object name and upload ID. In this case, the object name and upload ID are known.
    var imur_with_uploadid oss.InitiateMultipartUploadResult
    imur_with_uploadid.Key = objectName
    imur_with_uploadid.UploadID = uploadID
    // List all the uploaded parts.
    lsRes, err = bucket.ListUploadedParts(imur_with_uploadid)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Display the uploaded parts.
    fmt.Println("\nListUploadedParts by UploadID: ", uploadID)
    for _, upload := range lsRes.UploadedParts {
        fmt.Println("List PartNumber: ", upload.PartNumber, ", ETag: ", upload.ETag, ", LastModified: ", upload.LastModified)
    }

    // Complete the multipart upload task.
    cmur, err := bucket.CompleteMultipartUpload(imur, parts)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    fmt.Println("cmur:", cmur)
}

```

List multipart upload tasks

You can call the `Bucket.ListMultipartUploads` method to list all ongoing multipart upload tasks that are initiated but not completed or that are canceled. The following table describes the parameters you can configure.

Parameter	Description
Delimiter	The character used to group objects by name. Objects whose names contain the same string from the prefix and the next occurrence of the delimiter are grouped as a single result element in <code>CommonPrefixes</code> .
MaxUploads	The maximum number of multipart upload tasks to list this time. The default value is 1000. The maximum value is 1000.
KeyMarker	Specifies the key-marker value from which to start the list. Multipart upload tasks are listed in alphabetical order of their object names. You can use this parameter with the <code>UploadIDMarker</code> parameter to specify the start position to list the returned results.
Prefix	Specifies the prefix that returned object names must contain. Note that if you use a prefix for a query, the returned object names contain the prefix.
UploadIDMarker	The start position from which to list the returned results. This parameter is used together with the <code>KeyMarker</code> parameter. - This parameter is ignored if the <code>KeyMarker</code> parameter is not set. - If the <code>KeyMarker</code> parameter is set, the response includes the following tasks: - Specifies the key-marker value from which to start the list. Multipart upload tasks are listed in alphabetical order of their object names. - All multipart upload tasks in which the object names come after the <code>KeyMarker</code> value and the upload IDs are greater than the <code>UploadIDMarker</code> value.

- List multipart upload tasks by using default parameter values

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket to which to upload the object.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List all multipart upload tasks.
    keyMarker := ""
    uploadIdMarker := ""
    for {
        // By default, up to 1,000 records are listed each time.
        lsRes, err := bucket.ListMultipartUploads(oss.KeyMarker(keyMarker), oss.UploadIDMarker(uploadIdMarker))
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        // Display all multipart upload tasks.
        for _, upload := range lsRes.Uploads {
            fmt.Println("Upload: ", upload.Key, ", UploadID: ", upload.UploadID)
        }
        if lsRes.IsTruncated {
            keyMarker = lsRes.NextKeyMarker
            uploadIdMarker = lsRes.NextUploadIDMarker
        } else {
            break
        }
    }
}
```

- List multipart upload tasks by specifying a prefix

```
lsRes, err := bucket.ListMultipartUploads(oss.Prefix("<yourObjectNamePrefix>"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
fmt.Println("Uploads:", lsRes.Uploads)
```

- List up to 100 multipart upload tasks


```
lsRes, err := bucket.ListMultipartUploads(oss.MaxUploads(100))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
fmt.Println("Uploads:", lsRes.Uploads)
```

- List multipart upload tasks by specifying a prefix and the maximum number of records to be returned

```
lsRes, err := bucket.ListMultipartUploads(oss.Prefix("<yourObjectNamePrefix>"), oss.MaxUploads(100))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
fmt.Println("Uploads:", lsRes.Uploads)
```

1.4.5.6.6. Use a progress bar

Progress bars are used to indicate the progress of an upload or download.

The following code provides an example on how to use a progress bar when you call `Bucket.PutObjectFromFile` to upload an object:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
// Define a progress bar listener.
type OssProgressListener struct {
}
// Define the progress event handler.
func (listener *OssProgressListener) ProgressChanged(event *oss.ProgressEvent) {
    switch event.EventType {
    case oss.TransferStartedEvent:
        fmt.Printf("Transfer Started, ConsumedBytes: %d, TotalBytes %d.\n",
            event.ConsumedBytes, event.TotalBytes)
    case oss.TransferDataEvent:
        fmt.Printf("\rTransfer Data, ConsumedBytes: %d, TotalBytes %d, %d%%.",
            event.ConsumedBytes, event.TotalBytes, event.ConsumedBytes*100/event.TotalBytes)
    case oss.TransferCompletedEvent:
        fmt.Printf("\nTransfer Completed, ConsumedBytes: %d, TotalBytes %d.\n",
            event.ConsumedBytes, event.TotalBytes)
    case oss.TransferFailedEvent:
        fmt.Printf("\nTransfer Failed, ConsumedBytes: %d, TotalBytes %d.\n",
            event.ConsumedBytes, event.TotalBytes)
    default:
    }
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localFile := "<yourLocalFile>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // View the upload progress information while you upload an object.
    err = bucket.PutObjectFromFile(objectName, localFile, oss.Progress(&OssProgressListener{}))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.7. Download objects

1.4.5.7.1. Overview

Objects are the basic unit for data operations in OSS. OSS SDK for Go provides a variety of object download methods.

For the complete code that is used to download objects, visit [GitHub](#).

You can use one of the following methods to download objects:

- [Streaming download](#)

- [Range download](#)
- [Download to local files](#)
- [Resumable download](#)
- [Conditional download](#)

You can view the download progress by using progress bars. For more information, see [Use a progress bar](#).

1.4.5.7.2. Streaming download

This topic describes how to use streaming download.

Download an object to a stream

The following code provides an example on how to download an object to a stream:

```
package main
import (
    "fmt"
    "os"
    "io/ioutil"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Download the object to a stream.
    body, err := bucket.GetObject("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // If you do not close the connection after it is used, connection leaks may occur. Consequently, no connections are available and the program cannot run properly.
    defer body.Close()
    data, err := ioutil.ReadAll(body)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("data:", string(data))
}
```

Download an object to the buffer

The following code provides an example on how to download an object to the buffer:

```

package main
import (
    "fmt"
    "os"
    "io"
    "bytes"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Download the object to the buffer.
    body, err := bucket.GetObject("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer body.Close()
    buf := new(bytes.Buffer)
    io.Copy(buf, body)
    fmt.Println("buf:", buf)
}

```

Download an object to a local file stream

The following code provides an example on how to download a specified object to a local file stream:

```

package main
import (
    "fmt"
    "os"
    "io"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Download the object to a local file stream.
    body, err := bucket.GetObject("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer body.Close()
    fd, err := os.OpenFile("LocalFile", os.O_WRONLY|os.O_CREATE, 0660)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer fd.Close()
    io.Copy(fd, body)
}

```

1.4.5.7.3. Range download

If you need only part of the data in an object, you can use range download to download data within the specified range.

Specify a valid range

The following code provides an example on how to specify a valid range to download data:

```

package main
import (
    "fmt"
    "os"
    "io/ioutil"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the data that is within the range from byte 15 to byte 35, which includes a total of 21 bytes.
    // If the specified range is invalid, for example, the specified range includes a negative number, or the specified value is larger than the object size, the entire object is downloaded.
    body, err := bucket.GetObject("<yourObjectName>", oss.Range(15, 35))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // If you do not close the connection after it is used, connection leaks may occur. Consequently, no connections are available and the program cannot run properly.
    defer body.Close()
    data, err := ioutil.ReadAll(body)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("data:", string(data))
}

```

Specify an invalid range

The valid range for an object whose size is 1,000 bytes is from byte 0 to byte 999. If the specified range is not within the range, the range does not take effect. Then, OSS returns the 200 HTTP status code and the data of the entire object. The following list provides examples of invalid requests and the returned results:

- If you set Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns the 200 HTTP status code and the data of the entire object.
- If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns the 200 HTTP status code and the data of the entire object.

Compatible download

If you add x-oss-range-behavior:standard to the request header, the download behavior is modified when the specified range is not within the valid range. Example: Use range download to download an object whose size is 1,000 bytes.

- If you set Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns HTTP status code 206 and the data that is within the range from byte 500 to byte 999.
- If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns HTTP status code 416 and error code InvalidRange.

The following code provides an example on how to specify standard behaviors to download data by range:

```

package main
import (
    "fmt"
    "io/ioutil"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Upload 1,000 bytes.
    strContent := ""
    for i := 0; i < 100; i++ {
        strContent += "abcdefghij"
    }
    fmt.Printf("content len:%d\n", len(strContent))
    // Upload a string.
    err = bucket.PutObject("<yourObjectName>", strings.NewReader(strContent))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // If you set Range: bytes to 500-2000, the value at the end of the range is invalid. OSS returns the 206 HTTP status code and the data that is w
    ithin the range from byte 500 to byte 999.
    // If you set Range: bytes to 1000-2000, the value at the start of the range is invalid. OSS returns the 416 HTTP status code and the error code
    InvalidRange.
    body, err := bucket.GetObject("<yourObjectName>", oss.Range(500, 2000), oss.RangeBehavior("standard"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // If you do not close the connection after it is used, connection leaks may occur. Consequently, no connections are available and the program ca
    nnot run properly.
    defer body.Close()
    data, err := ioutil.ReadAll(body)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    if len(data) != 500 {
        fmt.Println("read data error, len:%d", len(data))
        os.Exit(-1)
    }
    fmt.Println("data:", string(data))
}

```

1.4.5.7.4. Download to local files

This topic describes how to download an object to a local file.

The following code provides an example on how to download a specified object to a local file:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Download the object to the local file.
    err = bucket.GetObjectToFile("<yourObjectName>", "LocalFile")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.7.5. Resumable download

You may fail to download a large object if the network is unstable or if the program stops responding. In some cases, you may still fail to download the object even after multiple attempts. To solve this problem, OSS provides the resumable download feature.

To enable resumable download, you need to split the object you want to download into multiple parts and download them separately. After you have downloaded all parts, these parts will be combined into a complete object.

During the download, the download progress is recorded in a checkpoint file. If the download of a part fails, the next download starts from the position recorded in the checkpoint file. The checkpoint file is deleted after resumable download is complete.

 **Note**

- The SDK records the download progress in the checkpoint file. Ensure that you have write permissions on the checkpoint file. The checkpoint file contains a checksum. This file cannot be edited. If the checkpoint file is damaged, you must download the entire object again.
- If the object being downloaded is modified (the ETag value of the object changes) or some parts are missing or modified, you must download the entire object again.

You can use `Bucket.DownloadFile` to download an object by using resumable download. The following table describes the parameters of this operation.

Parameter	Description
objectKey	The name of the object to download.
filePath	The local file path to which to download the object.
partSize	The size of each part. The part size ranges from 1 byte to 5 GB.
options	The optional parameters, including: • Routines: the number of concurrent download threads. The default value is 1, indicating that only one thread is used to download parts. • Checkpoint: specifies whether resumable download is enabled and the path of the checkpoint file. By default, resumable download is disabled. For example, <code>oss.Checkpoint(true, "")</code> indicates that resumable download is enabled and the checkpoint file is the <code>file.cp</code> file contained in the same directory as the destination local file. <code>file</code> is a variable that indicates the local file name. You can also use <code>oss.Checkpoint(true, "your-cp-file.cp")</code> to specify the checkpoint file. • Conditions: conditions for multipart download. For more information about the conditions, see Manage object metadata.

Run the following code for resumable download:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Start resumable download. Download the parts by using three threads concurrently.
    // yourObjectName is equivalent to objectKey and indicates the complete path of the object that you want to download from OSS. The path must include the extension of the object name. Example: abc/efg/123.jpg.
    // LocalFile is the filePath value. 100*1024 is the partSize value.
    err = bucket.DownloadFile("<yourObjectName>", "LocalFile", 100*1024, oss.Routines(3), oss.Checkpoint(true, ""))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

1.4.5.7.6. Conditional download

OSS allows you to configure one or more conditions for object downloads. If the specified conditions are met, the object can be downloaded. If the specified conditions are not met, an error is returned and the object cannot be downloaded.

The following table describes the download conditions.

Parameter	Description	Configuration method
IfModifiedSince	If the specified time is earlier than the time the object is modified, the object can be downloaded. Otherwise, error code 304 Not modified is returned.	oss.IfModifiedSince
IfUnmodifiedSince	If the specified time is later than or equal to the time the object is modified, the object can be downloaded. Otherwise, error code 412 Precondition failed is returned.	oss.IfUnmodifiedSince
IfMatch	If the specified ETag matches that of an object, the object can be downloaded. Otherwise, error code 412 Precondition failed is returned.	oss.IfMatch
IfNoneMatch	If the specified ETag does not match that of an object, the object can be downloaded. Otherwise, error code 304 Not modified is returned.	oss.IfNoneMatch

You can use If-Modified-Since and If-Unmodified-Since as object download conditions at the same time. You can use If-Match and If-None-Match as object download conditions at the same time. To obtain the ETag value of an object, use the Bucket.GetObjectDetailedMeta method.

You can use Bucket.GetObjectDetailedMeta to Obtain the ETag.

```
package main
import (
    "fmt"
    "os"
    "time"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Configure the download conditions.
    date := time.Date(2015, time.November, 10, 23, 0, 0, 0, time.UTC)
    // The object is not downloaded if it does not meet the specified conditions.
    err = bucket.GetObjectToFile("<yourObjectName>", "LocalFile", oss.IfModifiedSince(date))
    if err == nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // The object is downloaded if it meets the specified conditions.
    err = bucket.GetObjectToFile("<yourObjectName>", "LocalFile", oss.IfUnmodifiedSince(date))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

1.4.5.7.7. Use a progress bar

Progress bars are used to indicate the progress of an upload or download.

The following code provides an example on how to use a progress bar when you use `Bucket.GetObjectToFile` to download an object:


```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

// Define the progress bar listener.
type OssProgressListener struct {
}

// Define the progress event handler.
func (listener *OssProgressListener) ProgressChanged(event *oss.ProgressEvent) {
    switch event.EventType {
    case oss.TransferStartedEvent:
        fmt.Printf("Transfer Started, ConsumedBytes: %d, TotalBytes %d.\n",
            event.ConsumedBytes, event.TotalBytes)
    case oss.TransferDataEvent:
        fmt.Printf("\rTransfer Data, ConsumedBytes: %d, TotalBytes %d, %d%%.",
            event.ConsumedBytes, event.TotalBytes, event.ConsumedBytes*100/event.TotalBytes)
    case oss.TransferCompletedEvent:
        fmt.Printf("\nTransfer Completed, ConsumedBytes: %d, TotalBytes %d.\n",
            event.ConsumedBytes, event.TotalBytes)
    case oss.TransferFailedEvent:
        fmt.Printf("\nTransfer Failed, ConsumedBytes: %d, TotalBytes %d.\n",
            event.ConsumedBytes, event.TotalBytes)
    default:
    }
}

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localFile := "<yourLocalFile>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // View the download progress while you download an object.
    err = bucket.GetObjectToFile(objectName, localFile, oss.Progress(&OssProgressListener{}))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Transfer Completed.")
}

```

1.4.5.8. Manage objects

1.4.5.8.1. Overview

You can manage objects in a bucket by using a variety of operations including SetObjectAcl, GetObjectAcl, ListObjects, DeleteObject, and CopyObject.

By using the preceding operations, you can perform the following operations:

- [Determine whether objects exist](#)
- [Manage object ACLs](#)
- [Manage object metadata](#)
- [List objects](#)
- [Delete objects](#)
- [Copy objects](#)

1.4.5.8.2. Determine whether objects exist

This topic describes how to determine whether an object exists.

Run the following code to determine whether a specified object exists:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Determine whether the object exists.
    isExist, err := bucket.IsObjectExist("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Exist:", isExist)
}
```

1.4.5.8.3. Manage object ACLs

This topic describes how to manage the access control list (ACL) of an object.

For the complete code used to manage the ACL of an object, visit [GitHub](#).

The following table describes the ACLs that can be configured for objects.

ACL	Description	Value
Inherited from bucket	The ACL of an object is the same as that of the bucket in which the object is stored.	oss.ACLDefault
Private	Only the owner or authorized users of the bucket can read and write the object.	oss.ACLPrivate
Public read	Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object. Exercise caution when you use this ACL.	oss.ACLPublicRead
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	oss.PublicReadWrite

The ACL privileges of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

The following code provides a complete example on how to configure and query the ACL of an object:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Configure ACL for the object.
    err = bucket.SetObjectACL("<yourObjectName>", oss.ACLPublicReadWrite)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the ACL of the object.
    aclRes, err := bucket.GetObjectACL("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object ACL:", aclRes.ACL)
}

```

1.4.5.8.4. Manage object metadata

This topic describes how to use OSS SDK for Go to configure, modify and query object metadata.

For the complete code used to manage object metadata, visit [GitHub](#).

Configure object metadata

You can configure the metadata of an object when you upload the object. The following table describes the metadata that you can configure.

Parameter	Description
CacheControl	Specifies the caching behavior of a web page when the object is downloaded.
ContentDisposition	Specifies the name of the object when it is downloaded.
ContentEncoding	Specifies the encoding format that is used when the object is downloaded.
Expires	Specifies the expiration time of the cache in GMT.
ServerSideEncryption	Specifies the server-side encryption algorithm to use when OSS creates the object. Valid value: AES256.
ObjectACL	Specifies the ACL of the created object.
Meta	Specifies the user metadata of the object. The parameters of user metadata are prefixed with <code>X-Oss-Meta-</code> .

The following code provides an example on how to configure object metadata:

```

package main
import (
    "fmt"
    "os"
    "time"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Configure object metadata: Set the expiration time to 23:00:00, 10 January 2049 in GMT, ACL to public read, and MyProp to MyPropVal as user me
    tadata.
    expires := time.Date(2049, time.January, 10, 23, 0, 0, time.UTC)
    options := []oss.Option{
        oss.Expires(expires),
        oss.ObjectACL(oss.ACLPublicRead),
        oss.Meta("MyProp", "MyPropVal"),
    }
    // Use a data stream to upload the object.
    err = bucket.PutObject("<yourObjectName>", strings.NewReader("MyObjectValue"), options...)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the object metadata.
    props, err := bucket.GetObjectDetailedMeta("<yourObjectName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object Meta:", props)
}

```

Modify object metadata

The following code provides an example on how to modify a single or multiple object metadata parameters at a time:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Modify a single object metadata parameter at a time.
    err = bucket.SetObjectMeta(objectName, oss.Meta("MyMeta", "MyMetaValue1"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Modify multiple object metadata parameters at a time.
    options := []oss.Option{
        oss.Meta("MyMeta", "MyMetaValue2"),
        oss.Meta("MyObjectLocation", "HangZhou"),
    }
    err = bucket.SetObjectMeta(objectName, options...)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the object metadata.
    props, err := bucket.GetObjectDetailedMeta(objectName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object Meta:", props)
}

```

Obtain object metadata

The following code provides an example on how to obtain object metadata:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the object metadata.
    props, err := bucket.GetObjectDetailedMeta(objectName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object Meta:", props)
}

```

1.4.5.8.5. List objects

This topic describes how to list the objects that are contained in a bucket.

Objects in a bucket are sorted alphabetically. You can use `Bucket.ListObjects` to list the objects in a bucket. The following table describes the parameters for `Bucket.ListObjects`.

Parameter	Description
delimiter	Groups objects by name. Objects whose names contain the same string from the prefix and the next occurrence of the delimiter are grouped as a single result element in <code>commonPrefixes</code> .
prefix	Specifies the prefix that returned object names must contain.
maxKeys	Specifies the maximum number of objects that can be listed each time. The default value is 100. The maximum value is 1000.
marker	Specifies the name of the object after which the list begins.

For the complete code that is used to list objects, visit [GitHub](#).

Simple list

You can use the default parameters to query the list of objects in a bucket. By default, a maximum of 100 objects are listed. The following code provides an example on how to perform simple list:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // List all objects.
    marker := ""
    for {
        lsRes, err := bucket.ListObjects(oss.Marker(marker))
        if err != nil {
            HandleError(err)
        }
        // Display the listed objects. By default, a maximum of 100 records are returned each time.
        for _, object := range lsRes.Objects {
            fmt.Println("Bucket: ", object.Key)
        }
        if lsRes.IsTruncated {
            marker = lsRes.NextMarker
        } else {
            break
        }
    }
}
```

List a specified number of objects

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Set the maximum number of objects to list and list the objects.
    lsRes, err := bucket.ListObjects(oss.MaxKeys(200))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the listed objects.
    fmt.Println("Objects:", lsRes.Objects)
    for _, object := range lsRes.Objects {
        fmt.Println("Object:", object.Key)
    }
}

```

List objects whose names contain a specified prefix

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the objects whose names contain the specified prefix. By default, a maximum of 100 objects are listed.
    lsRes, err := bucket.ListObjects(oss.Prefix("my-object-"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the listed objects.
    fmt.Println("Objects:", lsRes.Objects)
    for _, object := range lsRes.Objects {
        fmt.Println("Object:", object.Key)
    }
}

```

List objects whose names come after a specified marker

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the objects whose names come after the specified marker. By default, a maximum of 100 objects are listed.
    lsRes, err := bucket.ListObjects(oss.Marker("my-object-xx"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the listed objects.
    fmt.Println("Objects:", lsRes.Objects)
    for _, object := range lsRes.Objects {
        fmt.Println("Object:", object.Key)
    }
}

```

List all objects by page

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List all objects by page, with up to 200 objects on each page.
    marker := oss.Marker("")
    for {
        lsRes, err := bucket.ListObjects(oss.MaxKeys(200), marker)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        marker = oss.Marker(lsRes.NextMarker)
        fmt.Println("Objects:", lsRes.Objects)
        if !lsRes.IsTruncated {
            break
        }
    }
}

```

List objects whose names come after a specified marker by page

The following code provides an example on how to list the objects that follow a specified marker by page. The maximum number of objects that can be listed on each page is specified by MaxKeys.


```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the objects whose names come after the specified marker by page, with up to 50 objects on each page.
    marker = oss.Marker("my-object-xx")
    for {
        lsRes, err := bucket.ListObjects(oss.MaxKeys(50), marker)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        marker = oss.Marker(lsRes.NextMarker)
        fmt.Println("Objects:", lsRes.Objects)
        if !lsRes.IsTruncated {
            break
        }
    }
}

```

List objects whose names contain a specified prefix by page

The following code provides an example on how to list the objects whose names contain a specified prefix by page. The maximum number of objects to list on each page is specified by `maxKeys`.

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the objects whose names contain the specified prefix by page, with up to 80 objects on each page.
    prefix := oss.Prefix("my-object-")
    marker := oss.Marker("")
    for {
        lsRes, err := bucket.ListObjects(oss.MaxKeys(80), marker, prefix)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        prefix = oss.Prefix(lsRes.Prefix)
        marker = oss.Marker(lsRes.NextMarker)
        fmt.Println("Objects:", lsRes.Objects)
        if !lsRes.IsTruncated {
            break
        }
    }
}

```

List the information about the objects whose names contain a specified prefix

The following code provides an example on how to list the information about the objects whose names contain a specified prefix. Such information includes the object size, the time when the object was last modified, and the object name.

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Traverse the objects.
    marker := oss.Marker("")
    prefix := oss.Prefix("<yourObjectPrefix>")
    for {
        lor, err := bucket.ListObjects(marker, prefix)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        for _, object := range lor.Objects {
            fmt.Printf("%s %d %s\n", object.LastModified, object.Size, object.Key)
        }
        prefix = oss.Prefix(lor.Prefix)
        marker = oss.Marker(lor.NextMarker)
        if ! lor.IsTruncated {
            break
        }
    }
}

```

List the information about the subfolders whose names contain a specified prefix

The following code provides an example on how to list the information about the subfolders whose names contain a specified prefix:

```

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    marker := oss.Marker("")
    prefix := oss.Prefix("<yourDirPrefix>")
    for {
        lor, err := bucket.ListObjects(marker, prefix, oss.Delimiter("/"))
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        for _, dirName := range lor.CommonPrefixes {
            fmt.Println(dirName)
        }
        prefix = oss.Prefix(lor.Prefix)
        marker = oss.Marker(lor.NextMarker)
        if ! lor.IsTruncated {
            break
        }
    }
}

```

1.4.5.8.6. Delete objects

This topic describes how to delete objects.

 **Warning** Deleted objects cannot be recovered. Exercise caution when you delete objects.

For the complete code that is used to delete an object, visit [GitHub](#).

Delete a single object

The following code provides an example on how to delete a single object:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    // Specify the name of the bucket from which you want to delete the object.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Delete the object. objectName indicates the full path that includes the extension of the object. Example: abc/efg/123.jpg.
    // To delete a directory, set objectName to the directory name. If the directory contains objects, delete all objects from the directory before you delete the directory.
    err = bucket.DeleteObject(objectName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Delete multiple objects

You can call the `Bucket.DeleteObjects` operation to delete multiple objects and specify the `DeleteObjectsQuiet` parameter to specify whether to return results of delete operations. By default, information of deleted objects is returned.

The following code provides an example on how to delete multiple objects at a time:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    // Specify the name of the bucket from which you want to delete the object.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify that information of deleted objects is returned.
    delRes, err := bucket.DeleteObjects([]string{"my-object-1", "my-object-2"})
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Deleted Objects:", delRes.DeletedObjects)
    // Specify that results of delete operations are not returned.
    _, err = bucket.DeleteObjects([]string{"my-object-3", "my-object-4"},
    oss.DeleteObjectsQuiet(true))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Delete objects whose names contain a specified prefix

The following code provides an example on how to delete objects whose names contain a specified prefix:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Specify the name of the bucket from which you want to delete the objects.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // List all objects whose names contain the specified prefix and delete the objects.
    marker := oss.Marker("")
    prefix := oss.Prefix("<yourObjectPrefix>")
    count := 0
    for {
        lor, err := bucket.ListObjects(marker, prefix)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }

        objects := []string{}
        for _, object := range lor.Objects {
            objects = append(objects, object.Key)
        }
        delRes, err := bucket.DeleteObjects(objects, oss.DeleteObjectsQuiet(true))
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }

        if len(delRes.DeletedObjects) > 0 {
            fmt.Println("these objects deleted failure,", delRes.DeletedObjects)
            os.Exit(-1)
        }

        count += len(objects)
        prefix = oss.Prefix(lor.Prefix)
        marker = oss.Marker(lor.NextMarker)
        if !lor.IsTruncated {
            break
        }
    }

    fmt.Printf("success,total delete object count:%d\n", count)
}
```

1.4.5.8.7. Copy objects

This topic describes how to copy objects.

For the complete code that is used to copy objects, visit [GitHub](#).

Copy objects within a bucket

The following code provides an example on how to copy an object within a bucket:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    destObjectName := "<yourDestObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Copy the object to another object within the bucket.
    _, err = bucket.CopyObject(objectName, destObjectName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

Copy objects across buckets

You can copy objects from other buckets to the current bucket or copy objects from the current bucket to other buckets. The following code provides an example on how to copy objects across buckets:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    srcBucketName := "<yourSrcBucketName>"
    dstBucketName := "<yourDstBucketName>"
    srcObjectName := "<yourSrcObjectName>"
    dstObjectName := "<yourDstObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Copy srcObjectName from srcBucketName to the current bucket.
    bucket.CopyObjectFrom(srcBucketName, srcObjectName, dstObjectName)
    if err != nil {
        fmt.Println("CopyObjectFrom Error:", err)
        os.Exit(-1)
    }
    // Copy srcObjectName from the current bucket to dstBucketName.
    bucket.CopyObjectTo(dstBucketName, dstObjectName, srcObjectName)
    if err != nil {
        fmt.Println("CopyObjectTo Error:", err)
        os.Exit(-1)
    }
}

```

Process object metadata during object copy

You can use `MetadataDirective` to process object metadata during object copy. Valid values of `MetadataDirective`:

- `oss.MetaCopy`: This is the default value. Copy metadata of the source object. The metadata of the destination object is the same as the metadata of the source object.
- `oss.MetaReplace`: overwrites the source object metadata by using the specified object metadata.

 **Notice** All metadata of the source object is overwritten if you use `oss.MetaReplace`. The metadata of the source object cannot be recovered. Exercise caution when you perform this operation.

The following table describes the parameter you can configure when you set `MetadataDirective` to `oss.MetaReplace`.

Parameter	Description
CacheControl	Specifies the caching behavior of a web page when the destination object is downloaded.
ContentDisposition	Specifies the name of the destination object when it is downloaded.
ContentEncoding	Specifies the encoding format that is used when the destination object is downloaded.
Expires	Specifies the expiration time of the cache in GMT.
ServerSideEncryption	Specifies the server-side encryption algorithm that is used when OSS creates the destination object. Valid value: AES256.
ObjectACL	Specifies the ACL of the destination object.
Meta	Specifies the user metadata of the destination object. The parameters of user metadata are prefixed with <code>X-Oss-Meta-</code> .

The following code provides an example on how to process object metadata during object copy:

```
package main
import (
    "fmt"
    "os"
    "time"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    destObjectName := "<yourDestObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify metadata for the destination object.
    expires := time.Date(2049, time.January, 10, 23, 0, 0, 0, time.UTC)
    options := []oss.Option{
        oss.MetadataDirective(oss.MetaReplace),
        oss.Expires(expires),
        oss.ObjectACL(oss.ACLPublicRead),
        oss.Meta("MyMeta", "MyMetaValue")
    }
    // Overwrite the metadata of the source object by using the specified metadata.
    _, err = bucket.CopyObject(objectName, destObjectName, options...)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Conditional copy

OSS allows you to configure one or more conditions for object copy. If the specified conditions are met, the object can be copied. If the specified conditions are not met, an error code is returned and the object cannot be copied. The following table describes the conditions that you can configure in object copy.

Parameter	Description
CopySourceIfMatch	If the ETag of the source object is the same as that of the destination object, the object is copied. Otherwise, an error is returned.
CopySourceIfNoneMatch	If the ETag of the source object is different from that of the destination object, the object is copied. Otherwise, an error is returned.
CopySourceIfModifiedSince	If the specified time is earlier than the actual modified time of the object, the object is copied. Otherwise, an error is returned.
CopySourceIfUnmodifiedSince	If the specified time is equal to or later than the actual modified time of the object, the object is copied. Otherwise, an error is returned.

You can set `CopySourceIfMatch` and `CopySourceIfNoneMatch` at the same time. You can set `CopySourceIfModifiedSince` and `CopySourceIfUnmodifiedSince` at the same time.

The following code provides an example on how to perform conditional copy:

```

package main
import (
    "fmt"
    "os"
    "time"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // bucketName := "<yourBucketName>"
    // objectName := "<yourObjectName>"
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    destMatchObjectName := "<yourMatchDestObjectName>"
    destUnMatchObjectName := "<yourUnmatchDestObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    date := time.Date(2011, time.November, 10, 23, 0, 0, 0, time.UTC)
    // Copy the objects that meet the specified condition.
    _, err = bucket.CopyObject(objectName, destMatchObjectName, oss.CopySourceIfModifiedSince(date))
    if err != nil {
        fmt.Println("CopyObject CopySourceIfModifiedSince Error:", err)
    }
    // Do not copy the objects that cannot meet the specified condition.
    _, err = bucket.CopyObject(objectName, destUnMatchObjectName, oss.CopySourceIfUnmodifiedSince(date))
    if err != nil {
        fmt.Println("CopyObject CopySourceIfUnmodifiedSince Error:", err)
    }
}

```

1.4.5.9. Versioning

1.4.5.9.1. Manage versioning

The versioning state of a bucket takes effect on all objects in the bucket.

If you enable versioning for a bucket, you can recover any previous version of an object in the bucket when you accidentally overwrite or delete the object. A bucket can be in one of the following versioning states: unversioned (default), versioning-enabled, or versioning-suspended.

 **Notice** If versioning is enabled for a bucket, the versioning state of the bucket cannot be set back to unversioned. However, you can set the versioning state of the bucket to suspended.

Configure versioning for a bucket

The following code provides an example on how to set the versioning state of a bucket to enabled or suspended:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Create a bucket.
    err = client.CreateBucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Set the versioning state of the bucket to Enabled or Suspended. In this example, the versioning state of the bucket is set to Enabled.
    config := oss.VersioningConfig{Status: "Enabled"}
    err = client.SetBucketVersioning("<yourBucketName>", config)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

Query the versioning state of a bucket

The following code provides an example on how to query the versioning state of a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the versioning state of the bucket.
    ret, err := client.GetBucketVersioning("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the versioning state of the bucket.
    fmt.Println("The bucket Versioning status:", ret.Status)
}
```

List the versions of all objects in a bucket

The following code provides an example on how to list the versions of all objects including delete markers in a specified bucket:

```
package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the objects whose names contain the specified prefix.
    ret, err := bucket.ListObjectVersions(oss.Prefix("yourObjectPrefix"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the version information about the objects.
    for _, object := range ret.ObjectVersions {
        fmt.Println("Object:", object)
    }
    // Display the version information about the delete markers.
    for _, marker := range ret.ObjectDeleteMarkers {
        fmt.Println("Delete Markers:", marker)
    }
}
```

1.4.5.9.2. Upload objects

This topic describes how to upload objects to a versioned bucket.

Simple upload

If you initiate a PutObject request to a versioning-enabled bucket, Object Storage Service (OSS) generates a unique version ID for the object to upload and includes the version ID in the x-oss-version-id response header. In a versioning-suspended bucket, the version ID generated for the object is null. If you upload an object with the same name as an existing object to a versioning-suspended bucket, the existing object is overwritten. This way, each object has only a single version whose version ID is null.

The following code provides an example on how to perform simple upload:


```

package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    var retHeader http.Header
    // Upload the string. Use oss.GetResponseHeader to obtain the returned header.
    err = bucket.PutObject("<yourObjectName>", strings.NewReader("yourObjectValue"), oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the value of x-oss-version-id.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
}

```

Append upload

In a versioning-enabled bucket, the AppendObject operation can be performed only on the current version of an appendable object.

Note

The following code provides an example on how to perform append upload:

```

package main
import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // The position for the first append upload is 0, and the position for the next append upload is included in the response. The position from which
    the next append upload starts is the current length of the object.
    var retHeader http.Header
    var nextPos int64 = 0
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectAppendValue1"), nextPos, oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the value of x-oss-version-id.
    fmt.Println("x-oss-version-id:", retHeader.Get("x-oss-version-id"))
    // Perform the second append operation. Use oss.GetResponseHeader to obtain the returned header.
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectAppendValue2"), nextPos, oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the value of x-oss-version-id.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
    // You can append content to an object multiple times.
}

```

Multipart upload

If you call `CompleteMultipartUpload` to complete the multipart upload task for an object in a versioning-enabled bucket, Object Storage Service (OSS) generates a unique version ID for the object and returns the version ID as the value of the `x-oss-version-id` response header.

The following code provides an example on how to perform multipart upload:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localFilename := "<yourLocalFilename>"
    // Use oss.GetResponseHeader to obtain the returned header.
    var retHeader http.Header
    // Query the bucket information.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    chunks, err := oss.SplitFileByPartNum(localFilename, 3)
    fd, err := os.Open(localFilename)
    defer fd.Close()
    // Step 1: Initiate a multipart upload event.
    imur, err := bucket.InitiateMultipartUpload(objectName)
    // Step 2: Upload parts.
    var parts []oss.UploadPart
    for _, chunk := range chunks {
        fd.Seek(chunk.Offset, os.SEEK_SET)
        // Call the UploadPart method to upload each part.
        part, err := bucket.UploadPart(imur, fd, chunk.Size, chunk.Number)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        parts = append(parts, part)
    }
    // Step 3: Complete the multipart upload task.
    cmur, err := bucket.CompleteMultipartUpload(imur, parts, oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("cmur:", cmur)
    // Display the value of x-oss-version-id.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
}
```

1.4.5.9.3. Download objects

By default, when you perform the `GetObject` operation on an object in a versioned bucket, only the current version of the object is returned.

When you perform the `GetObject` operation on an object in a bucket, you may obtain one of the following results:

- If the current version of the object is a delete marker, OSS returns 404 Not Found.
- If the version ID of the object is specified in the request, OSS returns the specified version of the object. If the version ID is set to null in the request, OSS returns the version whose version ID is null.
- If the version ID is set to a delete marker, OSS returns 405 Method Not Allowed.

The following code provides an example on how to download an object:

```

package main
import (
    "fmt"
    "net/http"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    var retHeader http.Header
    // Download the version whose ID is yourObjectVersionId to the cache.
    _, err = bucket.GetObject("yourObjectName", oss.VersionId("yourObjectVersionId"), oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the value of x-oss-version-id.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
}

```

1.4.5.9.4. Copy objects

This topic describes how to copy objects in a versioned bucket. You can call `CopyObject` to copy an object that is up to 1 GB in size and call `UploadPartCopy` to copy an object that is larger than 1 GB.

Copy a small object

You can call `CopyObject` to copy an object that is up to 1 GB in size from a source bucket to a destination bucket within the same region.

- By default, the `x-oss-copy-source` header in a `CopyObject` request specifies the current version of the object to copy. If the current version of the object is a delete marker, OSS returns the 404 status code, which indicates that the object does not exist. You can add a version ID in the `x-oss-copy-source` header to copy a specified object version. Delete markers cannot be copied.
- You can copy a previous version of an object to the same bucket. The copied previous version becomes the new version of the object. This way, the previous version of the object is recovered.
- If versioning is enabled for the destination bucket, OSS generates a unique version ID for the destination object. The version ID is returned in the response as the `x-oss-version-id` header value. If versioning is disabled or suspended for the destination bucket, OSS generates a version whose version ID is null for the destination object and overwrites the original version whose version ID is null.
- Appendable objects in a bucket for which versioning is enabled or suspended cannot be copied.

The following code provides an example on how to copy a small object:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    destObjectName := "<yourDestObjectName>"
    // Query the bucket information.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Copy the specified version of the object to another object in the same bucket and query the version ID of the destination object.
    var retHeader http.Header
    _, err = bucket.CopyObject(objectName, destObjectName, oss.VersionId("<yourObjectVersionId>"), oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the version ID of the destination object.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
    fmt.Println("x-oss-copy-source-version-id:", oss.GetCopySrcVersionId(retHeader))
}

```

Copy a large object

To copy an object that is larger than 1 GB in size, you must split the object into parts and copy the parts by calling `UploadPartCopy`.

By default, the `UploadPartCopy` operation uploads a part by copying data from the current version of an existing object. You can add a version ID to the `x-oss-copy-source` request header as the condition to upload a part by copying data from the specified version of an existing object. Example: `x-oss-copy-source : /SourceBucketName/SourceObjectName?versionId=111111`.

Note The value of `SourceObjectName` must be URL-encoded. The version ID of the copied object is returned as the `x-oss-copy-source-version-id` header value in the response.

If version ID is not specified in the request and the current version of the source object is a delete marker, OSS returns 404 Not Found. If version ID is specified in the request and the specified version of the source object is a delete marker, OSS returns 400 Bad Request.

The following code provides an example on how to use multipart copy to copy a large object:

```

package main
import (
    "fmt"
    "net/http"
    "os"
    "time"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("yourBucketName")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "yourBucketName"
    objectName := "yourObjectName"
    fileName := "yourFileName"
    futureDate := time.Date(2049, time.January, 10, 23, 0, 0, 0, time.UTC)
    var retHeader http.Header
    chunks, err := oss.SplitFileByPartNum(fileName, 3)
    // Upload a large object as the source object.
    err = bucket.PutObjectFromFile(objectName, fileName, oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the version ID of the uploaded source object.
    versionId := oss.GetVersionId(retHeader)
    options := []oss.Option{
        oss.Expires(futureDate), oss.Meta("my", "myprop"),
    }
    // Specify the name of the destination object.
    objectDest := objectName + "dest"
    var parts []oss.UploadPart
    // Configure the parameters for the object copy operation.
    copyOptions := []oss.Option{
        oss.VersionId(versionId), oss.GetResponseHeader(&retHeader),
    }
    // Copy the large object.
    imu, err := bucket.InitiateMultipartUpload(objectDest, options...)
    for _, chunk := range chunks {
        part, err := bucket.UploadPartCopy(imu, bucketName, objectName, chunk.Offset, chunk.Size, (int)(chunk.Number), copyOptions...)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        parts = append(parts, part)
        // Display the value of x-oss-copy-source-version-id.
        fmt.Println("x-oss-copy-source-version-id:", oss.GetCopySrcVersionId(retHeader))
    }
    _, err = bucket.CompleteMultipartUpload(imu, parts, oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the value of x-oss-version-id.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
}

```

1.4.5.9.5. List objects

You can list objects in a versioned bucket, including all objects, objects whose names contain a specified prefix, and objects and subdirectories in a specified directory.

Scenarios

The following figure shows the directories and objects in a bucket named examplebucket.

```

examplebucket
├── fun
│   ├── exampleobject.jpg
│   ├── examplefile.txt
│   └── destfolder
│       ├── image1.jpg
│       └── image2.png
├── srcfile.txt
└── oss.jpg

```

The following examples show how to list different objects by specifying conditions.

List the versions of all objects in a bucket

The following code provides an example on how to list the versions of all objects including delete markers in a specified bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // Create an OSSClient instance.
    // Specify the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set the endpoint to https://oss-cn-hangzhou.aliyuncs.com.
    // The AccessKey pair of an Apsara Stack tenant account has permissions on all API operations. Using these credentials to perform operations in OSS is a high-risk operation. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
    client, err := oss.New("yourEndpoint", "yourAccessKeyId", "yourAccessKeySecret")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "examplebucket"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // List the versions of all objects including delete markers in the bucket.
    keyMarker := oss.KeyMarker("")
    // The VersionIdMarker parameter is configured together with the KeyMarker parameter to specify the position from which the list operation starts
    versionIdMarker := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(keyMarker, versionIdMarker)
        if err != nil {
            HandleError(err)
        }
        // Display the versions of the listed objects.
        for _, dirName := range lor.ObjectVersions {
            fmt.Println("Versionid:", dirName.VersionId)
            fmt.Println("Key:", dirName.Key)
            fmt.Println("Is Latest", dirName.IsLatest)
        }
        // View the versions of the listed delete markers.
        for _, marker := range lor.ObjectDeleteMarkers {
            fmt.Println(marker.VersionId)
            fmt.Println(marker.Key)
        }
        // Check whether the required versions are listed. If the versions are incompletely listed, the list operation continues. If the versions are completely listed, the list operation stops.
        keyMarker = oss.KeyMarker(lor.NextKeyMarker)
        versionIdMarker = oss.VersionIdMarker(lor.NextVersionIdMarker)
        if !lor.IsTruncated {
            break
        }
    }
}
```

Returned results

```

Versionid: CAEQChiBgIDHzNPETHciIDYzZGQ5M2VjOTU2ODRmMzA4ZW40ODk0NjVmMWE****
Key: fun/
Is Latest true
Versionid: CAEQChiBgID6ltnEthciIDNmOGM2MTQ3ZjU1NTQ2MGFhYWJjNjQxMTQxZWZh****
Key: fun/destfolder/
Is Latest true
Versionid: CAEQChiBgMDczt3EthciIDEwYjliZmQ4MDNiMDQ2Njk4YWMxM2NhM2E5NzQ3****
Key: fun/destfolder/image1.jpg
Is Latest true
Versionid: CAEQChiBgIDszt3EthciIGU5OWU0ZTllMGY3NTRmMmU5NzVjZmJkYmE3ZWY****
Key: fun/destfolder/image2.png
Is Latest true
Versionid: CAEQChiBgICditzEthciIDbINTglZTZkMDRlMzRjNDdiMjRjMTBlOGUzMTM0****
Key: fun/examplefile.txt
Is Latest true
Versionid: CAEQChiBgMC.itzEthciIDEzNWVlYjNhYzNmMjQ4NW45Nzc2NzllY2FiYmQ3****
Key: fun/exampleobject.jpg
Is Latest true
Versionid: CAEQChiBgIDMgdbEthciIDUyMGIOmZlNThkODQwY2ZhNmZhNTQ1Njk4ZTd****
Key: oss.jpg
Is Latest true
Versionid: CAEQChiBgICigdbEthciIDdlM2Q1YjYxZDIyZDQyMzI4MTRkNzVmYzdiMTBh****
Key: srcfile.txt
Is Latest true

```

List the versions of objects whose names contain a specified prefix

The following code provides an example on how to list the versions of objects whose names contain a specified prefix:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("yourEndpoint", "yourAccessKeyId", "yourAccessKeySecret")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "examplebucket"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Specify the Prefix parameter to list the versions of objects whose names contain the fun prefix.
    prefix := oss.Prefix("fun")
    keyMarker := oss.KeyMarker("")
    versionIdMarker := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(prefix, keyMarker, versionIdMarker)
        if err != nil {
            HandleError(err)
        }
        // Display the versions of the listed objects.
        for _, dirName := range lor.ObjectVersions {
            fmt.Println("Versionid:", dirName.VersionId)
            fmt.Println("Key:", dirName.Key)
            fmt.Println("Is Latest", dirName.IsLatest)
        }
        // Check whether the required versions are listed. If the versions are incompletely listed, the list operation continues. If the versions are
        // completely listed, the list operation stops.
        keyMarker = oss.KeyMarker(lor.NextKeyMarker)
        versionIdMarker = oss.VersionIdMarker(lor.NextVersionIdMarker)
        if !lor.IsTruncated {
            break
        }
    }
}

```

Returned results

```

Versionid: CAEQChiBgIDHzNPETHciIDYzZGQ5M2VjOTU2ODRmMzA4ZW40Dk0NjVmMWEx****
Key: fun/
Is Latest true
Versionid: CAEQChiBgID6ltnEthciIDNmOGM2MTQ3ZjU1NTQ2MGFhYWJjNjQxMTQxZWZh****
Key: fun/destfolder/
Is Latest true
Versionid: CAEQChiBgMDczt3EthciIDEwYjliZmQ4MDNiMDQ2Njk4YW40Dk0NjVmMWEx****
Key: fun/destfolder/image1.jpg
Is Latest true
Versionid: CAEQChiBgIDsz3EthciIGU5OWU0ZTllMGY3NTRmMmU5NzVjZmJkYmE3ZWYy****
Key: fun/destfolder/image2.png
Is Latest true
Versionid: CAEQChiBgICditzEthciIDbINTglZTZkMDRlMzRjNDdiMjRjMTBLOGUzMtM0****
Key: fun/examplefile.txt
Is Latest true
Versionid: CAEQChiBgMC.itzEthciIDEzNWVlYjNhYzNmMjQ4NW40Dk0NjVmMWEx****
Key: fun/exampleobject.jpg
Is Latest true

```

List a specified number of object versions

The following code provides an example on how to list a specified number of object versions:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("yourEndpoint", "yourAccessKeyId", "yourAccessKeySecret")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "examplebucket"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Specify the MaxKeys parameter to list up to four object versions in alphabetical order of the object names.
    maxkey := oss.MaxKeys(4)
    keyMarker := oss.KeyMarker("")
    versionIdMarker := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(maxkey, keyMarker, versionIdMarker)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        // Display the versions of the listed objects.
        for _, dirName := range lor.ObjectVersions {
            fmt.Println("Versionid:", dirName.VersionId)
            fmt.Println("Key:", dirName.Key)
            fmt.Println("Is Latest", dirName.IsLatest)
        }
        // Check whether the required versions are listed. If the versions are incompletely listed, the list operation continues. If the versions are
        // completely listed, the list operation stops.
        keyMarker = oss.KeyMarker(lor.NextKeyMarker)
        versionIdMarker = oss.VersionIdMarker(lor.NextVersionIdMarker)
        if !lor.IsTruncated {
            break
        }
    }
}

```

Returned results

```

Versionid: CAEQChiBgIDHzNPETHciIDYzZGQ5M2VjOTU2ODRmMzA4ZW40Dk0NjVmMWEx****
Key: fun/
Is Latest true
Versionid: CAEQChiBgID6ltnEthciIDNmOGM2MTQ3ZjU1NTQ2MGFhYWJjNjQxMTQxZWZh****
Key: fun/destfolder/
Is Latest true
Versionid: CAEQChiBgMDczt3EthciIDEwYjliZmQ4MDNiMDQ2Njk4YW40Dk0NjVmMWEx****
Key: fun/destfolder/image1.jpg
Is Latest true
Versionid: CAEQChiBgIDsz3EthciIGU5OWU0ZTllMGY3NTRmMmU5NzVjZmJkYmE3ZWYy****
Key: fun/destfolder/image2.png
Is Latest true

```


List the versions of all objects by page

The following code provides an example on how to list the versions of all objects by page:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("yourEndpoint", "yourAccessKeyId", "yourAccessKeySecret")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "examplebucket"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // List the versions of all objects in the bucket by page.
    keyMarker := oss.KeyMarker("")
    // Specify the MaxKeys parameter to list up to four object versions in alphabetical order of the object names.
    maxkey := oss.MaxKeys(4)
    versionIdMarker := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(maxkey, keyMarker, versionIdMarker)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        // Display the versions of the listed objects.
        for _, dirName := range lor.ObjectVersions {
            fmt.Println("Versionid:", dirName.VersionId)
            fmt.Println("Key:", dirName.Key)
            fmt.Println("Is Latest", dirName.IsLatest)
        }
        fmt.Println("-----")
        // Check whether the required versions are listed. If the versions are incompletely listed, the list operation continues. If the versions are
        // completely listed, the list operation stops.
        if lor.IsTruncated {
            keyMarker = oss.KeyMarker(lor.NextKeyMarker)
            versionIdMarker = oss.VersionIdMarker(lor.NextVersionIdMarker)
        } else {
            break
        }
    }
}
```

Returned results

```
Versionid: CAEQChiBgIDHzNPETHciIDYzZGQ5M2VjOTU2ODRmMzA4ZW40ODk0NjVmMWE****
Key: fun/
Is Latest true
Versionid: CAEQChiBgID6ltnETHciIDNmOGM2MTQ3ZjU1NTQ2MGFhYWJjNjQxMTQxZWZh****
Key: fun/destfolder/
Is Latest true
Versionid: CAEQChiBgMDczt3ETHciIDEwYjliZmQ4MDNiMDQ2Njk4YWMxM2NhM2E5NzQ3****
Key: fun/destfolder/image1.jpg
Is Latest true
Versionid: CAEQChiBgIDsz3ETHciIGU5OWU0ZTllMGY3NTRmMmU5NzVjZmJkYmE3ZWY****
Key: fun/destfolder/image2.png
Is Latest true
-----
Versionid: CAEQChiBgICditzETHciIDbINTg1ZTZkMDRlMzRjNDdiMjRjMTBLOGUzMTM0****
Key: fun/examplefile.txt
Is Latest true
Versionid: CAEQChiBgMC.itzETHciIDEzNWVlYjNhYzNmMjQ4NW40ODk0NjVmMWE****
Key: fun/exampleobject.jpg
Is Latest true
Versionid: CAEQChiBgIDMgdbETHciIDUyMGU0NmZlNThkODQwY2ZhNmZhNTQ1Njk4ZTdj****
Key: oss.jpg
Is Latest true
Versionid: CAEQChiBgICgdbETHciIDdlM2Q1YjYxZDIyZDQyMzI4MTRkNzVmYzdiMTBh****
Key: srcfile.txt
Is Latest true
-----
```

List objects by directory

OSS uses a flat structure instead of a hierarchical structure to store objects. A directory is an object whose size is 0 and whose name ends with a forward slash (/). You can upload and download this object. By default, objects whose names end with a forward slash (/) are displayed as directories in the OSS console.

You can specify the delimiter and prefix parameters to list objects by directory.

- If you set `prefix` to a directory name in the request, objects and subdirectories whose names contain the prefix are listed.
- If you also set `delimiter` to a forward slash (/) in the request, objects and subdirectories whose names start with the specified prefix in the directory are listed. Each subdirectory is listed as a single result element in `CommonPrefixes`. The objects and directories in these subdirectories are not listed.

The following code provides examples on how to list objects by directory.

- List the versions of objects in the root directory of a bucket

The following code provides an example on how to list the versions of objects in the root directory of a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("yourEndpoint", "yourAccessKeyId", "yourAccessKeySecret")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "examplebucket"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Set the delimiter parameter to a forward slash (/) to list the versions of objects and the names of subdirectories in the root directory.
    delimiter := oss.Delimiter("/")
    keyMarker := oss.KeyMarker("")
    versionIdMarker := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(keyMarker, delimiter, versionIdMarker)
        if err != nil {
            HandleError(err)
        }
        // Display the versions of the listed objects.
        for _, dirName := range lor.ObjectVersions {
            fmt.Println("Versionid:", dirName.VersionId)
            fmt.Println("Key:", dirName.Key)
            fmt.Println("Is Latest", dirName.IsLatest)
        }
        // Display the directories whose names end with a forward slash (/).
        for _, common_prefix := range lor.CommonPrefixes {
            fmt.Println("common_prefix:", common_prefix)
        }
        // Check whether the required versions are listed. If the versions are incompletely listed, the list operation continues. If the versions are completely listed, the list operation stops.
        if lor.IsTruncated {
            keyMarker = oss.KeyMarker(lor.NextKeyMarker)
            versionIdMarker = oss.VersionIdMarker(lor.NextVersionIdMarker)
        } else {
            break
        }
    }
}
```

Returned results

```
Versionid: CAEQChiBgIDMgdbEthciIDUyMGI0NmZlNThKODQwY2ZhNmZhNTQ1Njk4ZTdjdj****
Key: oss.jpg
Is Latest true
Versionid: CAEQChiBgICgdbEthciIDdlM2Q1YjYxZDIyZDQyMzI4MTRkNzVmYzdiMTBh****
Key: srcfile.txt
Is Latest true
common_prefix: fun/
```

- List the objects and subdirectories in a specified directory

The following code provides an example on how to list the objects and the subdirectories in a directory in a bucket:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("yourEndpoint", "yourAccessKeyId", "yourAccessKeySecret")
    if err != nil {
        HandleError(err)
    }
    // Specify the name of the bucket.
    bucketName := "examplebucket"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Configure the Prefix parameter to list all objects and subdirectories in the fun folder. Set the delimiter parameter to a forward slash (/).

    prefix := oss.Prefix("fun/")
    delimiter := oss.Delimiter("/")
    keyMarker := oss.KeyMarker("")
    versionIdMarker := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(prefix, delimiter, keyMarker, versionIdMarker)
        if err != nil {
            HandleError(err)
        }
        // Display the versions of the listed objects.
        for _, dirName := range lor.ObjectVersions {
            fmt.Println("Versionid:", dirName.VersionId)
            fmt.Println("Key:", dirName.Key)
            fmt.Println("Is Latest:", dirName.IsLatest)
        }
        // Display the directories whose names end with a forward slash (/).
        for _, common_prefix := range lor.CommonPrefixes {
            fmt.Println("common_prefix:", common_prefix)
        }
        // Check whether the required versions are listed. If the versions are incompletely listed, the list operation continues. If the versions are completely listed, the list operation stops.
        if lor.IsTruncated {
            keyMarker = oss.KeyMarker(lor.NextKeyMarker)
            versionIdMarker = oss.VersionIdMarker(lor.NextVersionIdMarker)
        } else {
            break
        }
    }
}

```

Returned results

```

Versionid: CAEQChiBgIDHzNPETHciIDYzZGQ5M2VjOTU2ODRmMzA4ZW44ODk0NjVmMWE*****
Key: fun/
Is Latest: true
Versionid: CAEQChiBgICditzEthciIDBiNTglZTZkMDRlMzRjNDdiMjRjMTBLOGUzMtM0*****
Key: fun/examplefile.txt
Is Latest: true
Versionid: CAEQChiBgMC.itzEthciIDezNWVlYjNhYzNmMjQ4NW44ODk0NjVmMWE*****
Key: fun/exampleobject.jpg
Is Latest: true
common_prefix: fun/destfolder/

```

1.4.5.9.6. Delete objects

This topic describes how to delete a single object, multiple objects, or objects whose names contain a specified prefix from a versioned bucket.

Delete operations in a versioned bucket

When you delete an object from a versioned bucket, you must determine whether to specify a version ID in the request.

- Delete an object without specifying a version ID (temporary deletion)

By default, if you do not specify the version ID of the object that you want to delete in the request, Object Storage Service (OSS) does not delete the current version of the object but adds a delete marker to the object as the new current version. If you perform the `GetObject` operation on the object, OSS identifies the current version of the object as a delete marker and returns `404 Not Found`. Additionally, the `x-oss-delete-marker` header and the `x-oss-version-id` header that indicates the version ID of the created delete marker are included in the response to the `DeleteObject` request.

The value of the `x-oss-delete-marker` header in the response is true, which indicates that the `x-oss-version-id` value is the version ID of a delete marker.

- Delete an object by specifying a version ID (permanent deletion)

If you specify the version ID of the object that you want to delete in the request, OSS permanently deletes the version specified by the `versionId` field in the `params` parameter. To delete a version whose ID is null, add `params['versionId'] = "null"` to `params` in the request. OSS identifies the string "null" as the ID of the version to delete and deletes the version whose ID is null.

Delete a single object

The following examples show how to permanently or temporarily delete a single object from a versioned bucket.

- Permanently delete an object by specifying its version ID

The following code provides an example on how to permanently delete an object from a versioned bucket by specifying the version ID of the object in the request:

```
package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Permanently delete an object by specifying the version ID of the object.
    err = bucket.DeleteObject(key, oss.VersionId("yourObjectVersionId"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

- Temporarily delete an object without specifying its version ID

The following code provides an example on how to temporarily delete an object from a versioned bucket by sending a request in which no version ID is specified:

```
package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Temporarily delete an object without specifying its version ID. A delete marker is added to the object.
    err = bucket.DeleteObject("yourObjectName", oss.GetResponseHeader(&retHeader))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the information about the delete markers.
    fmt.Println("x-oss-version-id:", oss.GetVersionId(retHeader))
    fmt.Println("x-oss-delete-marker:", oss.GetDeleteMark(retHeader))
}
```

Delete multiple objects

The following examples describe how to permanently or temporarily delete multiple objects or delete markers from a versioned bucket.

- Permanently delete multiple objects by specifying their version IDs

The following code provides an example on how to permanently delete multiple objects with specified version IDs from a versioned bucket:

```

package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("yourBucketName")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the version IDs of the objects that you want to delete.
    keyArray := []oss.DeleteObject{
        oss.DeleteObject{Key: "objectName1", VersionId:"objectVersionId1"},
        oss.DeleteObject{Key: "objectName2", VersionId:"objectVersionId2"},
    }
    // Delete the objects with the specified version IDs.
    ret, err := bucket.DeleteObjectVersions(keyArray)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the information about the deleted objects.
    for _, object := range ret.DeletedObjectsDetail {
        fmt.Println("Key:", object.Key, "\n VersionId:", object.VersionId)
    }
}

```

- Permanently delete multiple delete markers by specifying their version IDs

The following code provides an example on how to permanently delete multiple delete markers with specified version IDs from a versioned bucket:

```

package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("yourBucketName")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the version IDs of the delete markers that you want to delete.
    keyArray := []oss.DeleteObject{
        oss.DeleteObject{Key: "objectName1", VersionId:"objectVersionId1"},
        oss.DeleteObject{Key: "objectName2", VersionId:"objectVersionId2"},
    }
    // Permanently delete the delete markers with the specified version IDs.
    ret, err := bucket.DeleteObjectVersions(keyArray)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the deleted objects and delete markers.
    for _, object := range ret.DeletedObjectsDetail {
        fmt.Println("key:", object.Key, "\n versionId:", object.VersionId, "\n DeleteMarker:", object.DeleteMarker, "\nDeleteMarkerVersionId", object.DeleteMarkerVersionId)
    }
}

```

- Temporarily delete multiple objects without specifying their version IDs

The following code provides an example on how to delete multiple objects from a versioned bucket by sending a request in which no version IDs are specified:

```
package main
import (
    "fmt"
    "net/http"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("yourBucketName")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Temporarily delete multiple object without specifying their version IDs. Delete markers are added to the objects.
    keyArray := []oss.DeleteObject{
        oss.DeleteObject{Key: "objectName1"},
        oss.DeleteObject{Key: "objectName2"},
    }
    res, err := bucket.DeleteObjectVersions(keyArray)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Display the delete markers that are added to the objects.
    for _, object := range res.DeletedObjectsDetail {
        fmt.Println("key:", object.Key, "\n DeleteMarker:", object.DeleteMarker, "\nDeleteMarkerVersionId", object.DeleteMarkerVersionId)
    }
}
```

Delete objects whose names contain a specified prefix

The following code provides an example on how to delete objects whose names contain a specified prefix:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Specify the name of the bucket.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // List the objects whose names contain the specified prefix and delete these objects.
    prefix := oss.Prefix("<yourObjectPrefix>")
    marker := oss.KeyMarker("")
    version := oss.VersionIdMarker("")
    for {
        lor, err := bucket.ListObjectVersions(marker, prefix, version)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        objects := []oss.DeleteObject{}
        for _, object := range lor.ObjectDeleteMarkers {
            objects = append(objects, oss.DeleteObject{
                Key:      object.Key,
                VersionId: object.VersionId,
            })
        }
        for _, object := range lor.ObjectVersions {
            objects = append(objects, oss.DeleteObject{
                Key:      object.Key,
                VersionId: object.VersionId,
            })
        }
        delRes, err := bucket.DeleteObjectVersions(objects, oss.DeleteObjectsQuiet(true))
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        if len(delRes.DeletedObjectsDetail) > 0 {
            fmt.Println("these objects deleted failure,", delRes.DeletedObjectsDetail)
            os.Exit(-1)
        }
        prefix = oss.Prefix(lor.Prefix)
        marker = oss.KeyMarker(lor.NextKeyMarker)
        version = oss.VersionIdMarker(lor.NextVersionIdMarker)
        if !lor.IsTruncated {
            break
        }
    }
    fmt.Printf("delete success\n")
}

```

1.4.5.9.7. Query object metadata

By default, when you call the `HeadObject` operation on an object in a versioned bucket, only the metadata of the current version of the object is returned.

Note If the current version of the object is a delete marker, OSS returns 404 Not Found. If you specify the version ID in the request, OSS returns the metadata of the specified version of the object.

The following code provides an example on how to query the metadata of an object:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    objectName := "<yourObjectName>"
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query partial metadata of the specified version of the object.
    props, err := bucket.GetObjectMeta(objectName, oss.VersionId("youObjectVersionId"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object Meta:", props)
    // Query all metadata of the specified version of the object.
    props, err := bucket.GetObjectDetailedMeta(objectName, oss.VersionId("youObjectVersionId"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object Meta:", ret)
}

```

1.4.5.9.8. Manage the ACL of an object

This topic describes how to manage the ACL of an object in a versioned bucket.

Configure the ACL of an object

By default, the PutObjectACL operation is called to configure the ACL of the current version of an object. If the current version of the object is a delete marker, OSS returns 404 Not Found. You can specify a version ID in the request to configure the ACL of the specified version of an object.

The following code provides an example on how to configure the ACL of an object:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Configure the ACL of the specified version of the object.
    err = bucket.SetObjectACL("<yourObjectName>", oss.ACLPublicReadWrite, oss.VersionId("youObjectVersionId"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

Query the ACL of an object

By default, the GetObjectACL operation is called to query the ACL of the current version of an object. If the current version of the object is a delete marker, OSS returns 404 Not Found. You can specify a version ID in the request to query the ACL of the specified version of an object.

The following code provides an example on how to query the ACL of an object:


```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the bucket information.
    bucket, err := client.Bucket("<yourBucketName>")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Query the ACL of the specified version of the object.
    aclRes, err := bucket.GetObjectACL("<yourObjectName>", oss.VersionId("youObjectVersionId"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object ACL:", aclRes.ACL)
}
```

1.4.5.10. Authorized access

This topic describes how to authorize access to Object Storage Service (OSS) by using Security Token Service (STS) or a signed URL.

Use STS for temporary access authorization

Alibaba Cloud STS is a web service that provides a temporary access token to a cloud computing user. You can use STS to grant a set of temporary access credentials that have a custom validity period and custom permissions to a third-party application or a RAM user managed by you.

STS provides the following benefits:

- You need only to generate an access token and send the access token to a third-party application. You do not need to expose your AccessKey pair to the third-party application. You can specify the access permissions and validity period of this token.
- The token automatically expires after the validity period. Therefore, you do not need to manually revoke the access permissions of a token.

The following code provides an example on how to create a signature request by using STS:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"
// After you obtain a temporary STS credential for your OSSClient, an OSSClient instance is created based on the security token and temporary AccessKey
// pair that are contained in the credential. The AccessKey pair consists of an AccessKey ID and an AccessKey secret.
// Create an OSSClient instance.
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.SecurityToken("<yourSecurityToken>"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
// Perform operations in OSS.
}
```

Use a signed URL for temporary access authorization

You can generate a signed URL and provide the URL to a visitor for temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of time during which the visitor can access OSS.

For the complete code that is used to authorize temporary access by using a signed URL, visit [GitHub](#).

- Use a signed URL to upload an object

The following code provides an example on how to use a signed URL to upload an object:

```

package main
import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localFilename := "<yourLocalFilename>"
    // Specify the name of the bucket to which to upload the object.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Upload the object by using the signed URL.
    signedURL, err := bucket.SignURL(objectName, oss.HTTPPut, 60)
    if err != nil {
        HandleError(err)
    }
    var val = "Go with Alibaba Cloud"
    err = bucket.PutObjectWithURL(signedURL, strings.NewReader(val))
    if err != nil {
        HandleError(err)
    }
    // Upload the object by using the signed URL with optional parameters configured.
    options := []oss.Option{
        oss.Meta("myprop", "mypropval"),
        oss.ContentType("image/tiff"),
    }
    signedURL, err = bucket.SignURL(objectName, oss.HTTPPut, 60, options...)
    if err != nil {
        HandleError(err)
    }
    err = bucket.PutObjectFromFileWithURL(signedURL, localFilename, options...)
    if err != nil {
        HandleError(err)
    }
}

```

 **Note** For more information about optional parameters, see [Manage object metadata](#).

- Use a signed URL to download an object

The following code provides an example on how to use a signed URL to download an object:

```
package main
import (
    "fmt"
    "os"
    "io/ioutil"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    localDownloadedFilename := "<yourDownloadedFilename>"
    // Specify the name of the bucket from which you want to delete the objects.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Download the object to a stream by using the signed URL.
    signedURL, err := bucket.SignURL(objectName, oss.HTTPGet, 60)
    if err != nil {
        HandleError(err)
    }
    body, err := bucket.GetObjectWithURL(signedURL)
    if err != nil {
        HandleError(err)
    }
    // Read the content of the object.
    data, err := ioutil.ReadAll(body)
    body.Close()
    data = data // use data
    // Download the object to a local device by using the signed URL.
    err = bucket.GetObjectToFileWithURL(signedURL, localDownloadedFilename)
    if err != nil {
        HandleError(err)
    }
}
```

1.4.5.11. IMG

OSS Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service that can process large amounts of data. After you upload source images to OSS, you can call RESTful API operations to process the images anytime, anywhere, and on any Internet device.

Basic features

OSS IMG provides the following features:

- Query image information.
- Convert image formats.
- Resize, crop, and rotate images.
- Apply image effect.
- Add image, text, and text-and-image watermarks to images.
- Customize image processing styles. Log on to the OSS console. Click a bucket name to go to the bucket details page. Click the Image Processing (IMG) tab and click **Create Rule** to customize an image processing style.
- Use cascade processing to call multiple IMG features.

Usage

You can use standard `HTTP GET` requests to access IMG. All processing parameters are encoded in the query string of a URL.

- Anonymous access

You can anonymously access a processed image by using a third-level domain in the following format:

`http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction>,<yourParamValue>`

Parameter	Description
bucket	The name of the bucket.
endpoint	The domain name used to access the region.
object	The name of the image object.
image	The identifier reserved by IMG.
action	The operations performed on images, such as scaling, cropping, and rotating.

Parameter	Description
param	The parameters corresponding to the operations performed on the images.

- Basic operations

For example, you can scale an image to a width of 100 and adjust the height based on the ratio:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customize an image style

You can anonymously access IMG by using a third-level domain in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- style: the identifier reserved by IMG for the custom style.
- yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when you create the custom style in the console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade processing

You can use cascade processing to perform multiple operations in sequence on an image by using a URL in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Access by using HTTPS

IMG supports access by using HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Authorized access

If the ACL of an image object is private, you must have permissions on the image object before you can perform operations on the image object.

Bucket ACL	Object ACL
private	default
Any ACL	private

The following code provides an example on how to generate a signed URL to access IMG:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    // Obtain the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    ossImageName := "<yourImageName>"
    signedURL, err := bucket.SignURL(ossImageName, oss.HTTPGet, 600, oss.Process("image/format,png"))
    if err != nil {
        HandleError(err)
    } else {
        fmt.Println(signedURL)
    }
}
```

- Access by using SDKs

You can use OSS SDKs to access and process image objects of any ACLs.

- Basic operations

The following code provides an example on how to perform basic IMG operations including obtaining image information, converting image formats, resizing, cropping, rotating, watermarking, and applying effects:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    // Obtain the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Resize the image.
    sourceImageName := "example.png"
    targetImageName := "example-resize.png"
    style := "image/resize,m_fixed,w_100,h_100"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
    // Crop the image.
    targetImageName = "example-crop.png"
    style = "image/crop,w_100,h_100,x_100,y_100,r_1"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
    // Rotate the image.
    targetImageName = "example-rotate.png"
    style = "image/rotate,90"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
    // Sharpen the image.
    targetImageName = "example-sharpen.png"
    style = "image/sharpen,100"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
    // Add watermarks to the image.
    targetImageName = "example-watermark.png"
    style = "image/watermark,text_SGVsbG8g5Zu-54mh5pyN5YqhIQ"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
    // Convert the image format.
    targetImageName = "example-formatconvert.jpg"
    style = "image/format,jpg"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
}
```

- Custom image styles

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    // Obtain the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Customize the image style.
    sourceImageName := "example.png"
    targetImageName := "example-custom.png"
    style := "style/<yourCustomStyleName>"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
}
```

- Cascade processing

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    // Obtain the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Customize the image style.
    sourceImageName := "example.png"
    targetImageName := "example-cascade.png"
    style := "image/resize,m_fixed,w_100,h_100/rotate,90"
    err = bucket.GetObjectToFile(sourceImageName, targetImageName, oss.Process(style))
    if err != nil {
        HandleError(err)
    }
}
```

Persistent storage of processed images

```
package main
import (
    "fmt"
    "os"
    "encoding/base64"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
}
```

```

// Obtain the bucket.
bucketName := "<yourBucketName>"
bucket, err := client.Bucket(bucketName)
if err != nil {
    HandleError(err)
}

// Scale the image and store the processed image. By default, the processed image is stored in the same bucket as the original image.
sourceImageName := "example.png"
targetImageName := "example-resize.png"
style := "image/resize,m_fixed,w_100,h_100"
process := fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)))
result, err := bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}

// Scale the image and store the processed image. The processed image is stored in the different bucket from the original image.
targetBucketName := "<yourTargetBucketName>"
targetImageName = "example-crop-different-bucket.png"
style = "image/resize,m_fixed,w_100,h_100"
process = fmt.Sprintf("%s|sys/saveas,o_%v,b_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)), base64.URLEncoding.EncodeToString([]byte(targetBucketName)))
result, err = bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}

// Crop the image and store the processed image.
targetImageName = "example-crop.png"
style = "image/crop,w_100,h_100,x_100,y_100,r_1"
process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)))
result, err = bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}

// Rotate the image and store the processed image.
targetImageName = "example-rotate.png"
style = "image/rotate,90"
process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)))
result, err = bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}

// Sharpen the image and store the processed image.
targetImageName = "example-sharpen.png"
style = "image/sharpen,100"
process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)))
result, err = bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}

// Add watermarks to the image and store the processed image.
targetImageName = "example-watermark.png"
style = "image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ"
process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)))
result, err = bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}

// Convert the image format and store the processed image.
targetImageName = "example-formatconvert.jpg"
style = "image/format,jpg"
process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targetImageName)))
result, err = bucket.ProcessObject(sourceImageName, process)
if err != nil {
    HandleError(err)
} else {
    fmt.Println(result)
}
}

```

1.4.5.12. Error handling

This topic describes how to handle errors when you use OSS SDK for Go.

Client errors

If errors occur when you use OSS SDK for Go, a unified operation used to handle errors is returned. The following code defines the error operation:

```
type error interface {
    Error() string
}
```

Other specific errors are further defined based on this error operation. The following code provides an example of the operation used to handle an HTTP request error:

```
//net.Error
type Error interface {
    error
    Timeout() bool // Is the error a timeout
    Temporary() bool // Is the error temporary
}
```

Server errors

If request errors occur when you use OSS SDK for Go, corresponding errors are returned. In case of HTTP request errors and I/O errors, errors defined in OSS SDK for Go are returned. If an error occurs when the OSS server processes a request, the following error is returned, which is an implementation of the error operation:

```
type ServiceError struct {
    Code      string // The error code returned by OSS.
    Message   string // The error details provided by OSS.
    RequestId string // The UUID used to identify the request.
    HostId    string // The host ID used to identify the accessed OSS cluster.
    StatusCode int    // The HTTP status code.
}
```

If the HTTP status code returned by OSS is not as expected, the following error is returned, which is an implementation of the error operation:

```
type UnexpectedStatusCodeError struct {
    allowed []int // The expected HTTP status code returned by OSS.
    got     int  // The actual HTTP status code returned by OSS.
}
```

Example

The following code provides an example on how to handle errors:

```
package main
import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
// The function used to handle the error.
func HandleError(err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>")
    if err != nil {
        HandleError(err)
    }
    bucketName := "<yourBucketName>"
    objectName := "<yourObjectName>"
    // Specify the name of the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        HandleError(err)
    }
    // Use simple upload to upload strings.
    err = bucket.PutObject(objectName, strings.NewReader("yourObjectValueContentString"))
    if err != nil {
        HandleError(err)
    }
}
```

OSS error codes

The following table describes the OSS error codes.

Error code	Error message	HTTP status code
AccessDenied	The error message returned because access is denied.	403
BucketAlreadyExists	The error message returned because the bucket already exists.	409

Error code	Error message	HTTP status code
BucketNotEmpty	The error message returned because the bucket is not empty.	409
EntityTooLarge	The error message returned because the entity exceeds the maximum allowed size.	400
EntityTooSmall	The error message returned because the entity is smaller than the minimum allowed size.	400
FileGroupTooLarge	The error message returned because the object group exceeds the maximum allowed size.	400
InvalidLinkName	The error message returned because the symbolic link points to an object that has the same name as the symbolic link.	400
LinkPartNotExist	The error message returned because the symbolic link points to an object that does not exist.	400
ObjectLinkTooLarge	The error message returned because the number of objects the symbolic link points to exceeds the limit.	400
FieldItemTooLong	The error message returned because the total length of the form fields in the POST request exceeds the limit.	400
FilePartIntegrity	The error message returned because the part is modified.	400
FilePartNotExist	The error message returned because the specified part does not exist.	400
FilePartStale	The error message returned because the part expires.	400
IncorrectNumberOfFilesInPOSTRequest	The error message returned because the number of objects in the POST request is invalid.	400
InvalidArgument	The error message returned because the parameter format is invalid.	400
InvalidAccessKeyId	The error message returned because the specified Accesskey ID does not exist.	403
InvalidBucketName	The error message returned because the bucket name is invalid.	400
InvalidDigest	The error message returned because the digest is invalid.	400
InvalidEncryptionAlgorithmError	The error message returned because the specified encryption algorithm based on entropy encoding is invalid.	400
InvalidObjectName	The error message returned because the object name is invalid.	400
InvalidPart	The error message returned because the part is invalid.	400
InvalidPartOrder	The error message returned because the part sequence is invalid.	400
InvalidPolicyDocument	The error message returned because the policy document is invalid.	400
InvalidTargetBucketForLogging	The error message returned because the destination bucket specified for logging is invalid.	400
InternalError	The error message returned when an internal OSS error occurs.	500
MalformedXML	The error message returned because the XML format is invalid.	400
MalformedPOSTRequest	The error message returned because the format of the POST request body is invalid.	400
MaxPOSTPreDataLengthExceededError	The error message returned because the size of the POST request body, excluding the object content to upload, exceeds the limit.	400
MethodNotAllowed	The error message returned because the method is not supported.	405
MissingArgument	The error message returned because the required parameters are not set.	411

Error code	Error message	HTTP status code
MissingContentLength	The error message returned because the content length is not specified.	411
NoSuchBucket	The error message returned because the specified bucket does not exist.	404
NoSuchKey	The error message returned because the specified object does not exist.	404
NoSuchUpload	The error message returned because the specified multipart upload ID does not exist.	404
NotImplemented	The error message returned because the method cannot be implemented.	501
PreconditionFailed	The error message returned because a precondition error occurs.	412
RequestTimeTooSkewed	The error message returned because the difference between the time on the client when the request is sent and the time on the server when the request is received is greater than 15 minutes.	403
RequestTimeout	The error message returned because the request times out.	400
RequestIsNotMultiPartContent	The error message returned because the content-type value specified in the POST request is invalid.	400
SignatureDoesNotMatch	The error message returned because the signature is invalid.	403
TooManyBuckets	The error message returned because the number of the buckets managed by a user exceeds the limit.	400
InvalidEncryptionAlgorithmError	The error message returned because the specified encryption algorithm based on entropy encoding is invalid.	400

Note In the preceding table, the error code is specified by the `OssServiceError.Code` parameter. The HTTP status code is specified by the `OssServiceError.StatusCode` parameter.

1.5. Obtain an endpoint

You can obtain an endpoint of a service in the Apsara Uni-manager Operations Console.

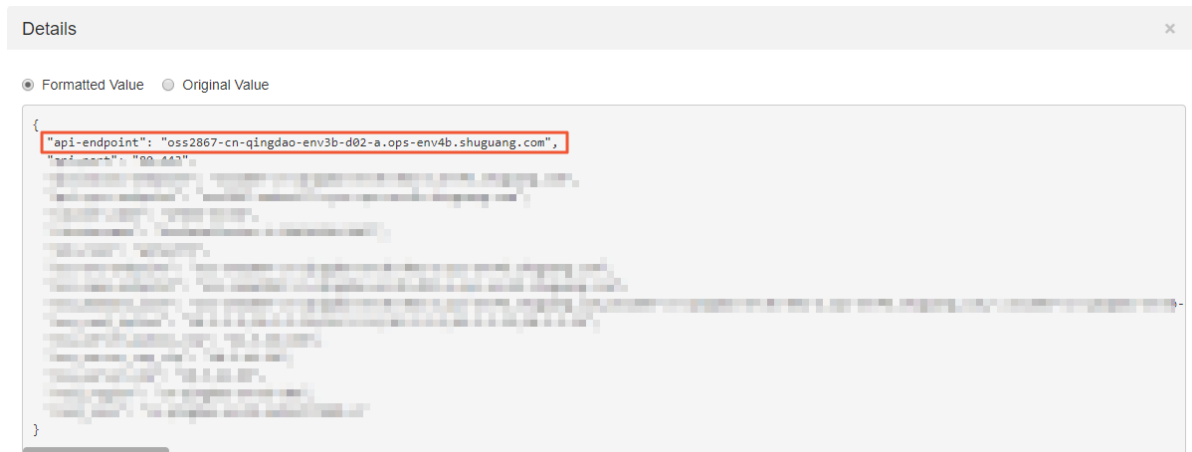
Procedure

- Log on to the Apsara Uni-manager Operations console.
For more information about how to log on to the Apsara Uni-manager Operations console, see *Log on to the Apsara Uni-manager Operations console* in *Operation and Maintenance Guide*.
- In the left-side navigation pane, choose **Products > Product List** to go to the Product List page. In the Apsara Stack O&M section, click **Apsara Infrastructure Management Framework**.
- Click **Reports**.
- In the list, click **Registration Vars of Services**.
- In the top navigation bar, choose **Reports > System Reports**.
- On the **Registration Vars of Services** page, find the **oss-server** row. Right-click the **Service Registration** column and choose **Show More** from the short cut menu.

Registration Vars of Services			
Service	Service Registration	Cluster	Update Time
oss	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss2	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss3	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss4	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss5	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss6	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss7	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss8	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss9	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss10	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss11	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss12	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss13	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss14	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss15	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss16	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss17	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss18	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss19	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss20	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss21	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss22	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss23	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss24	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss25	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss26	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss27	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss28	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss29	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss30	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss31	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss32	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss33	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss34	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss35	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss36	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss37	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss38	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss39	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss40	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss41	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss42	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss43	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss44	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss45	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss46	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss47	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss48	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss49	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss50	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss51	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss52	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss53	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss54	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss55	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss56	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss57	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss58	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss59	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss60	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss61	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss62	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss63	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss64	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss65	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss66	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss67	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss68	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss69	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss70	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss71	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss72	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss73	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss74	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss75	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss76	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss77	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss78	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss79	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss80	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss81	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss82	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss83	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss84	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss85	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss86	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss87	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss88	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss89	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss90	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss91	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss92	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss93	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss94	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss95	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss96	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss97	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss98	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss99	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49
oss100	endpoint: oss-cn-qingdao.aliyuncs.com	OssSataCluster-A-20191030-2867	02/14/20, 08:12:49

- Click **Formatted Value**.

The value of **api-endpoint** is the OSS endpoint.



1.6. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. AccessKey pairs are used to implement symmetric encryption to verify the identity of requesters. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt a signature string. This topic describes how to obtain an AccessKey pair.

Prerequisites

Only operations administrators or level-1 organization administrators can obtain the AccessKey pair of an organization.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following operations:

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

Note When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Login**.
4. In the top navigation bar, click **Enterprise**.
5. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
6. In the organization navigation tree, click an organization name.
7. In the Current Organization section, click **Obtain an accesskey**.
8. In the AccessKey message, view the AccessKey pair of the organization.

Note An AccessKey pair is automatically allocated to each level-1 organization. Subordinate organizations use the same AccessKey pair of their level-1 organization.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following steps:

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

Note When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Login**.
4. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
5. In the **Apsara Stack AccessKey Pair** section, view your AccessKey pair.

Apsara Stack AccessKey Pair You must use the AccessKey pair when you access Apsara Stack resources.

The AccessKey pair including the AccessKey ID and AccessKey secret is the credential to for you to use Apsara Stack resources with full permissions. You must keep the AccessKey pair confidential.

Region	AccessKey ID	AccessKey Secret
cn-qingdao-env4b-d01		Show

Note The AccessKey pair consists of an AccessKey ID and an AccessKey secret. AccessKey pairs of accounts grant you full permissions to access Apsara Stack resources. You must keep your AccessKey pair confidential.

2. NAS

2.1. Preparations

2.1.1. Log on to the API Tool console

You can log on to the API Tool console from the Apsara Uni-manager Management Console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment personnel.
- A browser is available. We recommend that you use Google Chrome.

Procedure

1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press Enter.
2. Enter your username and password.

Obtain the username and password that are used to log on to the Apsara Uni-manager Management Console from the operations administrator.

Note When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your account. For security purposes, your password must meet the minimum complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA verification code and click **Authenticate**.

Note For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Others > API Tool**.

2.1.2. View the information about API operations

This topic describes how to view the information about API operations. The information includes service names, API operation names, and API versions. The information is required when you use an SDK to call API operations.

Context

The API operations on the API Directory page are only for reference. For information about how to call API operations, see developer guides of the corresponding cloud services. If the information on the API Directory page is inconsistent with that in the developer guide of the corresponding cloud service, the information in the developer guide prevails.

Procedure

1. Go to the API Tool console. For more information, see [Log on to the API Tool console](#).
2. In the left-side navigation pane, click **API Directory**.
3. On the **API Directory** page, select the service that you want to use from the drop-down list, enter a keyword of an API operation name in the search box, and then click Search.

API Directory		
All Products ▾	cre	
Product Name	Operation Name	API Version
cr	CreateMetadataOccurrence	2018-12-01
cr	CreateUserInfo	2016-06-07
cr	CreateNamespaceAuthorization	2016-06-07
cr	CreateNamespace	2016-06-07

4. View and record the service name, API operation name, and API version of the API operation that you want to call.

2.1.3. Obtain an AccessKey pair

AccessKey credentials are assigned by Resource Access Management (RAM) or Security Token Service (STS). You can use one type of AccessKey credentials for authentication when you call API operations. This topic describes how to obtain an AccessKey pair.

Context

An AccessKey ID and an AccessKey secret are used to implement symmetric encryption and verify the identity of request senders. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt a signature string.

AccessKey credentials can be assigned by RAM or STS:

- An AccessKey pair that is assigned by RAM to a third party consists of an AccessKey ID and an AccessKey secret.
- An AccessKey 3-tuple that is assigned by STS to a third party consists of an AccessKey ID, an AccessKey secret, and a security token. For more information, see [Obtain an STS AccessKey 3-tuple](#).

The Apsara Uni-manager Management Console provides AccessKey pairs for personal accounts and organizations. Only operations administrators and the administrators of level-1 organizations can obtain the AccessKey pairs of organizations. We recommend that you use the AccessKey pairs of personal accounts to call the operations provided by the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, configure the request headers that are described in the following table for access control. Otherwise, an error that indicates insufficient permissions may occur.

 **Warning** The AccessKey pair of a personal account is managed by the permission system of the Apsara Uni-manager Management Console. If you need to obtain the AccessKey pair of an organization, make sure that your operation is allowed by an administrator. This ensures security.

Parameter	Description
x-acs-regionid	The region ID, for example, cn-hangzhou-*
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For information about how to obtain an organization ID, see the "GetOrganizationList" section in Apsara Uni-manager Management Console Developer Guide.
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For information about how to obtain a resource set ID, see the "ListResourceGroup" section in Apsara Uni-manager Management Console Developer Guide.
x-acs-instanceid	The ID of the instance on which an operation is performed. You can obtain the ID of a NAS file system from the NAS file system list.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **Personal Information**.
3. In the **AccessKey Pair** section, view the AccessKey pair of your account.

AccessKey Create AccessKey Pair				
 The AccessKey ID and AccessKey Secret are the keys used to access the cloud service resources. They have full permissions on the account. Please keep them properly.				
AccessKey ID	AccessKey Secret	Status	Created At	Actions
W1YRFvK...	*****	 Enable	Sep 18, 2021, 11:05:57	Disable Delete
IsJsAUyX...	*****	 Enable	Sep 10, 2021, 12:28:35	Disable Delete

 **Note** An AccessKey pair consists of an AccessKey ID and an AccessKey secret. These credentials provide full permissions on cloud resources within your account. You must keep the AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization list, click the name of the level-1 organization whose AccessKey pair you want to obtain.
5. Click **Management Accesskey**.
6. In the dialog box that appears, view the AccessKey pair of the organization.

2.1.4. Obtain an STS AccessKey 3-tuple

This topic describes how to use Security Token Service (STS) to call Apsara Stack API operations.

Background information

STS allows you to manage temporary credentials to your Apsara Stack resources. You can use STS to call Apsara Stack API operations.

An STS AccessKey 3-tuple consists of an AccessKey ID, an AccessKey secret, and a security token. When you use STS to call API operations, you must specify an STS AccessKey 3-tuple. For more information about STS, see [What is STS?](#)

Procedure

1. Obtain the `RAM role`.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. Move the pointer over the profile picture in the upper-right corner of the page and click **Personal Information**.
 - iii. Click **View Current Role Policy**.

In the **View Policies of Current Role** dialog box, view your `RAM role`.

2. Obtain the STS AccessKey 3-tuple.

Note

In the following sample code, replace the variables with the actual values. Delete the angle brackets <> in the final code.

```
public static void main(String[] args) {
    // Create and initialize a DefaultAcsClient instance.
    // Specify the region ID, AccessKey ID, and AccessKey secret.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKeyId>", "<yourAccessKeySecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    String assumeRole = callAssumeRole(client);
    if (StringUtils.isEmpty(assumeRole)) return;
    JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
    JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
    // Obtain the STS AccessKey 3-tuple.
    String stsAK = credentials.getString("AccessKeyId");
    String stsSK = credentials.getString("AccessKeySecret");
    String stsToken = credentials.getString("SecurityToken");
}

private static String callAssumeRole(IAcsClient client) {
    CommonRequest request = new CommonRequest();
    // Replace sts.aliyuns.com with the endpoint of STS. The endpoint of STS is in the sts-vpc.${global:region}.${global:internet-domain} format.

    request.setSysDomain("sts.aliyuns.com");
    request.setSysProduct("Sts");
    request.setSysAction("AssumeRole");
    request.setSysVersion("2015-04-01");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace <yourRoleSessionName> with the name of the current session.
    request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
    // Replace "<yourRoleArn>" with the actual ARN that you obtained for the RAM role in the preceding step.
    request.putQueryParameter("RoleArn", "<yourRoleArn>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        return response.getData();
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
    return null;
}
```

3. Use the STS AccessKey 3-tuple to call an API operation of a cloud service. The following sample code provides an example on how to call the `DescribeRegions` operation of Apsara File Storage NAS.

Note

In the following sample code, replace the variables with the actual values. Delete the angle brackets <> in the final code.

```

public static void main(String[] args) {
    // Replace "<yourRegionID>" with the actual region ID.
    // Replace "<yourStsAccessKeyID>" and "<yourStsAccessKeySecret>" with the AccessKey ID and AccessKey secret of the 3-tuple obtained in the preceding step.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourStsAccesskey>", "<yourStsAccessSecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    CommonRequest request = new CommonRequest();
    request.setSysDomain("nas.aliyuns.com");
    request.setSysProduct("NAS");
    request.setSysAction("DescribeRegions");
    request.setSysVersion("2017-06-26");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace "<yourRegionID>" with the actual region ID.
    request.putHeadParameter("x-acs-regionid", "<yourRegionID>");
    // Replace "<yourStsToken>" with the security token of the 3-tuple obtained in the preceding step.
    request.putQueryParameter("SecurityToken", "<yourStsToken>");
    // Optional. Replace "<yourResourceGroupID>" with the actual resource set ID.
    // request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");
    // Required. Replace "<yourOrganizationID>" with the actual organization ID.
    request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        System.out.println(response.getData());
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
}

```

2.1.5. Obtain the values of common header parameters

You must provide multiple header parameters when you call Apsara Stack API operations. This topic describes the header parameters and how to obtain the values of the parameters. This topic also describes the return values of some parameters after you call API operations.

Header parameters

Parameter	Description
x-acs-regionid	The ID of the region. For more information about how to obtain a region ID, see Obtain a region ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain an organization ID, see Obtain an organization ID .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain a resource set ID, see Obtain a resource set ID .
x-acs-instanceid	The ID of the NAS file system on which you want to perform operations. For more information about how to obtain a file system ID, see Obtain a file system ID .

Obtain a region ID

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Regions**.
4. Click the name of the organization whose region ID you want to view.
5. View the region ID in the **Regions** section.

Obtain an organization ID

You can call the `GetOrganizationList` operation to obtain the organization ID. For more information, see *GetOrganizationList* in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a resource set ID

You can call the `ListResourceGroup` operation to obtain the resource set ID. For more information, see *ListResourceGroup* in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a file system ID

You can obtain the ID of a NAS file system in the Apsara Uni-manager Management Console. Perform the following operations to obtain a file system ID:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Storage > NAS**.
3. In the left-side navigation pane, click **File System List**.
4. On the **File System List** page, view the ID of the NAS file system.

Response parameters

The following table describes the additional response parameters returned by default when you call operations to query the instances of cloud services.

Parameter	Description
ResourceGroupName	The name of the resource set.
ResourceGroup	The ID of the resource set.
DepartmentName	The name of the organization.
Department	The ID of the organization.

2.1.6. Obtain the endpoint of NAS

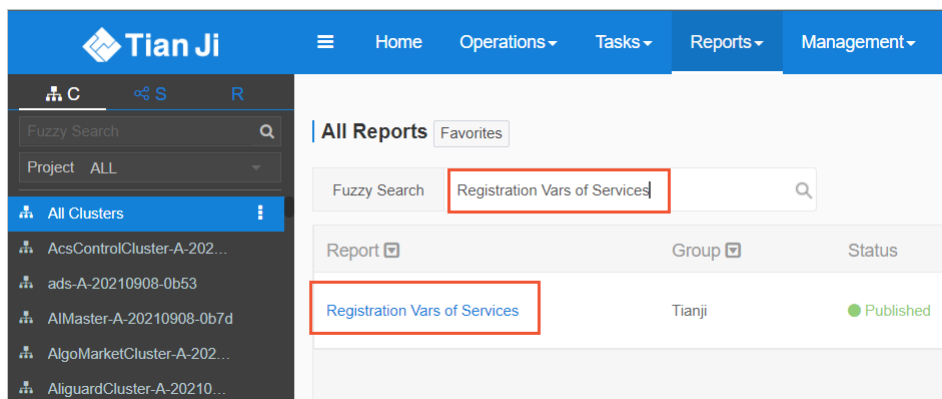
This topic describes how to obtain the endpoint of Apsara File Storage NAS.

Background information

When you call Apsara Stack API operations by using the Portable OpenAPI Proxy (POP) gateway, you must obtain the endpoint of your cloud service. You can obtain an endpoint from onsite Apsara Stack O&M engineers. You can also perform the following operations to obtain an endpoint.

Procedure

1. Log on to the Apsara Infrastructure Management console.
 - i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, choose **Products > Platforms > Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.
3. On the **All Reports** page, search for `Registration Vars of Services`. In the report list that appears, click the name of the report.



4. On the **Registration Vars of Services** page, click the  icon next to **Service** and search for `NAS`.
5. Find `nas-yaochi`. Right-click the value in the **Service Registration** column and select **Show More**.

On the **Details** page, view the `nas.openapi.endpoint` value, which is the `endpoint`.

2.1.7. Obtain NAS SDKs

When you call Apsara Stack API operations by using the Portable OpenAPI Proxy (POP) gateway, you must use an SDK of your cloud service. This topic describes how to obtain Apsara File Storage NAS SDKs.

An SDK includes code and examples. You can create cloud applications by using an SDK in the programming language that you require.

The SDKs for Java, Python, and Go are provided. You can obtain the SDKs from [Apsara Stack SDKs](#) based on your business requirements.

Note

We recommend that you use an SDK for Java or Python. If you use SDKs in other programming languages, the SDKs may be incompatible with your system.

We recommend that you use the SDK core library that is provided on the Alibaba Cloud official website. The SDK core library is shared by all cloud services and is compatible with Apsara Stack.

The following table provides links to the references and GitHub source code of NAS SDKs for different programming languages.

Programming language	GitHub source code
Java	NAS SDK for Java
Python	NAS SDK for Python

2.2. Call examples for SDK for Java

2.2.1. Quick start

This topic describes how to obtain and use Apsara File Storage NAS SDK for Java.

Prerequisites

- An authorized account and an AccessKey pair are obtained. The AccessKey pair consists of an AccessKey ID and an AccessKey secret. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of NAS is obtained. For more information, see [Obtain the endpoint of NAS](#).
- Apsara Stack SDK for Java 1.8 or later is installed.

Install NAS SDK for Java

If you use Apache Maven to manage a Java project, you need to only import the JAR file of Apsara Stack SDK for Java to the pom.xml file of the project.

Note

To use a NAS SDK, you must install the core library of Apsara Stack SDK for Java and NAS SDK for Java. We recommend that you use the core library V4.5.17 or later, and use NAS SDK for Java V2.0.2.

Use NAS SDK for Java

To use NAS SDK for Java, perform the following steps:

1. Create and initialize a DefaultAcsClient instance.
2. Create an API request and configure the required parameters.
3. Initiate the API request and handle the response or exceptions.

Note

In the following sample code, replace the variables with the actual values. Delete the angle brackets <> in the final code.

```

import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.nas.model.v20170626.DescribeFileSystemsRequest;
import com.aliyuncs.nas.model.v20170626.DescribeFileSystemsResponse;
public class Main {
    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace the parameter with the region ID.
            "<yourAccessKeyID>", // Replace the parameter with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace the parameter with your AccessKey secret.
        // Specify the endpoint of the service. Replace the variables with the actual values.
        profile.addEndpoint("<yourRegionID>", "<Product>", "<Endpoint>");
        // Configure the request parameters.
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setIgnoreSSLCerts(true); // Ignore the certificate.
        clientConfig.setReadTimeoutMillis(<readTimeout>); // Optional. Specify the read timeout period. Unit: milliseconds.
        clientConfig.setConnectionTimeoutMillis(<connectionTimeout>); // Optional. Specify the connection timeout period. Unit: milliseconds.
        clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>); // Optional. Specify the maximum number of requests for each host.
        clientConfig.setMaxRequests(<maxRequests>); // Optional. Specify the maximum number of requests.
        clientConfig.setProtocolType(ProtocolType.HTTP); // Optional. Specify the request protocol. Default value: HTTP.
        /* Optional. Specify the IP address and port of the proxy.
        clientConfig.setHttpProxy("http://192.168.XX.XX:33XX");
        or
        clientConfig.setHttpsProxy("https://192.168.XX.XX:33XX");
        */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the parameters.
        DescribeFileSystemsRequest request = new DescribeFileSystemsRequest();
        // Business parameters.
        request.setPageSize(10);
        request.setPageNumber(1);
        // The header parameters that are used for resource isolation. Replace the values with your actual organization ID and resource set ID.
        request.putHeadParameter("x-acs-organizationid", "1");
        request.putHeadParameter("x-acs-resourcegroupid", "11");
        // Initiate the request and handle the response or exceptions.
        DescribeFileSystemsResponse response;
        try {
            response = client.getAcsResponse(request);
            for (DescribeFileSystemsResponse.FileSystem fileSystem:rsp.getFileSystems()) {
                System.out.println(fileSystem.getFileSystemId());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}

```

2.2.2. Use SDK

2.2.2.1. Install NAS SDK for Java

This topic describes how to install NAS SDK for Java.

Apsara Stack SDK for Java supports Java Development Kit (JDK) 1.8 or later. You can install Apsara Stack SDK for Java by adding Maven dependencies or downloading the SDK. For more information, see [Alibaba Cloud SDKs](#).

To use a NAS SDK, you must install the core library of Apsara Stack SDK for Java and NAS SDK for Java. We recommend that you use the core library V4.5.17 or later, and use NAS SDK for Java V2.0.2.

To use NAS SDK for Java, you must install the core library of Apsara Stack SDK for Java and add the following dependencies to the pom.xml file. For more information about the specific SDK version, see [Alibaba Cloud SDKs](#).

```

<!-- Alibaba Cloud SDK for Java -->
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.17</version>
</dependency>
<!-- NAS Java SDK -->
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-nas</artifactId>
    <version>2.0.2</version>
</dependency>

```

2.2.2.2. Configure authentication credentials

When you use Alibaba Cloud SDKs to access Apsara Stack services, you must provide an Apsara Stack tenant account for authentication. This topic describes how to configure authentication credentials.

Apsara Stack SDK for Java supports authentication based on AccessKey pairs. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

If you want to use your AccessKey pair to access Apsara Stack, you must specify the AccessKey pair as the credentials when you initialize the client.

Note

Do not disclose any code that contains your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.

```
DefaultProfile profile = DefaultProfile.getProfile(
    "<yourRegionID>",           // Replace the parameter with the region ID.
    "<yourAccessKeyID>",        // Replace the parameter with your AccessKey ID.
    "<yourAccessKeySecret>"); // Replace the parameter with your AccessKey secret.
```

Note

In the preceding sample code, replace the variables with the actual values. Delete the angle brackets <> in the final code.

2.2.2.3. Configure connection requests

This topic describes how to configure a connection request by using the HttpClientConfig object.

You can use the following code to configure the HttpClientConfig object. We recommend that you select and configure the following parameters based on your business requirements.

Note

You must use actual values to replace the variables in the following sample code. Do not retain the angle brackets (<>) in the final code.

The following list describes the variables:

- <readTimeout>: Specify the read timeout period based on your business requirements. Unit: milliseconds.
- <connectionTimeout>: Specify the connection timeout period based on your business requirements. Unit: milliseconds.
- <maxRequestsPerHost>: Specify the maximum number of requests that are supported by each server based on your business requirements. The supported data type is INTEGER.
- <maxRequests>: Specify the maximum number of requests based on your business requirements. The supported data type is INTEGER.
- <http://192.168.XX.XX:33XX>: Replace it with the actual IP address and port of the proxy.

```
// Obtain the HttpClientConfig object.
HttpClientConfig clientConfig = HttpClientConfig.getDefault();
// Specify whether to ignore certificates.
clientConfig.setIgnoreSSLCerts(true);
// (Optional) Specify the read timeout period. Unit: milliseconds.
clientConfig.setReadTimeoutMillis(<readTimeout>);
// (Optional) Specify the connection timeout period. Unit: milliseconds.
clientConfig.setConnectionTimeoutMillis(<connectionTimeout>);
// (Optional) Specify the maximum number of requests that are supported by each server. The supported data type is INTEGER.
clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>);
// (Optional) Specify the maximum number of requests. The supported data type is INTEGER.
clientConfig.setMaxRequests(<maxRequests>);
// (Optional) Specify the protocol type.
clientConfig.setProtocolType(ProtocolType.HTTP);
/** (Optional) Specify whether to configure the proxy based on your business requirements. The following code is available:
clientConfig.setHttpProxy("<http://192.168.XX.XX:33XX>");
Or
clientConfig.setHttpsProxy("<https://192.168.XX.XX:33XX>");
Replace https://192.168.XX.XX:33XX with the actual IP address and port of the proxy.
*/
profile.setHttpClientConfig(clientConfig);
```

2.2.2.4. Initiate a request

This topic describes how to use the SDK for Java to initiate a request.

Procedure

1. Create an AcsClient instance.

```
IAcsClient client = new DefaultAcsClient(profile);
```

2. Create a request.

Name the request in the `${apiName}Request` format. Replace `${apiName}` with the name of the API operation, for example, `DescribeFileSystems`.

When multiple service SDKs are used, different requests may have the same name. You can differentiate the requests based on package names.

```
DescribeFileSystemsRequest request = new DescribeFileSystemsRequest();
// Business parameters.
request.setFileSystemType("all");
request.setPageSize(10);
// The header parameters that are used for resource isolation. Replace the values with your actual organization ID and resource set ID.
request.putHeadParameter("x-acs-organizationid", "1");
request.putHeadParameter("x-acs-resourcegroupid", "");
```

3. Initiate the request and handle the response.

```
try {
    DescribeFileSystemsResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    System.out.println("ErrCode:" + e.getErrCode());
    System.out.println("ErrMsg:" + e.getErrMsg());
    System.out.println("RequestId:" + e.getRequestId());
}
```

In normal cases, all returned fields are deserialized into the response. You can call `response.getXXX()` to obtain the values of the response fields.

```
instanceStatus = response.getStatus();
```

However, if an exception occurs or you need to obtain the original HTTP response, call `doAction()` to obtain the original response.

```
HttpResponse response = client.doAction(request);
```

2.2.2.5. Handle errors

If the server or SDK encounters an error when you use the SDK for Java, the SDK returns an error response. The error response contains an error code and an error message.

Among the errors returned by the SDK for Java, you do not need to handle the SDK errors, but must solve the server errors.

- `ServerException` indicates a server error.
- `ClientException` indicates an SDK error.

For example, if the following error occurs, you can modify the AccessKey ID based on the error message:

```
com.aliyuncs.exceptions.ClientException: InvalidAccessKeyId.NotFound : Specified access key is not found.
```

If you want to handle the SDK errors returned by the SDK, refer to the following example to write your code:

```
try {
    FooResponse response = client.getAcsResponse(request);
    // Handle the response.
} catch (ServerException e) {
    // You can add your error handling logic here.
    // For example, display the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
    // If you are unable to fix the error, submit a ticket and provide RequestId to Alibaba Cloud technical support personnel.
    System.out.println("ResponseId=" + e.getRequestId());
} catch (ClientException e) {
    // You can add your error handling logic here.
    // For example, display the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
}
```

2.2.2.6. Use CommonRequest to call API operations

If you want to call the API operations of a cloud service that does not provide SDKs, or you cannot use an SDK to call API operations whose requests and responses meet your business requirements, you can use CommonRequest to call API operations. You can use CommonRequest to call all API operations.

Benefits

CommonRequest provides the following benefits:

- Lightweight: You only need to download the core package. You do not need to download and install SDKs of Apsara Stack services.
- Easy to use: You can call the newly released API operations of a service without the need to update the SDK for the service.
- Fast iteration.

Use CommonRequest to call an API operation

If you want to send a CommonRequest request, you must obtain the values of the following parameters. You can obtain the values of these parameters from the API reference of Apsara File Storage NAS.

- Domain: the endpoint of NAS.
- Version: the version number of the API, in the YYYY-MM-DD format.
You can obtain the API version in the [Common parameters](#) topic of the API reference.
- Operation information: the name of the API operation that you want to call.

You need to obtain the value of the Action parameter and specify an API operation in `request.setSysAction("<Action>")`.

For example, if you want to call the `CreateFileSystem` operation, you must use `request.setSysAction("CreateFileSystem")` to specify the API operation when you initiate a `CommonRequest` request.

Example: Call an RPC API operation

This example shows how to call the `DescribeFileSystems` operation to query file systems.

Note In the following code, replace the variables with actual values. Delete the angle brackets <> in the final code.

You need to configure the following variables:

- <yourRegionID>: Replace the parameter with the region ID.
- <yourAccessKeyID>: Replace the parameter with your AccessKey ID.
- <yourAccessKeySecret>: Replace the parameter with your AccessKey secret.
- <yourSecurityToken>: Replace the parameter with your security token.
- <yourResourceGroupID>: Replace the parameter with the resource set ID.
- <yourOrganizationID>: Replace the parameter with the organization ID.
- <yourInstanceID>: Replace the parameter with the instance ID.
- <http://192.168.XX.XX:33XX>: Replace the parameter with the IP address and port of the proxy.

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
class DescribeFileSystems {
    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace the parameter with the region ID.
            "<yourAccessKeyID>", // Replace the parameter with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace the parameter with your AccessKey secret.
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setProtocolType(ProtocolType.HTTP);
        /**
         * Optional. Configure the HTTP proxy based on your business requirements.
         * Format:
         * clientConfig.setHttpProxy("http://192.168.XX.XX:33XX");
         * Specify the IP address and port of the proxy.
         */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the parameters.
        CommonRequest request = new CommonRequest();
        // Specify the endpoint of the service.
        request.setSysDomain("nas.cn-hangzhou.com");
        request.setSysProduct("NAS");
        request.setSysVersion("2017-06-26");
        request.setSysAction("DescribeFileSystems");
        request.putQueryParameter("PageNumber", "1");
        request.putQueryParameter("PageSize", "10");
        request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");// Optional. Specify the resource set ID.
        request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");// Required. Specify the organization ID.
        request.putHeadParameter("x-acs-caller-sdk-source", "ApiTestDemo");// Required. Specify the source of the API call.
        request.putHeadParameter("x-acs-regionid", "<yourRegionID>");// Optional. Specify the region ID. If you use Apsara Stack Enterprise V3.15.0 or earlier, this parameter is required.
        request.putHeadParameter("x-acs-instanceid", "<yourInstanceID>");// Optional Specify the instance ID. If you need to delete, modify, or query an instance, this parameter is required.
        /**
         * When you use an STS AccessKey 3-tuple to call an API operation, you must add the SecurityToken field in the request parameters.
         * Format:
         * request.putQueryParameter("SecurityToken", "<yourSecurityToken>");
         * Specify your security token.
         */
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

2.3. Call examples for SDK for Python

2.3.1. Quick start

This topic describes how to obtain and use Apsara File Storage NAS SDK for Python.

Prerequisites

- An authorized account and an AccessKey pair (AccessKey ID and AccessKey secret) are obtained. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of NAS is obtained. For more information, see [Obtain the endpoint of NAS](#).

Install the SDK for Python

1. Install the SDK core library.

- If you use Python 2.x, run the following command to install the SDK core library:

```
pip install aliyun-python-sdk-core
```

- If you use Python 3.x, run the following command to install the SDK core library:

```
pip3 install aliyun-python-sdk-core-v3
```

2. Install NAS SDK.

- If you use Python 2.x, run the following command to install NAS SDK:

```
pip install aliyun-python-sdk-nas
```

- If you use Python 3.x, run the following command to install NAS SDK:

```
pip3 install aliyun-python-sdk-nas
```

Use the SDK for Python

To use the SDK for Python, perform the following steps:

1. Create a client instance. When you create a client instance, you must obtain a region ID, AccessKey ID, and AccessKey secret.
2. Create an API request and configure the required parameters.
3. Initiate the API request and handle the response or exception.

```
#!/usr/bin/env python
# coding=utf-8
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdknas.request.v20170626 import DescribeFileSystemsRequest
# global config
region_id = '<yourRegionID>' # Specify the region ID.
access_key = '<yourAccessKeyID>' # Specify the AccessKey ID.
access_secret = '<yourAccessKeySecret>' # Specify the AccessKey secret.
# product config
product_name = 'NAS' # Specify the service name.
endpoint = 'nas.aliyuns.com' # Specify endpoint of the service.
# create client
region_provider.modify_point(product_name, region_id, endpoint)
clt = client.AcsClient(access_key, access_secret, region_id)
# setup request
request = DescribeFileSystemsRequest.DescribeFileSystemsRequest()
request.set_accept_format('json') # Specify the response format.
# get response
response = clt.do_action_with_exception(request)
print(response)
```

2.3.2. Use SDK

2.3.2.1. Install the SDK for Python

This topic describes how to install Alibaba Cloud SDK for Python and Apsara File Storage NAS SDK for Python.

1. Install Alibaba Cloud SDK for Python.

Apsara Stack SDK for Python supports Python 2.6.x, Python 2.7.x, and Python 3.x or later. You can install Apsara Stack SDK for Python by using pip.

 **Note** To use the SDK of an Apsara Stack service, you must install the core library of Apsara Stack SDK for Python. For example, if you want to use NAS SDK for Python, you must install the SDK core library for Python and NAS SDK for Python.

- If you use Python 2.x, run the following command to install the SDK core library:

```
pip install aliyun-python-sdk-core
```

- If you use Python 3.x, run the following command to install the SDK core library:

```
pip3 install aliyun-python-sdk-core-v3
```

2. Install NAS SDK for Python.

- If you use Python 2.x, run the following command to install NAS SDK for Python:

```
pip install aliyun-python-sdk-nas
```

- If you use Python 3.x, run the following command to install NAS SDK for Python:

```
pip3 install aliyun-python-sdk-nas
```

2.3.2.2. Configure authentication credentials

When you use an Apsara Stack SDK to access Apsara Stack services, you must provide an authorized account for authentication.

NAS SDK for Python supports authentication based on an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

If you want to use your AccessKey pair to access Apsara Stack, you must configure the AccessKey pair as the credentials when you initialize the client.

 **Notice** Do not disclose any code containing your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.

```
clt = client.AcsClient(access_key, access_secret, region_id)
```

2.3.2.3. Set HTTPS requests

Apsara Stack SDK for Python allows you to initiate API requests over HTTP and HTTPS.

You can specify the HTTP or HTTPS protocol for a specific request, or set a global default protocol.

 **Note** The default HTTP or HTTPS protocol of a service takes precedence over the global default protocol.

Add OpenSSL support

The HTTPS protocol support of Apsara Stack SDK for Python is based on the OpenSSL support in Python. To use Apsara Stack SDK for Python to send HTTPS requests, you must add OpenSSL support in Python. OpenSSL supports TLS 1.2 and later.

Run the `python -c "import ssl"` command to check whether your Python environment supports OpenSSL. If the error message

```
ImportError: No module named ssl
```

 is not returned, OpenSSL is already supported.

Otherwise, run the following command to install it:

```
pip install pyopenssl
```

Set the protocol to HTTP or HTTPS for a single request

The following sample code provides an example on how to set HTTPS as the protocol for a request:

```
request = DescribeInstancesRequest()
request.set_protocol_type("https")
# Set the value to https or http.
```

Set a global default protocol

The following sample code provides an example on how to set a global default protocol:

```
import aliyunsdkcore.request
aliyun_sdkcore.request.set_default_protocol_type("https")
# Create a request and call client.do_action_with_exception() to send the request.
```

2.3.2.4. Initiate a request

This topic describes how to use Apsara File Storage NAS SDK for Python to initiate a request.

Procedure

1. Import NAS SDK for Python.

```
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdknas.request.v20170626 import DescribeFileSystemsRequest
```

2. Create an AcsClient instance.

```
region_provider.modify_point(product_name, region_id, endpoint)
clt = client.AcsClient(access_key, access_secret, region_id)
```

3. Create a request object.

```
request = DescribeFileSystemsRequest.DescribeFileSystemsRequest()
request.set_accept_format('json')
```

4. Initiate the request and handle the response.

```
response = clt.do_action_with_exception(request)
print(response)
```

2.3.2.5. Handle errors

If the server or SDK encounters an error when you use the SDK for Python, the SDK returns an error response. The error response contains an error code and an error message.

Among the errors returned by the SDK for Python, you do not need to handle the SDK errors, but must solve the server errors.

- `ServerException` indicates a server error.
- `ClientException` indicates an SDK error.

For example, if the following error occurs, you can modify the AccessKey ID based on the error message:

```
aliyun_sdkcore.acs\_exception.exceptions.ServerException: HTTP Status: 404 Error:InvalidAccessKeyId.NotFound Specified access key is not found.
```

If you want to handle the SDK errors returned by the SDK, refer to the following example to write your code:

```
try:
    response = client.do_action_with_exception(request)
except ServerException as e:
    # You can add your error handling logic here.
    # For example, display the error message.
    print e.get_http_status()
    print e.get_error_code()
    print e.get_error_msg()
```

2.3.2.6. Use CommonRequest to call API operations

If you want to call the API operations of a cloud service that does not provide SDKs, or you cannot use an SDK to call API operations whose requests and responses meet your business requirements, you can use CommonRequest to call API operations. You can use CommonRequest to call all API operations.

Benefits

CommonRequest provides the following benefits:

- Lightweight: You only need to download the core package. You do not need to download and install SDKs of Apsara Stack services.
- Easy to use: You can call the newly released API operations of a service without the need to update the SDK for the service.
- Fast iteration.

Use CommonRequest to call an API operation

If you want to send a CommonRequest request, you must obtain the values of the following parameters. You can obtain the values of these parameters from the API reference of Apsara File Storage NAS.

- Domain: the endpoint of NAS.
 - Version: the version number of the API, in the YYYY-MM-DD format.
- You can obtain the API version in the [Common parameters](#) topic of the API reference.

- Operation information: the name of the API operation that you want to call.

You need to obtain the value of the Action parameter and specify an API operation in `request.set_action_name("Action")`.

For example, if you want to call the CreateFileSystem operation, you must use `request.set_action_name("CreateFileSystem")` to specify the API operation when you initiate a CommonRequest request.

Example: Call an RPC API operation

This example shows how to call the DescribeFileSystems operation to query file systems.

 **Note** In the following code, replace the variables with actual values. Delete the angle brackets <> in the final code.

You need to configure the following variables:

- <yourRegionID>: Replace the parameter with the region ID.
- <yourAccessKeyID>: Replace the parameter with your AccessKey ID.
- <yourAccessKeySecret>: Replace the parameter with your AccessKey secret.
- <yourSecurityToken>: Replace the parameter with your security token.
- <yourResourceGroupID>: Replace the parameter with the resource set ID.
- <yourOrganizationID>: Replace the parameter with the organization ID.
- <yourInstanceID>: Replace the parameter with the instance ID.

```

from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.request import CommonRequest
# Specify the AccessKey ID, AccessKey secret, and region ID.
client = AcsClient('{<yourAccessKeyId>}', '{<yourAccessKeySecret>}', '{<yourRegionID>}')
request = CommonRequest()
request.set_product('NAS')
request.set_action_name('DescribeFileSystems')
request.set_version('2017-06-26')
request.set_domain('<nas.aliyuncs.com>') # Specify the endpoint of the service.
# or:
# request = CommonRequest(product='NAS', action_name='DescribeFileSystems', version='2017-06-26', domain='nas.aliyuncs.com')
#####
# Specify the request protocol. Valid values: https and http. Default value: http.
# You can specify HTTPS as the protocol:
# request.set_protocol_type('https')
#####
request.set_accept_format('json')
request.set_method('POST')
request.add_query_param('PageNumber', '1')
request.add_query_param('PageSize', '30')
#####
# When you use an STS AccessKey 3-tuple to call an API operation, you must add the SecurityToken field in the request parameters.
# Format:
# Specify your security token.
# request.add_query_param("SecurityToken", "<yourSecurityToken>");
#####
# Optional. Specify the resource set ID.
request.add_header("x-acs-resourcegroupid", "<yourResourceGroupID>")
# Required. Specify the organization ID.
request.add_header("x-acs-organizationid", "<yourOrganizationID>")
# Required. Specify the source of the API call.
request.add_header("x-acs-caller-sdk-source", "ApiTestDemo")
# Optional. Specify the region ID. If you use Apsara Stack Enterprise V3.15.0 or earlier, this parameter is required.
request.add_header("x-acs-regionid", "<yourRegionID>")
# Optional. Specify the instance ID. If you need to delete, modify, or query an instance, this parameter is required.
# request.add_header("x-acs-instanceid", "<yourInstanceID>")
response = client.do_action_with_exception(request)
print(str(response, encoding = 'utf-8'))

```

2.4. API Reference

2.4.1. List of operations by function

The table lists the API operations that are available for use in Apsara File Storage NAS.

API operations

No.	Category	API operation	Description
1	Region	DescribeRegions	Queries all region IDs.
2	File system	CreateFileSystem	Creates a file system.
3	File system	DeleteFileSystem	Deletes a file system.
4	File system	DescribeFileSystems	Queries the information of file systems.
5	File system	ModifyFileSystem	Modifies the description of a file system.
6	Mount target	CreateMountTarget	Creates a mount target.
7	Mount target	DeleteMountTarget	Deletes a mount target.
8	Mount target	DescribeMountTargets	Queries the information of mount targets.
9	Mount target	ModifyMountTarget	Modifies a mount target.
10	Permission group	CreateAccessGroup	Creates a permission group.
11	Permission group	DeleteAccessGroup	Deletes a permission group.
12	Permission group	DescribeAccessGroups	Queries the information of permission groups.
13	Permission group	ModifyAccessGroup	Modifies a permission group.
14	Permission group	CreateAccessRule	Creates a rule for a permission group.
15	Permission group	DeleteAccessRule	Deletes a rule from a permission group.
16	Permission group	DescribeAccessRules	Queries the information of rules.
17	Permission group	ModifyAccessRule	Modifies a rule in a permission group.

2.4.2. Make API requests

2.4.2.1. Common parameters

Common request parameters

Common request parameters are available to all APIs.

Common request parameters

Name	Type	Required	Description
Format	String	No	The type of response. Valid values: JSON and XML. Default value: JSON.
Version	String	Yes	The API version, which is represented as a date in the format of <code>YYYY-MM-DD</code> . The current API version is 2017-06-26.
AccessKeyId	String	Yes	The AccessKey ID that is used to access services. For more information, see Obtain an AccessKey pair .
Signature	String	Yes	The string that represents signature.
SignatureMethod	String	Yes	The type of signature. Valid value: HMAC-SHA1.
Timestamp	String	Yes	The time when you send the request. Specify the time in the ISO 8601 standard in the <code>YYYY-MM-DDThh:mm:ssZ</code> format. The time must be in UTC. For example, 2017-05-26T12:00:00Z.
SignatureVersion	String	Yes	The version of the signature algorithm. The current version is 1.0.
SecurityToken	String	No	If Security Token Service (STS) credentials are required, you must specify this parameter.
SignatureNonce	String	No	A unique random number. This parameter is used to prevent replay attacks. You must use different random numbers for different requests.

Sample request

```
GET https://nas.regionid.example.com/?Action=<Action>
&Format=xml
&Version=2017-06-26
&AccessKeyId=key-test
&Signature=Pc5***%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&Timestamp=2012-06-01T12:00:00Z
...
```

Common response parameters

Whenever you send an API request, a GUID that is included in the **RequestId** parameter will be returned to you regardless of whether the request is successful.

Common response parameters

Name	Type	Description
RequestId	Integer	The GUID generated by Alibaba Cloud for the request.

Sample response

XML format

```
<? xml version="1.0" encoding="UTF-8"? >
<An API name+Response>
  <!--Returned RequestId-->
  <RequestId> 6D9F62C5-BF52-447C-AA34-C77F7AFCCC12</RequestId>
  <!--Response parameters-->
</An API name+Response>
```

JSON format

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
  /* Response parameters */
}
```

2.4.2.2. Request structure

Endpoints

The format of a NAS API endpoint is described in [Generate the endpoint of ASAPI](#).

Protocols

You can send API requests by using HTTP or HTTPS. We recommend that you use HTTPS to ensure high security.

Request methods

You can send requests by using the HTTP GET method. This method requires that you specify request parameters in the request URL.

Request parameters

You must specify the required operation for the Action parameter of each request, such as `DescribeRegions`. Each operation must include common request parameters and other request parameters that are specific to the operation.

Character encoding

Requests or responses are all encoded by `UTF-8`.

2.4.3. Region

2.4.3.1. DescribeRegions

Queries available regions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeRegions	The operation that you want to perform. Set the value to DescribeRegions.
PageSize	Integer	No	10	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	2	The number of entries returned per page.
Regions	Array of Region		The array of regions.
Region			
LocalName	String	East China 1	The name of the region.
RegionEndpoint	String	nas.cn-hangzhou.aliyuncs.com	The endpoint of the region.
RegionId	String	cn-hangzhou	The ID of the region.
RequestId	String	A70BEE5D-76D3-49FB-B58F-1F398211****	The ID of the request.
TotalCount	Integer	4	The total number of regions.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DescribeRegions
&<Common request parameters>
```

Sample success responses

XML format

```
<TotalCount>4</TotalCount>
<PageSize>2</PageSize>
<RequestId>A70BEE5D-76D3-49FB-B58F-1F398211****</RequestId>
<PageNumber>1</PageNumber>
<Regions>
  <RegionId>cn-hangzhou</RegionId>
  <RegionEndpoint>nas.cn-hangzhou.aliyuncs.com</RegionEndpoint>
  <LocalName>East China 1</LocalName>
</Regions>
```

JSON format

```
{
  "TotalCount": 4,
  "PageSize": 2,
  "RequestId": "A70BEE5D-76D3-49FB-B58F-1F398211****",
  "PageNumber": 1,
  "Regions": {
    "RegionId": "cn-hangzhou",
    "RegionEndpoint": "nas.cn-hangzhou.aliyuncs.com",
    "LocalName": "East China 1"
  }
}
```

2.4.4. NAS instance

2.4.4.1. CreateFileSystem

Creates a file system.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateFileSystem	The operation that you want to perform. Set the value to CreateFileSystem.
ProtocolType	String	Yes	NFS	The type of the protocol. Valid values: - NFS: Network File System (NFS) - SMB: Server Message Block (SMB)
StorageType	String	Yes	Performance	The type of the storage. Valid values: - Performance: Performance NAS file systems - Capacity: Capacity NAS file systems
ZoneId	String	No	cn-hangzhou-b	The ID of the zone. Each region has multiple isolated locations known as zones. The power supply and network of each zone are independent. Note - An Elastic Compute Service (ECS) instance can access a NAS file system in the same region but a different zone. - We recommend that you select the zone where the ECS instance resides. This prevents cross-zone latency between the file system and the ECS instance.
EncryptType	Integer	No	1	Specifies whether to encrypt the data in the file system. You can use keys that are hosted by Key Management Service (KMS) to encrypt the data that is stored in a file system. When you access the encrypted data, the data is automatically decrypted. Valid values: 0: The data in the file system is not encrypted. 1: The data in the file system is encrypted.

Parameter	Type	Required	Example	Description
Description	String	No	balabala	The description of the file system. Limits: - The description must be 2 to 128 characters in length. - The description must start with a letter but cannot start with <code>http://</code> or <code>https://</code>. - The description can contain digits, colons (:), underscores (_), and hyphens (-).
VolumeSize	Long	No	100	The capacity of the General-purpose NAS file system. Unit: bytes. Valid values: - Valid values for Capacity NAS file systems: 0 to 10 PB. - Valid values for Performance NAS file systems: 0 to 1 PB.

Response parameters

Parameter	Type	Example	Description
FileSystemId	String	1ca404****	The ID of the file system that is created.
RequestId	String	98696EF0-1607-4E9D-B01D-F20930B6****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateFileSystem
&ProtocolType=NFS
&StorageType=Performance
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>98696EF0-1607-4E9D-B01D-F20930B6****</RequestId>
<FileSystemId>1ca404****</FileSystemId>
```

JSON format

```
{
  "RequestId": "98696EF0-1607-4E9D-B01D-F20930B6****",
  "FileSystemId": "1ca404****"
}
```

2.4.4.2. DeleteFileSystem

Deletes a file system.

Description

- Before you delete a file system, you must remove all the mount targets from the file system.
- After the file system is deleted, the data on the file system cannot be restored. Proceed with caution.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteFileSystem	The operation that you want to perform. Set the value to DeleteFileSystem.
FileSystemId	String	Yes	1ca404****	The ID of the file system.

Response parameters

Parameter	Type	Example	Description
RequestId	String	9E15E394-38A6-457A-A62A-D9797C9A****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteFileSystem
&FileSystemId=1ca404****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>9E15E394-38A6-457A-A62A-D9797C9A****</RequestId>
```

JSON format

```
{
  "RequestId": "9E15E394-38A6-457A-A62A-D9797C9A****"
}
```

2.4.4.3. ModifyFileSystem

Modifies the capacity and details of a file system.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyFileSystem	The operation that you want to perform. Set the value to ModifyFileSystem.
FileSystemId	String	Yes	1ca404****	The ID of the file system.
Description	String	No	Left empty	The description of the file system. Limits: - The description must be 2 to 128 characters in length. - The description must start with a letter but cannot start with http:// or https://. - The description can contain digits, colons (:), underscores (_), and hyphens (-).
VolumeSize	Long	No	1000	The capacity of the file system. Valid values: - Valid values for Capacity NAS file systems: 0 to 10 PB. - Valid values for Performance NAS file systems: 0 to 1 PB.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5BC5CB97-9F28-42FE-84A4-0CD0DF42****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ModifyFileSystem
&FileSystemId=1ca404****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>5BC5CB97-9F28-42FE-84A4-0CD0DF42****</RequestId>
```

JSON format

```
{
  "RequestId": "5BC5CB97-9F28-42FE-84A4-0CD0DF42****"
}
```

2.4.4.4. DescribeFileSystems

Queries the information of file systems.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeFileSystems	The operation that you want to perform. Set the value to DescribeFileSystems.
FileSystemId	String	No	109c04****	The ID of the file system.
FileSystemType	String	No	standard	The type of the file system. Valid value: standard, which indicates General-purpose NAS file systems.
VpcId	String	No	vpc-bp1sevsqtqvk5gxml****	The ID of the virtual private cloud (VPC). If you want to mount the file system on an Elastic Compute Service (ECS) instance, you must set this parameter to the VPC where the ECS instance resides.
PageSize	Integer	No	1	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	1	The number of entries returned per page.
RequestId	String	035B3A3A-E514-4B41-B906-5D906CFB****	The ID of the request.
TotalCount	Integer	1	The total number of file systems.
FileSystems	Array of FileSystem		The list of file systems.
FileSystem			
Bandwidth	Long	150	The bandwidth of the file system. Unit: MB/s. This parameter is unavailable for General-purpose NAS file systems.
ChargeType	String	PayAsYouGo	The billing method of the file system. Valid values: - Subscription: The subscription billing method is used. - PayAsYouGo: The pay-as-you-go billing method is used. - Package: A storage plan is attached to the file system.
CreateTime	String	2017-05-27T15:43:06CST	The time when the file system was created.
Description	String	This can be the description of the created file system.	The description of the file system.
EncryptType	Integer	1	The type of the encryption. Valid values: 0: The file system is not encrypted. 1: The file system is encrypted.
ExpiredTime	String	1572174830000	The time when the file system expires.
FileSystemId	String	109c04****	The ID of the file system.

Parameter	Type	Example	Description
FileSystemType	String	all	The type of the file system.
Ldap	Struct		The details of the Lightweight Directory Access Protocol (LDAP) service. This parameter is available only for CPFS file systems.
BindDN	String	cn=alibaba,dc=com	An LDAP entry.
SearchBase	String	dc=example	An LDAP search base.
URI	String	ldap://ldap.example.example	The URL of an LDAP server.
MeteredIASize	Long	1000	The used capacity of the Infrequent Access (IA) storage medium.
MeteredSize	Long	1611661312	The used capacity of the file system. The value of this parameter is the maximum used capacity of the file system over the last hour. Unit: bytes.
MountTargets	Array of MountTarget		The list of mount targets.
MountTarget			
AccessGroupName	String	test-001	The name of the permission group that is attached to the mount target.
MountTargetDomain	String	109c042666-w****.cn-hangzhou.nas.aliyuncs.com	The domain name of the mount target.
NetworkType	String	vpc	The type of the network. Valid values: - vpc: the virtual private cloud (VPC) network. - classic: the classic network.
Status	String	active	The status of the mount target. Valid values: - Active: The mount target is available. - Inactive: The mount target is unavailable.
VpcId	String	vpc-bp1sevsgtqvk5gxb****	The ID of the VPC.
VswId	String	vsw-bp1omfzsszekkvaxn****	The ID of the vSwitch.
Packages	Array of Package		The list of storage plans.
Package			
ExpiredTime	String	2020-01-05T16:00:00Z	The time when the storage plan expires.
PackageId	String	naspackage-0be9c4b624-37****	The ID of the storage plan.
PackageType	String	hybrid	The type of the storage plan.
Size	Long	107374182400	The capacity of the storage plan.
StartTime	String	2019-12-05T01:40:56Z	The start time of the validity period for the storage plan.
ProtocolType	String	NFS	The type of the protocol. Valid values: - NFS: the Network File System (NFS) protocol. - SMB: the Server Message Block (SMB) protocol.
QuotaSize	Long	100	The capacity of the file system.

Parameter	Type	Example	Description
RegionId	String	cn-hangzhou	The ID of the region.
Status	String	Pending	The status of the file system. This parameter is returned for Extreme NAS file systems. Valid values: • Pending: The file system is being created or modified. • Running: The file system is available. Before you create a mount target for the file system, make sure that the file system is in the Running state. • Stopped: The file system is unavailable. • Extending: The file system is being scaled up. • Stopping: The file system is being stopped. • Deleting: The file system is being deleted.
StorageType	String	Performance	The type of the storage. Valid values: • Valid values for General-purpose NAS file systems: Capacity and Performance. • Valid values for Extreme NAS file systems: standard and advance.
SupportedFeatures	List	Lifecycle	The features that are supported by the file system. Valid values: • Lifecycle: the lifecycle management feature. • Quota: the quota management feature.
VolumeSize	Long	1000	The capacity of the General-purpose NAS file system. Valid values: • Valid values for NAS Capacity file systems: 0 to 10 PB. • Valid values for NAS Performance file systems: 0 to 1 PB.
ZoneId	String	cn-hangzhou-b	The ID of the zone where the file system resides.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeFileSystems
&<Common request parameters>
```

Sample success responses

XML format

```

<FileSystems>
  <Status>Pending</Status>
  <StorageType>Performance</StorageType>
  <Description>This can be the description of the created file system. </Description>
  <ZoneId>cn-hangzhou-b</ZoneId>
  <MeteredIASize>1000</MeteredIASize>
  <CreateTime>2017-05-27T15:43:06CST</CreateTime>
  <QuotaSize>100</QuotaSize>
  <MeteredSize>1611661312</MeteredSize>
  <EncryptType>1</EncryptType>
  <ChargeType>PayAsYouGo</ChargeType>
  <ProtocolType>NFS</ProtocolType>
  <Bandwidth>150</Bandwidth>
  <FileSystemType>all</FileSystemType>
  <ExpiredTime>1572174830000</ExpiredTime>
  <FileSystemId>109c04****</FileSystemId>
  <RegionId>cn-hangzhou</RegionId>
  <VolumeSize>1000</VolumeSize>
</FileSystems>
<FileSystems>
  <MountTargets>
    <Status>active</Status>
    <VswId>vsw-bplomfzsszekkvaxn****</VswId>
    <VpcId>vpc-bplsevsqtkv5gxb1****</VpcId>
    <MountTargetDomain>109c042666-w****.cn-hangzhou.nas.aliyuncs.com</MountTargetDomain>
    <NetworkType>vpc</NetworkType>
    <AccessGroupName>test-001</AccessGroupName>
  </MountTargets>
  <Packages>
    <StartTime>2019-12-05T01:40:56Z</StartTime>
    <PackageId>naspackage-0be9c4b624-37****</PackageId>
    <Size>107374182400</Size>
    <ExpiredTime>2020-01-05T16:00:00Z</ExpiredTime>
    <PackageType>hybrid</PackageType>
  </Packages>
</FileSystems>
<FileSystems>
  <SupportedFeatures>Lifecycle</SupportedFeatures>
</FileSystems>
<FileSystems>
  <Ldap>
    <SearchBase>dc=example</SearchBase>
    <URI>ldap://ldap.example.example</URI>
    <BindDN>cn=alibaba,dc=com</BindDN>
  </Ldap>
</FileSystems>
<TotalCount>1</TotalCount>
<PageSize>1</PageSize>
<RequestId>035B3A3A-E514-4B41-B906-5D906CFB****</RequestId>
<PageNumber>1</PageNumber>

```

JSON **format**

```
{
  "FileSystems": [
    {
      "Status": "Pending",
      "StorageType": "Performance",
      "Description": "This can be the description of the created file system.",
      "ZoneId": "cn-hangzhou-b",
      "MeteredIASize": 1000,
      "CreateTime": "2017-05-27T15:43:06CST",
      "QuotaSize": 100,
      "MeteredSize": 1611661312,
      "EncryptType": 1,
      "ChargeType": "PayAsYouGo",
      "ProtocolType": "NFS",
      "Bandwidth": 150,
      "FileSystemType": "all",
      "ExpiredTime": 1572174830000,
      "FileSystemId": "109c04****",
      "RegionId": "cn-hangzhou",
      "VolumeSize": 1000
    },
    {
      "MountTargets": {
        "Status": "active",
        "VswId": "vsw-bplomfzsszekkvaxn****",
        "VpcId": "vpc-bplsevgqtqvk5gxbl****",
        "MountTargetDomain": "109c042666-w****.cn-hangzhou.nas.aliyuncs.com",
        "NetworkType": "vpc",
        "AccessGroupName": "test-001"
      },
      "Packages": {
        "StartTime": "2019-12-05T01:40:56Z",
        "PackageId": "naspacage-0be9c4b624-37****",
        "Size": 107374182400,
        "ExpiredTime": "2020-01-05T16:00:00Z",
        "PackageType": "hybrid"
      }
    },
    {
      "SupportedFeatures": "Lifecycle"
    },
    {
      "Ldap": {
        "SearchBase": "dc=example",
        "URI": "ldap://ldap.example.example",
        "BindDN": "cn=alibaba,dc=com"
      }
    }
  ],
  "TotalCount": 1,
  "PageSize": 1,
  "RequestId": "035B3A3A-E514-4B41-B906-5D906CFB****",
  "PageNumber": 1
}
```

2.4.5. Mount point

2.4.5.1. CreateMountTarget

Creates a mount target.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateMountTarget	The operation that you want to perform. Set the value to CreateMountTarget.
AccessGroupName	String	Yes	classic-test	The name of the permission group. Default permission groups: DEFAULT_VPC_GROUP_NAME (the default permission group for VPC) and DEFAULT_CLASSIC_GROUP_NAME (the default permission group for the classic network).
FileSystemId	String	Yes	174494****	The ID of the file system.

Parameter	Type	Required	Example	Description
NetworkType	String	Yes	VPC	The network type of the mount target. Valid values: - Vpc: the virtual private cloud (VPC) network. - Classic: the classic network.
VpcId	String	No	vpc-2zesj9afh3y518k9o****	The ID of the VPC. This parameter is valid and required if the mount target resides in a VPC.
VSwitchId	String	No	vsw-2zevmwkwytjuoffg****	The ID of the vSwitch. This parameter is valid and required if the mount target resides in a VPC.
SecurityGroupId	String	No	sg-bp1fg655nh68xyz9****	The ID of the security group.

Response parameters

Parameter	Type	Example	Description
MountTargetDomain	String	174494b666-x****.cn-hangzhou.nas.aliyuncs.com	The domain name of the mount target.
RequestId	String	70EACC9C-D07A-4A34-ADA4-77506C42****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CreateMountTarget
&AccessGroupName=classic-test
&FileSystemId=174494****
&NetworkType=VPC
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>70EACC9C-D07A-4A34-ADA4-77506C42****</RequestId>
<MountTargetDomain>174494b666-x****.cn-hangzhou.nas.aliyuncs.com</MountTargetDomain>
```

JSON format

```
{
  "RequestId": "70EACC9C-D07A-4A34-ADA4-77506C42****",
  "MountTargetDomain": "174494b666-x****.cn-hangzhou.nas.aliyuncs.com"
}
```

2.4.5.2. DeleteMountTarget

Deletes a mount target.

Description

After you delete a mount target, the mount target cannot be restored. Proceed with caution.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteMountTarget	The operation that you want to perform. Set the value to DeleteMountTarget.
FileSystemId	String	Yes	174494****	The ID of the file system.
MountTargetDomain	String	Yes	174494b666-x****.cn-hangzhou.nas.aliyuncs.com	The domain name of the mount target.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5BC5CB97-9F28-42FE-84A4-0CD0DF42****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteMountTarget
&FileSystemId=174494****
&MountTargetDomain=174494b666-x****.cn-hangzhou.nas.aliyuncs.com
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>5BC5CB97-9F28-42FE-84A4-0CD0DF42****</RequestId>
```

JSON format

```
{
  "RequestId": "5BC5CB97-9F28-42FE-84A4-0CD0DF42****"
}
```

2.4.5.3. DescribeMountTargets

Queries the information of mount targets.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeMountTargets	The operation that you want to perform. Set the value to DescribeMountTargets.
FileSystemId	String	Yes	1ca404****	The ID of the file system.
MountTargetDomain	String	No	1ca404a666-x****.cn-hangzhou.nas.aliyuncs.com	The domain name of the mount target.
PageSize	Integer	No	1	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.

Response parameters

Parameter	Type	Example	Description
MountTargets	Array of MountTarget		The list of mount targets.
MountTarget			
AccessGroup	String	DEFAULT_VPC_GROUP_NAME	The name of the permission group that is attached to the mount target.
MountTargetDomain	String	1ca404a666-w****.cn-hangzhou.nas.aliyuncs.com	The domain name of the mount target.
NetworkType	String	Vpc	The network type of the mount target.
Status	String	Active	The status of the mount target. Valid values: - Active: The mount target is available. - Inactive: The mount target is unavailable.  Note A file system can be mounted only if the file system is in the Active state.

Parameter	Type	Example	Description
VpcId	String	vpc-2zesj9afh3y518k9o****	The ID of the virtual private cloud (VPC).
VswId	String	vsw-2zevmkwkwyztjuoffg****	The ID of the vSwitch.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	1	The number of entries returned per page.
RequestId	String	3BAB90FD-B4A0-48DA-9F09-2B963510A0EF	The ID of the request.
TotalCount	Integer	1	The total number of mount targets.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeMountTargets
&FileSystemId=1ca404****
&<Common request parameters>
```

Sample success responses

XML format

```
<TotalCount>1</TotalCount>
<PageSize>1</PageSize>
<RequestId>3BAB90FD-B4A0-48DA-9F09-2B963510A0EF</RequestId>
<PageNumber>1</PageNumber>
<MountTargets>
  <Status>Active</Status>
  <VswId>vsw-2zevmkwkwyztjuoffg****</VswId>
  <VpcId>vpc-2zesj9afh3y518k9o****</VpcId>
  <MountTargetDomain>1ca404a666-w****.cn-hangzhou.nas.aliyuncs.com</MountTargetDomain>
  <AccessGroup>DEFAULT_VPC_GROUP_NAME</AccessGroup>
  <NetworkType>Vpc</NetworkType>
</MountTargets>
```

JSON format

```
{
  "TotalCount": 1,
  "PageSize": 1,
  "RequestId": "3BAB90FD-B4A0-48DA-9F09-2B963510A0EF",
  "PageNumber": 1,
  "MountTargets": {
    "Status": "Active",
    "VswId": "vsw-2zevmkwkwyztjuoffg****",
    "VpcId": "vpc-2zesj9afh3y518k9o****",
    "MountTargetDomain": "1ca404a666-w****.cn-hangzhou.nas.aliyuncs.com",
    "AccessGroup": "DEFAULT_VPC_GROUP_NAME",
    "NetworkType": "Vpc"
  }
}
```

2.4.5.4. ModifyMountTarget

Modifies a mount target.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyMountTarget	The operation that you want to perform. Set the value to ModifyMountTarget.
FileSystemId	String	Yes	1ca404****	The ID of the file system.
MountTargetDomain	String	Yes	1ca404a666-w****.cn-hangzhou.nas.aliyuncs.com	The domain name of the mount target.
AccessGroupName	String	No	classic-test	The name of the permission group that is attached to the mount target.

Parameter	Type	Required	Example	Description
Status	String	No	Inactive	The status of the mount target. Valid values: - Active: The mount target is available. - Inactive: The mount target is unavailable.

Response parameters

Parameter	Type	Example	Description
RequestId	String	FF387D95-34C4-4879-B65A-99D1FA1****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyMountTarget
&FileSystemId=lca404****
&MountTargetDomain=lca404a666-w****.cn-hangzhou.nas.aliyuncs.com
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>FF387D95-34C4-4879-B65A-99D1FA1****</RequestId>
```

JSON format

```
{
  "RequestId": "FF387D95-34C4-4879-B65A-99D1FA1****"
}
```

2.4.6. Permission group

2.4.6.1. CreateAccessGroup

Creates a permission group.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateAccessGroup	The operation that you want to perform. Set the value to CreateAccessGroup.
AccessGroupName	String	Yes	classic-test	The name of the permission group. Limits: - The name must be 3 to 64 characters in length. - The name must start with a letter and can contain letters, digits, underscores (_), and hyphens (-). - The name of the permission group must be different from the name of the default permission groups. Default permission groups: DEFAULT_VPC_GROUP_NAME (the default permission group for VPC) and DEFAULT_CLASSIC_GROUP_NAME (the default permission group for the classic network).
AccessGroupType	String	Yes	Classic	The type of the permission group. Valid values: - Vpc: the virtual private cloud (VPC) network. - Classic: the classic network.

Parameter	Type	Required	Example	Description
Description	String	No	classic-test-access-group	The description of the permission group. Limits: - By default, the description of a permission group is the same as the name of the permission group. The description must be 2 to 128 characters in length. - The description must start with a letter and cannot start with http:// or https://. - The description can contain digits, colons (:), underscores (_), and hyphens (-).
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
AccessGroupName	String	classic-test	The name of the permission group.
RequestId	String	55C5FFD6-BF99-41BD-9C66-FFF39189****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CreateAccessGroup
&AccessGroupName=classic-test
&AccessGroupType=Classic
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>55C5FFD6-BF99-41BD-9C66-FFF39189****</RequestId>
<AccessGroupName>classic-test</AccessGroupName>
```

JSON format

```
{
  "RequestId": "55C5FFD6-BF99-41BD-9C66-FFF39189****",
  "AccessGroupName": "classic-test"
}
```

2.4.6.2. DeleteAccessGroup

Deletes a permission group.

Description

Default permission groups include DEFAULT_VPC_GROUP_NAME and DEFAULT_CLASSIC_GROUP_NAME. You cannot delete the default permission groups.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteAccessGroup	The operation that you want to perform. Set the value to DeleteAccessGroup.
AccessGroupName	String	Yes	classic-test	The name of the permission group.
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
RequestId	String	9E15E394-38A6-457A-A62A-D9797C9A****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteAccessGroup
&AccessGroupName=classic-test
&<Common request parameters>
```

Sample success responses

XML **format**

```
<RequestId>9E15E394-38A6-457A-A62A-D9797C9A****</RequestId>
```

JSON **format**

```
{
  "RequestId": "9E15E394-38A6-457A-A62A-D9797C9A****"
}
```

2.4.6.3. ModifyAccessGroup

Modifies a permission group.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyAccessGroup	The operation that you want to perform. Set the value to ModifyAccessGroup.
AccessGroupName	String	Yes	classic-test	The name of the permission group. Limits: - The name must be 3 to 64 characters in length. - The name must start with a letter. The name can contain letters, digits, underscores (_), and hyphens (-).
Description	String	No	classic-test	The description of the permission group. Limits: - By default, the description of the permission group is the same as the name of the permission group. The description must be 2 to 128 characters in length. - The description must start with a letter and cannot start with <code>http://</code> or <code>https://</code>. - The description can contain digits, colons (:), underscores (_), and hyphens (-).
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
RequestId	String	ED2AE737-9D50-4CA4-B0DA-31BD610C****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyAccessGroup
&AccessGroupName=classic-test
&<Common request parameters>
```

Sample success responses

XML **format**

```
<RequestId>ED2AE737-9D50-4CA4-B0DA-31BD610C****</RequestId>
```

JSON format

```
{
  "RequestId": "ED2AE737-9D50-4CA4-B0DA-31BD610C****"
}
```

2.4.6.4. DescribeAccessGroups

Queries the information of permission groups.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeAccessGroups	The operation that you want to perform. Set the value to DescribeAccessGroups.
AccessGroupName	String	No	classic-test	The name of the permission group.
PageSize	Integer	No	2	The number of entries to return per page. Valid values: 1 to 100. Default value: 10.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.
UseUTCDateTime	Boolean	No	true	Specifies whether the time to return is in UTC. Valid values: - true: the default value. The time is returned in the UTC format. - false: The time is not returned in the UTC format.
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
AccessGroups	Array of AccessGroup		The list of permission groups.
AccessGroup			
AccessGroupName	String	classic-test	The name of the permission group.
AccessGroupType	String	Classic	The type of the permission group. Valid values: - Vpc: the virtual private cloud (VPC) network. - Classic: the classic network.
Description	String	This is a classic test access group~	The description of the permission group.
MountTargetCount	Integer	0	The number of mount targets to which the permission group is attached.
RuleCount	Integer	0	The total number of rules in the permission group.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	2	The number of entries returned per page.
RequestId	String	2514F97E-FFF0-4A1F-BF04-729CEAC64B6F	The ID of the request.
ResourceOwnerId	String	258456	The ID of the resource.

Parameter	Type	Example	Description
TotalCount	Integer	2	The total number of permission groups.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeAccessGroups
&<Common request parameters>
```

Sample success responses

XML **format**

```
<AccessGroups>
  <RuleCount>0</RuleCount>
  <AccessGroupType>Classic</AccessGroupType>
  <Description>This is a classic test access group~</Description>
  <MountTargetCount>0</MountTargetCount>
  <AccessGroupName>classic-test</AccessGroupName>
</AccessGroups>
<TotalCount>2</TotalCount>
<PageSize>2</PageSize>
<RequestId>2514F97E-FFF0-4A1F-BF04-729CEAC64B6F</RequestId>
<PageNumber>1</PageNumber>
<ResourceOwnerId>258456</ResourceOwnerId>
```

JSON **format**

```
{
  "AccessGroups": {
    "RuleCount": 0,
    "AccessGroupType": "Classic",
    "Description": "This is a classic test access group~",
    "MountTargetCount": 0,
    "AccessGroupName": "classic-test"
  },
  "TotalCount": 2,
  "PageSize": 2,
  "RequestId": "2514F97E-FFF0-4A1F-BF04-729CEAC64B6F",
  "PageNumber": 1,
  "ResourceOwnerId": 258456
}
```

2.4.6.5. CreateAccessRule

Creates a rule for a permission group.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateAccessRule	The operation that you want to perform. Set the value to CreateAccessRule.
AccessGroupName	String	Yes	classic-test	The name of the permission group.
SourceCidrIp	String	Yes	10.0.*.*/**	The IP address or CIDR block of the authorized object. It must be an IP address or CIDR block.  Note If the permission group resides in the classic network, only IP addresses are supported.
RWAccessType	String	No	RDWR	Grants access permissions on the file system to the authorized object. Valid values: - RDWR: the default value, which indicates read and write permissions. - RDONLY: the read-only permissions.

Parameter	Type	Required	Example	Description
UserAccessType	String	No	no_squash	The access permissions for different types of users in the authorized object. Valid values: - no_squash: allows access from root users to the file system. - root_squash: grants root users the least permissions as the nobody user. - all_squash: grants all users the least permissions as the nobody user. The nobody user has the least permissions in Linux and can access only the public content of the file system. This ensures the security of the file system.
Priority	Integer	No	1	The priority of the rule. The rule with the highest priority takes effect if multiple rules are attached to the authorized object. Valid values: 1 to 100. The value 1 indicates the highest priority.
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
AccessRuleId	String	1	The ID of the rule.
RequestId	String	A323836B-5BC6-45A6-8048-60675C23****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CreateAccessRule
&AccessGroupName=classic-test
&SourceCidrIp=10.0. *. */**
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>A323836B-5BC6-45A6-8048-60675C23****</RequestId>
<AccessRuleId>1</AccessRuleId>
```

JSON format

```
{
  "RequestId": "A323836B-5BC6-45A6-8048-60675C23****",
  "AccessRuleId": 1
}
```

2.4.6.6. DeleteAccessRule

Deletes a rule from a permission group.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteAccessRule	The operation that you want to perform. Set the value to DeleteAccessRule.
AccessGroupName	String	Yes	classic-test	The name of the permission group.
AccessRuleId	String	Yes	1	The ID of the rule.

Parameter	Type	Required	Example	Description
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5B4511A7-C99E-4071-AA8C-32E2529D****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteAccessRule
&AccessGroupName=classic-test
&AccessRuleId=1
&<Common request parameters>
```

Sample success responses

XML **format**

```
<RequestId>5B4511A7-C99E-4071-AA8C-32E2529D****</RequestId>
```

JSON **format**

```
{
  "RequestId": "5B4511A7-C99E-4071-AA8C-32E2529D****"
}
```

2.4.6.7. ModifyAccessRule

Modifies a rule in a permission group.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyAccessRule	The operation that you want to perform. Set the value to ModifyAccessRule.
AccessGroupName	String	Yes	classic-test	The name of the permission group.
AccessRuleId	String	Yes	1	The ID of the rule.
SourceCidrIp	String	Yes	192.1**.*	The IP address or CIDR block of the authorized object. It must be an IP address or CIDR block.  Note If the permission group resides in the classic network, only IP addresses are supported.
RWAccessType	String	No	RDWR	Grants access permissions on the file system to the authorized object. Valid values: - RDWR: the default value, which indicates the read and write permissions. - RDONLY: the read-only permissions.

Parameter	Type	Required	Example	Description
UserAccessType	String	No	all_squash	The access permissions for different types of users in the authorized object. Valid values: - no_squash: allows access from root users to the file system. - root_squash: grants root users the least permissions as the nobody user. - all_squash: grants all users the least permissions as the nobody user. The nobody user has the least permissions in Linux and can access only the public content of the file system. This ensures the security of the file system.
Priority	Integer	No	1	The priority of the rule in the permission group. Valid values: 1 to 100. Default value: 1, which indicates the highest priority.
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
RequestId	String	6299428C-3861-435D-AE54-9B330A00****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ModifyAccessRule
&AccessGroupName=classic-test
&AccessRuleId=1
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>6299428C-3861-435D-AE54-9B330A00****</RequestId>
```

JSON format

```
{
  "RequestId": "6299428C-3861-435D-AE54-9B330A00****"
}
```

2.4.6.8. DescribeAccessRules

Queries the information about rules.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeAccessRules	The operation that you want to perform. Set the value to DescribeAccessRules.
AccessGroupName	String	Yes	classic-test	The name of the permission group.
AccessRuleId	String	No	1	The ID of the rule.
PageSize	Integer	No	1	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.

Parameter	Type	Required	Example	Description
FileSystemType	String	No	standard	The type of the file system. Default value: standard, which indicates General-purpose NAS file systems.

Response parameters

Parameter	Type	Example	Description
AccessRules	Array of AccessRule		The list of rules.
AccessRule			
AccessRuleId	String	1	The ID of the rule.
Ipv6SourceCidrIp	String	2001:250:6000::***	The IPv6 address or IPv6 CIDR block of the authorized object.
Priority	Integer	1	The priority of the rule. If multiple rules are attached to the authorized object, the rule with the highest priority takes effect. Valid values: 1 to 100. The value 1 indicates the highest priority.
RWAccess	String	RDWR	Grants access permissions on the file system to the authorized object. Valid values: - RDWR: the default value, which indicates the read and write permissions. - RDONLY: the read-only permissions.
SourceCidrIp	String	10.0.*.*/**	The IP address or CIDR block of the authorized object.
UserAccess	String	no_squash	The access permissions for different types of users in the authorized object. Valid values: no_squash, root_squash, and all_squash. Default value: no_squash.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	1	The number of entries returned per page.
RequestId	String	86D89E82-4297-4343-8E1E-A2495B35CC70	The ID of the request.
TotalCount	Integer	1	The total number of rules.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DescribeAccessRules
&AccessGroupName=classic-test
&<Common request parameters>
```

Sample success responses

XML format

```
<TotalCount>1</TotalCount>
<PageSize>1</PageSize>
<RequestId>86D89E82-4297-4343-8E1E-A2495B35CC70</RequestId>
<PageNumber>1</PageNumber>
<AccessRules>
  <RWAccess>RDWR</RWAccess>
  <UserAccess>no_squash</UserAccess>
  <Priority>1</Priority>
  <SourceCidrIp>10.0.*.*/**</SourceCidrIp>
  <Ipv6SourceCidrIp>2001:250:6000::***</Ipv6SourceCidrIp>
  <AccessRuleId>1</AccessRuleId>
</AccessRules>
```

JSON format


```
{
  "TotalCount": 1,
  "PageSize": 1,
  "RequestId": "86D89E82-4297-4343-8E1E-A2495B35CC70",
  "PageNumber": 1,
  "AccessRules": {
    "RWAccess": "RDWR",
    "UserAccess": "no_squash",
    "Priority": 1,
    "SourceCidrIp": "10.0.*.*/*",
    "Ipv6SourceCidrIp": "2001:250:6000::***",
    "AccessRuleId": 1
  }
}
```

2.4.7. Appendix: Error codes

An error code consists of four components, as listed in .

Error codes

Component	Description
HttpCode	The HTTP response code that complies with HTTP syntax
Code	The error code
Message	The error message
Advice	The recommended solution

Error codes

HTTP status code	Code	Message	Advice
500	InternalServerError	A system internal error occurred.	Please try again later.
503	ServiceUnavailable	This service is unavailable right now.	Please try again later.
504	ServiceTimeout	Service request timed out.	Please try again later.
503	SubRam.CheckFailed	The sub user ram check has failed due to some internal error.	Please try again later.
503	VpcRam.CheckFailed	The Vpc ram check has failed due to some internal error.	Please try again later.
503	VpcSubRam.CheckFailed	The Vpc sub user ram check has failed due to some internal error.	Please try again later.
400	MissingParameter.callerUid	Request parameter missing: callerUid is not specified.	Add the callerUid request parameter.
400	MissingParameter.RegionId	Request parameter missing: RegionId of the domain name is not specified.	Add the RegionId field to the domain name.
400	MissingParameter.VSwitchId	Request parameter missing: VSwitchId is not specified.	Add the VSwitchId request parameter.
400	MissingParameter.VpcId	Request parameter missing: VpcId is not specified.	Add the VpcId request parameter.
400	InvalidParameter.StorageType	Invalid request parameter: StorageType is invalid.	Set StorageType to Performance or Capacity.
400	InvalidParameter.ProtocolType	Invalid request parameter: ProtocolType is invalid.	Set ProtocolType to NFS.
400	InvalidParameter.Description	Invalid request parameter: Description contains invalid characters.	Set Description to a string that contains English letters and special characters such as at signs (@), number signs (#), dollar signs (\$), underscores (_), plus signs (+), and hyphens (-).
404	InvalidFileSystem.NotFound	The specified NAS instance does not exist.	The specified NAS instance does not exist or has been deleted.
403	InvalidFileSystem.AlreadyExists	The specified NAS instance already exists.	Call DescribeFileSystems and specify the NAS instance ID to view the instance information.
404	InvalidMountTarget.NotFound	The specified mount point does not exist.	The specified mount point does not exist or has been deleted.

HTTP status code	Code	Message	Advice
404	InvalidAccessGroup.NotFound	The specified permission group does not exist.	The specified permission group does not exist or has been deleted.
403	InvalidAccessGroup.AlreadyExisted	The specified permission group already exists.	Call DescribeMountTargets and specify the permission group to view its information.
403	InvalidAccessGroup.AlreadyAttached	The specified permission group has been bound to a mount point.	Unbind the permission group from the mount point before deleting the permission group.
404	InvalidAccessRule.NotFound	The specified permission rule does not exist.	The specified permission rule does not exist or has been deleted.
404	InvalidLocatin.NotFound	The specified location does not exist.	The requested location does not exist or is not providing services.
404	InvalidRegionId.NotFound	The specified region does not exist.	The requested region does not exist or is not providing services.
404	InvalidAZone.NotFound	The specified zone does not exist.	The requested zone does not exist or is not providing services.
404	InvalidVpc.NotFound	The specified VPC does not exist.	Create a VPC first.
404	InvalidVip.NotFound	The specified VIP does not exist.	Make sure that there are available IP addresses in your VPC.
404	InvalidLBid.NotFound	The specified LBid does not exist.	Please try again later.
404	InvalidVpcTunnelId.NotFound	The specified VPC tunnelId does not exist.	Check the VPC information first.
404	InvalidVpc.CreationFailed	The specified vpc instance creation failed.	Make sure that the VPC information is correct and there is an available IP address for the VPC, and then try again.
403	OperationDenied.DefaultAccessGroupCan notDelete	Operation not allowed: The default permission group cannot be deleted.	You can only edit custom permission groups.
403	OperationDenied.DefaultAccessGroupCan notModify	Operation not allowed: The default permission group cannot be modified.	You can only edit custom permission groups.
403	OperationDenied.NetworkTypeNotMatched	Operation not allowed: The mount point network type does not match the AccessGroupType.	Make sure that the NetworkType of the mount point and the AccessGroupType of the permission group are consistent.
403	OperationDenied.MountTargetNotEmpty	Operation not allowed: The specified NAS instance already has a mount point. Delete the existing mount point first.	Delete the existing mount points first.
403	OperationDenied.FileSystemCountsExceeded	The number of NAS instances has already reached the upper limit (a user can create a maximum of 10 NAS instances).	To raise the limit, contact technical support.
403	OperationDenied.MountTargetCountsExceeded	The number of mount points has already reached the upper limit (a maximum of two mount points can be created for each NAS instance).	To raise the limit, contact technical support.
403	OperationDenied.AccessGroupCountsExceeded	The number of permission groups has already reached the upper limit.	To raise the limit, contact technical support.
403	OperationDenied.AccessRuleCountsExceeded	The number of permission rules has already reached the upper limit.	To raise the limit, contact technical support.
403	Forbidden.InvalidUserType	No permission: The user type is invalid.	Sorry, this API does not support your account type.
403	Forbidden.Ram	User not authorized to operate on the specified resource, or this API doesn't support RAM.	Access RAM to grant required NAS permissions.
403	NotQualified	No public beta qualification yet	Please initiate an application for a trial of the related features. If your application fails to be approved, contact the technical support.

Error response examples

- XML format

```
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <RequestId>B71B4C68-6C6B-4D4B-AD70-8951BD7D188E</RequestId>
  <HostId>nas.regionid.example.com</HostId>
  <Code>InvalidFileSystem.NotFound</Code>
  <Message>The specified NAS instance does not exist.</Message>
</Error>
```

- JSON format

```
{
  "RequestId": "B71B4C68-6C6B-4D4B-AD70-8951BD7D188E",
  "HostId": " nas.regionid.example.com ",
  "Code": "InvalidFileSystem.NotFound",
  "Message": "The specified NAS instance does not exist."
}
```

3.EBS

3.1. Usage notes

If you call an API operation to use a feature of Elastic Block Storage (EBS), you must specify a service name and an API version. The service name and API version vary depending on the feature that you want to use.

- To use an EBS feature related to cloud disks, snapshots, or storage sets, call the corresponding API operation by using an Elastic Computer Service (ECS) SDK. When you use an ECS SDK to call the corresponding API operation, specify `ecs` as the service name and 2014-05-26 as the API version.
- To use the asynchronous replication or replication pair-consistent group feature of EBS, call the corresponding API operation by using an EBS SDK. When you use an EBS SDK to call the corresponding API operation, specify `ebs` as the service name and 2021-07-30 as the API version.

 **Notice** Asynchronous replication works based on cloud disks. Therefore, if you want to use the asynchronous replication or replication pair-consistent group feature, we recommend that you also install an ECS SDK.

3.2. Preparations

3.2.1. Log on to the API Tool console

You can log on to the API Tool console from the Apsara Uni-manager Management Console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment personnel.
- A browser is available. We recommend that you use Google Chrome.

Procedure

1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press Enter.
2. Enter your username and password.

Obtain the username and password that are used to log on to the Apsara Uni-manager Management Console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your account. For security purposes, your password must meet the minimum complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA verification code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Others > API Tool**.

3.2.2. Obtain API information

To call API operations by using an SDK, you must first obtain information about the service name, API operation name, and API version.

Context

The information on the API Directory page of the Apsara Uni-manager Management Console is for reference only. For more information about how to call API operations, see the developer guides of cloud services. If the information on the API Directory page is different from that in the developer guide of a cloud service, the information in the developer guide prevails.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Products** and choose **Others > API Tool**.
3. In the left-side navigation pane, click **API Directory**.
4. On the **API Directory** page, select the required cloud service from the Product Name drop-down list, enter a keyword of an operation name in the search box, and then click Search.
5. In the search result, you can view the **cloud service name**, **API operation name**, and **API version**.

3.2.3. Obtain access credentials assigned by RAM

Access credentials can be assigned by RAM or STS. You can use access credentials assigned by either of the preceding services for authentication when you call API operations. This topic describes how to obtain access credentials that are assigned by RAM.

Background information

An AccessKey ID and an AccessKey secret are used to implement symmetric encryption and verify the identity of a request sender. The AccessKey ID is used to identify a user. The AccessKey secret is a string that is used to encrypt a signature.

RAM and STS assign the following access credentials to a request sender:

- RAM: an AccessKey ID and an AccessKey secret.
- STS: an AccessKey ID, an AccessKey secret, and a security token. For more information, see [Obtain access credentials assigned by STS](#).

The Apsara Uni-manager Management Console provides AccessKey pairs for personal accounts and organizations. Only operations administrators and the administrators of level-1 organizations can obtain the AccessKey pairs of organizations. We recommend that you use the AccessKey pairs of personal accounts to call the API operations provided by the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, configure the request headers that are described in the following table for access control. Otherwise, an error that indicates insufficient permissions may occur.

Warning

The AccessKey pair of a personal account is managed by the permission system of the Apsara Uni-manager Management Console. If you need to obtain the AccessKey pair of an organization, make sure that your operation is allowed by an administrator. This improves security.

Request header	Description
x-acs-regionid	The ID of the region. Example: cn-hangzhou-*
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For information about how to obtain an organization ID, see the <i>GetOrganizationList</i> section in <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For information about how to obtain a resource set ID, see the <i>ListResourceGroup</i> section in <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-instanceid	The ID of the instance on which you want to perform the operation. For example, you can use the ID of an ECS instance obtained from the ECS instance list.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, click the profile picture and then click **User Information**.
3. In the **AccessKey Pair** section, view the AccessKey pair of your account.

Note

The AccessKey pair consists of an AccessKey ID and an AccessKey secret. These credentials provide you with full permissions on Apsara Stack resources within your account. You must keep the AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization navigation tree, click the name of the level-1 organization whose AccessKey pair you want to obtain.
5. Click **Management AccessKey**.
6. In the Management AccessKey dialog box, view the AccessKey pair of the organization.

3.2.4. Obtain access credentials assigned by STS

This topic describes how to use STS to call an EBS API operation.

Background information

STS allows you to use temporary credentials to access Apsara Stack resources. You can use STS to call API operations in Apsara Stack.

To call API operations in Apsara Stack by using STS, you must obtain an AccessKey ID, an AccessKey secret, and a security token assigned by STS. For more information about STS, see [What is STS?](#)

Procedure

1. Obtain the ARN of your `RAM role`.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. In the upper-right corner of the homepage, click the profile picture and then click **User Information**.

iii. Click **View Current Role Policy**.

In the **View Policies of Current Role** dialog box, view the ARN of your `RAM role`.

2. Obtain the access credentials assigned by STS.

 Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Variables:

- <yourRegionID>: Replace the variable with the actual region ID.
- <yourAccessKeyID>: Replace the variable with your AccessKey ID.
- <yourAccessKeySecret>: Replace the variable with your AccessKey secret.
- <yourRoleSessionName>: Replace the variable with the actual session name. For more information, see [AssumeRole](#).
- <yourRoleArn>: Replace the variable with your RAM role.
- <sts.aliyuns.com>: Replace the variable with the actual endpoint of STS. For more information, see [AssumeRole](#).

```
public static void main(String[] args) {
    // Create a DefaultAcsClient instance and initialize the instance.
    // Replace <yourRegionID>, <yourAccessKeyID>, and <yourAccessKeySecret> with your region ID, AccessKey ID, and AccessKey secret.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKeyID>", "<yourAccessKeySecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    String assumeRole = callAssumeRole(client);
    if (StringUtils.isEmpty(assumeRole)) return;
    JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
    JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
    // Obtain the access credentials assigned by STS.
    String stsAK = credentials.getString("AccessKeyId");
    String stsSK = credentials.getString("AccessKeySecret");
    String stsToken = credentials.getString("SecurityToken");
}

private static String callAssumeRole(IAcsClient client) {
    CommonRequest request = new CommonRequest();
    // Replace sts.aliyuns.com with the actual endpoint of STS. The endpoint of STS is in the sts-vpc.${global:region}.${global:internet-domain}
    // format.
    request.setSysDomain("sts.aliyuns.com");
    request.setSysProduct("Sts");
    request.setSysAction("AssumeRole");
    request.setSysVersion("2015-04-01");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace <yourRoleSessionName> with the name of the current session.
    request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
    // Replace <yourRoleArn> with the ARN of the RAM role that you obtained in the preceding step.
    request.putQueryParameter("RoleArn", "<yourRoleArn>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        return response.getData();
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
    return null;
}
```

3. Use the access credentials assigned by STS to call an operation of a service.

 Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify **2014-05-26** as the API version and **Ecs** as the service name.
- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify **2021-07-30** as the API version and **ebs** as the service name.

The following sample code provides an example in which the `DescribeDisks` operation is called to query the information about cloud disks:

```

public static void main(String[] args) {
    // Replace <yourRegionID> with the actual region ID.
    // Replace <yourStsAccessKeyID> and <yourStsAccessKeySecret> with the AccessKey ID and AccessKey secret assigned by STS.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourStsAccessKey>", "<yourStsAccessSecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    CommonRequest request = new CommonRequest();
    request.setSysDomain("ecs.aliyuns.com");
    // Specify Ecs as the service name.
    request.setSysProduct("Ecs");
    // Specify the name of the API operation. It must be the same as the name of the API operation in API Reference.
    request.setSysAction("DescribeDisks");
    // Specify 2014-05-26 as the API version.
    request.setSysVersion("2014-05-26");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace <yourRegionID> with the actual region ID.
    request.putHeadParameter("x-acs-regionid", "<yourRegionID>");
    // Replace <yourStsToken> with the security token assigned by STS.
    request.putQueryParameter("SecurityToken", "<yourStsToken>");
    // Optional. Replace <yourResourceGroupID> with the actual resource set ID.
    // request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");
    // Required. Replace <yourOrganizationID> with the actual organization ID.
    request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        System.out.println(response.getData());
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
}

```

The following sample code provides an example in which the DescribeDiskReplicaPairs operation is called to query the information about replication pairs:

```

public static void main(String[] args) {
    // Replace <yourRegionID> with the actual region ID.
    // Replace <yourStsAccessKey> and <yourStsAccessSecret> with the AccessKey ID and AccessKey secret assigned by STS.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourStsAccessKey>", "<yourStsAccessSecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    CommonRequest request = new CommonRequest();
    request.setSysDomain("ebs.aliyuns.com");
    // Specify ebs as the service name.
    request.setSysProduct("ebs");
    // Specify the name of the API operation. It must be the same as the name of the API operation in API Reference.
    request.setSysAction("DescribeDiskReplicaPairs");
    // Specify 2021-07-30 as the API version.
    request.setSysVersion("2021-07-30");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace <yourRegionID> with the actual region ID.
    request.putHeadParameter("x-acs-regionid", "<yourRegionID>");
    // Replace <yourStsToken> with the security token assigned by STS.
    request.putQueryParameter("SecurityToken", "<yourStsToken>");
    // Optional. Replace <yourResourceGroupID> with the actual resource set ID.
    // request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");
    // Required. Replace <yourOrganizationID> with the actual organization ID.
    request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        System.out.println(response.getData());
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
}

```

3.2.5. Obtain the values of common request headers

You must set request headers when you call API operations in Apsara Stack. This topic describes the request headers and how to obtain the values of the request headers. This topic also describes the response parameters of specific headers after you call API operations in Apsara Stack.

Request headers

Request header	Description
x-acs-regionid	The ID of the region. For more information about how to obtain a region ID, see Obtain a region ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain an organization ID, see Obtain an organization ID .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain a resource set ID, see Obtain a resource set ID .
x-acs-instanceid	The ID of the instance on which you want to perform the operation. For more information about how to obtain an instance ID, see Obtain an instance ID .

Obtain a region ID

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Regions**.
4. In the organization navigation tree, click the name of the organization whose region ID you want to obtain.
5. Obtain the region ID in the **Regions** section on the right.

Obtain an organization ID

You can call the `GetOrganizationList` operation to obtain the organization ID. For more information, see the *GetOrganizationList* section in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a resource set ID

You can call the `ListResourceGroup` operation to obtain the resource set ID. For more information, see the *ListResourceGroup* section in *Apsara Uni-manager Management Console Developer Guide*.

Obtain an instance ID

You can obtain the ID of an instance on related pages of the corresponding cloud service in the Apsara Uni-manager Management Console. The steps vary depending on the feature that you want to use.

To use a feature related to cloud disks or snapshots, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Elastic Compute Service**.
3. In the left-side navigation pane, click **Disks**, **Snapshots**, or **Automatic Snapshot Policies**.
4. On the page that appears, obtain a resource ID, such as a cloud disk or snapshot ID.

To use the asynchronous replication or replication pair-consistent group feature, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Elastic Block Storage**.
3. In the left-side navigation pane, click **EBS Async Replication** or **Replication Pair-Consistent Group**.
4. On the page that appears, obtain a resource ID, such as a replication pair or replication pair-consistent group ID.

Response parameters

The following table describes the response parameters that are returned by default when you call operations to query the instances of cloud services.

Parameter	Description
ResourceGroupName	The name of the resource set.
ResourceGroup	The ID of the resource set.
DepartmentName	The name of the organization.
Department	The ID of the organization.

3.2.6. Obtain an endpoint

This topic describes how to obtain the endpoints of ECS and EBS.

Background information

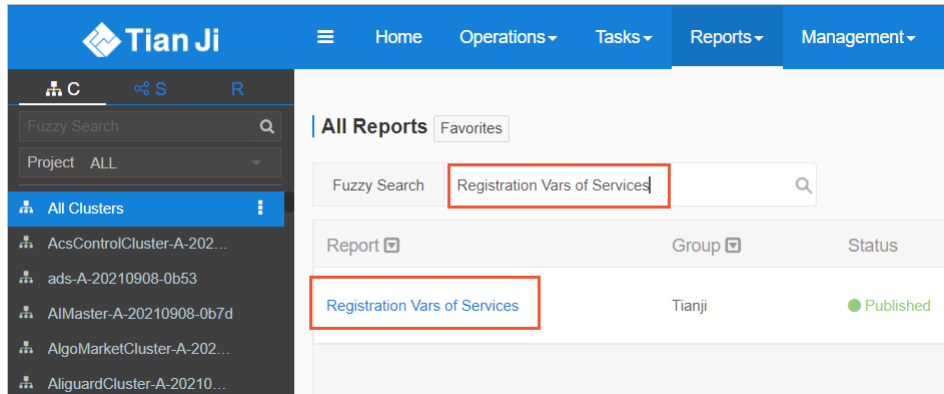
If you want to call API operations in Apsara Stack by using the POP gateway, you must specify the endpoint of the corresponding cloud service. You can obtain an endpoint from onsite Apsara Stack O&M engineers. You can also perform the following operations to obtain an endpoint.

The endpoint that you need to obtain varies depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, obtain the endpoint of ECS. For more information, see [Obtain the endpoint of ECS](#).
- If you want to use the asynchronous replication or replication pair-consistent group feature, obtain the endpoint of EBS. For more information, see [Obtain the endpoint of EBS](#).

Obtain the endpoint of ECS

1. Log on to the Apsara Infrastructure Management console.
 - i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, choose **Products > Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.
3. On the **All Reports** page, search for `Registration Vars of Services`. In the report list that appears, click the name of the report.

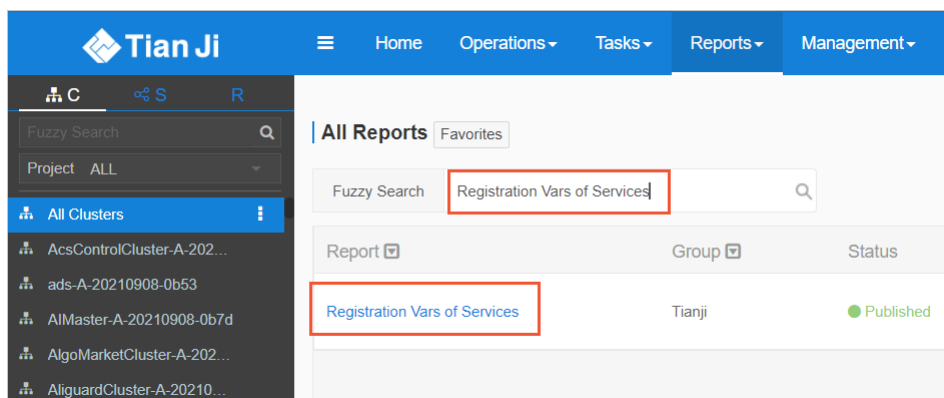


4. On the **Registration Vars of Services** page, click the icon next to **Service** and search for `ECS`.
5. Find the `ecs-yaochi` service, right-click the value in the **Service Registration** column, and then select **Show More**.

In the **Details** dialog box, view the value of `ecs-internal.openapi.endpoint`. The value indicates the `endpoint` of ECS.

Obtain the endpoint of EBS

1. Log on to the Apsara Infrastructure Management console.
 - i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, choose **Products > Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.
3. On the **All Reports** page, search for `Registration Vars of Services`. In the report list that appears, click the name of the report.



4. On the **Registration Vars of Services** page, click the icon next to **Service** and search for `EBS`.
5. Find the `ebstiangangapi` service, right-click the value in the **Service Registration** column, and then select **Show More**.

In the **Details** dialog box, view the value of `ebstiangang.openapi.endpoint`. The value indicates the `endpoint` of EBS.

3.2.7. Obtain an SDK

If you want to call API operations in Apsara Stack by using the POP gateway, you must obtain an ECS or EBS SDK. This topic describes how to obtain an SDK. The SDK that you need to obtain varies depending on the EBS feature that you want to use. We recommend that you obtain both an ECS SDK and an EBS SDK.

- If you want to use a feature related to cloud disks or snapshots, obtain an ECS SDK.

An ECS SDK encapsulates the API released on May 26, 2014. The SDK identifies callers based on their AccessKey pairs and provides features such as automatic signing to allow you to create and manage resources by calling API operations.

To use the corresponding features, you must install both an ECS SDK and the Alibaba Cloud core library. For example, to use ECS SDK for Java, you must download `aliyun-java-sdk-core` and `aliyun-java-sdk-ecs`.

- If you want to use the asynchronous replication or replication pair-consistent group feature, obtain an EBS SDK.

An EBS SDK encapsulates the API released on July 30, 2021. The SDK identifies callers based on their AccessKey pairs and provides features such as automatic signing to allow you to create and manage resources by calling API operations.

To use the corresponding features, you must install both an EBS SDK and the Alibaba Cloud core library. For example, to use EBS SDK for Java, you must download `aliyun-java-sdk-core` and `aliyun-java-sdk-ebs`.

An SDK includes code and examples. You can create cloud applications by using an SDK for the programming language that you require. The SDKs for Java, Python, and Go are provided. You can select an SDK from [Alibaba Cloud SDKs](#) based on your business requirements.

Note

We recommend that you use an SDK for Java, Python, or Go. If you use SDKs for other programming languages, the SDKs may be incompatible with your system.

We recommend that you use the SDK core library that is provided on the Alibaba Cloud official website. The SDK core library is ideal for all cloud services and is compatible with Apsara Stack.

3.3. Java SDK call example

3.3.1. Quick start

This topic describes how to obtain and use ECS SDK for Java and EBS SDK for Java to call API operations.

Environment requirements

- An Apsara Stack tenant account that has the required permissions and an AccessKey pair that consists of an AccessKey ID and an AccessKey secret are obtained. For more information, see [Obtain access credentials assigned by RAM](#).
- The endpoint of the corresponding service is obtained. The service varies depending on the feature that you want to use. For more information, see [Obtain an endpoint](#).
- Java Development Kit (JDK) 1.8 or later is installed.

Install ECS or EBS SDK for Java

If you use Apache Maven to manage a Java project, you need to only import the JAR file to the `pom.xml` file of the project.

To use an SDK of a service, you must install both the SDK core library and the SDK. We recommend that you use V4.5.17 or later of the core library. In this example, ECS SDK for Java V4.23.2 and EBS SDK for Java V1.0.3 are used. For more information, see [Install ECS or EBS SDK for Java](#).

Use ECS or EBS SDK for Java

To use ECS or EBS SDK for Java, perform the following steps:

1. Create a `DefaultAcsClient` instance and initialize the instance.
2. Create an API request and set the required parameters.
3. Send the API request and handle the response or exceptions.

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify `2014-05-26` as the API version and `Ecs` as the service name.
- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify `2021-07-30` as the API version and `ebs` as the service name.

The following sample code provides an example in which the `DescribeDisks` operation is called to query the information about cloud disks:

Note

You must replace the variables in the following code with actual values. Delete the angle brackets (`<>`) before you submit the code.

Variables:

- `<yourRegionID>`: Replace the variable with the actual region ID.
- `<yourAccessKeyId>`: Replace the variable with your AccessKey ID.
- `<yourAccessKeySecret>`: Replace the variable with your AccessKey secret.
- `<Product>`: Replace the variable with the name of the required service.
- `<Endpoint>`: Replace the variable with the endpoint of the required service.
- `<readTimeout>`: Replace the variable with a read timeout period based on your business requirements. Unit: milliseconds.
- `<connectionTimeout>`: Replace the variable with a connection timeout period based on your business requirements. Unit: milliseconds.
- `<maxRequestsPerHost>`: Replace the variable with the maximum number of requests that are supported by each host.
- `<maxRequest>`: Replace the variable with the maximum number of requests.
- `<http://192.168.XX:33XX>`: Replace the variable with the IP address and port of the proxy that you use.

```

import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.ecs.model.v20140526.*;

public class Main {
    public static void main(String[] args) {
        // Create a DefaultAcsClient instance and initialize the instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace the variable with the actual region ID.
            "<yourAccessKeyID>", // Replace the variable with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace the variable with your AccessKey secret.
        // Specify the endpoint of the required service. Replace the variables with actual values. Specify Ecs as the service name.
        profile.addEndpoint("<yourRegionID>", "Ecs", "<Endpoint>");
        // Set the request parameters.
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setIgnoreSSLCerts(true); // Specifies whether to disable certificate validation.
        clientConfig.setReadTimeoutMillis(<readTimeout>); // Optional. The read timeout period. Unit: milliseconds.
        clientConfig.setConnectionTimeoutMillis(<connectionTimeout>); // Optional. The connection timeout period. Unit: milliseconds.
        clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>); // Optional. The maximum number of requests that are supported by each host.
        clientConfig.setMaxRequests(<maxRequests>); // Optional. The maximum number of requests.
        clientConfig.setProtocolType(ProtocolType.HTTP); // Optional. The request protocol. Default value: HTTP.
        /* Optional. Specify the IP address and port of the proxy.
        clientConfig.setHttpProxy("<http://192.168.XX:33XX>");
        Or
        clientConfig.setHttpsProxy("<https://192.168.XX:33XX>");
        */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and set the required parameters.
        DescribeDisksRequest request = new DescribeDisksRequest()
            .setRegionId("cn-qingdao")
            .setZoneId("cn-qingdao-zone-1");
        // Specify the organization ID and resource set ID. This way, resources can be isolated. Modify the values of the headers based on your business requirements.
        request.putHeaderParameter("x-acs-organizationid", "1");
        request.putHeaderParameter("x-acs-resourcegroupid", "11");
        // Send the API request and handle the response or exceptions.
        DescribeDisksResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println(response.getRequestId());
            System.out.println(response.getTotalCount());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}

```

The following sample code provides an example in which the `DescribeDiskReplicaPairs` operation is called to query the information about replication pairs:

```

package com.aliyun.sample;
import com.aliyun.tea.*;
import com.aliyun.ebs20210730.*;
import com.aliyun.ebs20210730.models.*;
import com.aliyun.teaopenapi.*;
import com.aliyun.teaopenapi.models.*;

public class Sample {

    /**
     * Use your AccessKey ID and AccessKey secret to initialize the client.
     * @param accessKeyId
     * @param accessKeySecret
     * @return Client
     * @throws Exception
     */
    public static com.aliyun.ebs20210730.Client createClient(String accessKeyId, String accessKeySecret) throws Exception {
        Config config = new Config()
            // Your AccessKey ID.
            .setAccessKeyId(accessKeyId)
            // Your AccessKey secret.
            .setAccessKeySecret(accessKeySecret);
        // Specify the endpoint of the required service. Replace the variables with actual values.
        config.endpoint = "ebs.cn-beijing.aliyuncs.com";
        return new com.aliyun.ebs20210730.Client(config);
    }

    public static void main(String[] args_) throws Exception {
        java.util.List<String> args = java.util.Arrays.asList(args_);
        com.aliyun.ebs20210730.Client client = Sample.createClient("<yourAccessKeyId>", "<yourAccessKeySecret>");
        DescribeDiskReplicaPairsRequest describeDiskReplicaPairsRequest = new DescribeDiskReplicaPairsRequest()
            // Specify the actual region ID.
            .setRegionId("cn-beijing");
        // Run the following sample code to obtain the response parameter values of the API operation.
        DescribeDiskReplicaPairsResponse resp = client.describeDiskReplicaPairs(describeDiskReplicaPairsRequest);
        for (int i = 0; i < resp.body.getReplicaPairs().size(); i++) {
            DescribeDiskReplicaPairsResponseBody.DescribeDiskReplicaPairsResponseBody.ReplicaPairs pair
                = resp.body.getReplicaPairs().get(i);
            System.out.println(pair.getReplicaPairId());
            System.out.println(pair.getStatus());
        }
    }
}

```

3.3.2. User Manual

3.3.2.1. Install ECS or EBS SDK for Java

This topic describes how to install and use ECS SDK for Java and EBS SDK for Java.

ECS or EBS SDK for Java supports JDK 1.8 or later. You can install ECS or EBS SDK for Java by adding Maven dependencies or downloading the SDK package. For more information, see [Alibaba Cloud SDKs](#).

To use EBS features, you must install both the core library of Alibaba Cloud SDK for Java and ECS or EBS SDK for Java. We recommend that you use V4.5.17 or later of the core library. In this example, ECS SDK for Java V4.23.2 and EBS SDK for Java V1.0.3 are used.

The SDK that you need to install varies depending on the EBS feature that you want to use. We recommend that you install both ECS SDK for Java and EBS SDK for Java.

- To use a feature related to cloud disks or snapshots, you must install the core library of Alibaba Cloud SDK for Java and ECS SDK for Java. In addition, you must add the following dependencies to the pom.xml file. For more information about the SDK version, see [Alibaba Cloud SDKs](#).

```

<!-- Alibaba Cloud SDK for Java -->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.17</version>
</dependency>
<!-- ECS Java SDK -->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-ecs</artifactId>
  <version>4.23.2</version>
</dependency>

```

- To use the asynchronous replication or replication pair-consistent group feature, you must install the core library of Alibaba Cloud SDK for Java and EBS SDK for Java. In addition, you must add the following dependencies to the pom.xml file. For more information about the SDK version, see [Alibaba Cloud SDKs](#).

```
<!-- Alibaba Cloud SDK for Java -->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.17</version>
</dependency>
<!-- EBS Java SDK -->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>ebs20210730</artifactId>
  <version>1.0.3</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>tea-openapi</artifactId>
  <version>0.2.2</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>tea-console</artifactId>
  <version>0.0.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>tea-util</artifactId>
  <version>0.2.13</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>tea</artifactId>
  <version>1.1.14</version>
</dependency>
```

3.3.2.2. Configure authentication credentials

When you use Alibaba Cloud SDKs to access Apsara Stack services, you must provide an Apsara Stack tenant account for authentication. This topic describes how to configure authentication credentials.

Alibaba Cloud SDK for Java allows you to complete the authentication process by using an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain access credentials assigned by RAM](#).

If you want to use your AccessKey pair to access Apsara Stack, you must configure the AccessKey pair as credentials when you initialize a client instance.

Note

Do not disclose any code that contains your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.

```
DefaultProfile profile = DefaultProfile.getProfile(
    "<yourRegionID>",           // Replace the variable with the actual region ID.
    "<yourAccessKeyID>",       // Replace the variable with your AccessKey ID.
    "<yourAccessKeySecret>"); // Replace the variable with your AccessKey secret.
```

Note

You must replace the variables in the preceding code with actual values. Delete the angle brackets (<>) before you submit the code.

3.3.2.3. Configure connection requests

This topic describes how to configure a connection request by using the HttpClientConfig object.

You can use the following code to configure the HttpClientConfig object. We recommend that you select and configure the following parameters based on your business requirements.

Note

You must use actual values to replace the variables in the following sample code. Do not retain the angle brackets (<>) in the final code.

The following list describes the variables:

- <readTimeout>: Specify the read timeout period based on your business requirements. Unit: milliseconds.
- <connectionTimeout>: Specify the connection timeout period based on your business requirements. Unit: milliseconds.
- <maxRequestsPerHost>: Specify the maximum number of requests that are supported by each server based on your business requirements. The supported data type is INTEGER.
- <maxRequest>: Specify the maximum number of requests based on your business requirements. The supported data type is INTEGER.
- <http://192.168.XX.XX:33XX>: Replace it with the actual IP address and port of the proxy.

```
// Obtain the HttpClientConfig object.
HttpClientConfig clientConfig = HttpClientConfig.getDefault();
// Specify whether to ignore certificates.
clientConfig.setIgnoreSSLCerts(true);
// (Optional) Specify the read timeout period. Unit: milliseconds.
clientConfig.setReadTimeoutMillis(<readTimeout>);
// (Optional) Specify the connection timeout period. Unit: milliseconds.
clientConfig.setConnectionTimeoutMillis(<connectionTimeout>);
// (Optional) Specify the maximum number of requests that are supported by each server. The supported data type is INTEGER.
clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>);
// (Optional) Specify the maximum number of requests. The supported data type is INTEGER.
clientConfig.setMaxRequests(<maxRequests>);
// (Optional) Specify the protocol type.
clientConfig.setProtocolType(ProtocolType.HTTP);
/** (Optional) Specify whether to configure the proxy based on your business requirements. The following code is available:
clientConfig.setHttpProxy("<http://192.168.XX.XX:33XX>");
Or
clientConfig.setHttpsProxy("<https://192.168.XX.XX:33XX>");
Replace https://192.168.XX.XX:33XX with the actual IP address and port of the proxy.
*/
profile.setHttpClientConfig(clientConfig);
```

3.3.2.4. Initiate a request

This topic describes how to use ECS SDK for Java and EBS SDK for Java to initiate requests.

Use ECS SDK for Java to initiate a request

1. Create an `AcsClient` instance.

```
IACSClient client = new DefaultAcsClient(profile);
```

2. Create a request.

The naming convention for requests is `#{apiName}Request`. `#{apiName}` indicates the API operation name. Example: `DescribeDisksRequest`.

When the SDKs of multiple services are used, different requests may have the same name. Differentiate the requests by SDK.

```
DescribeDisksRequest request;
// Set service-related parameters.
request.setPageSize(10);
request.setPageNumber(1);
// Specify the organization ID and resource set ID. This way, resources can be isolated. Modify the values of the headers based on your business requirements.
request.putHeaderParameter("x-acs-organizationid", "1");
request.putHeaderParameter("x-acs-resourcegroupid", "");
```

3. Initiate the request and handle the response.

```
DescribeDisksResponse response;
try {
    response = client.getAcsResponse(request);
    System.out.println(response.getRequestId());
    System.out.println(response.getTotalCount());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
```

In normal cases, all returned fields are deserialized into the response. You can call `response.getXXX()` to obtain the values of the response fields.

```
Integer count = response.getTotalCount();
```

If an exception occurs or you want to obtain the original HTTP response, call `doAction()` to obtain the original response.

```
HttpResponse response = client.doAction(request);
```

Use EBS SDK for Java to initiate a request

1. Create a client instance.

```
public static com.aliyun.ebs20210730.Client createClient(String accessKeyId, String accessKeySecret) throws Exception {
    Config config = new Config()
        // Your AccessKey ID.
        .setAccessKeyId(accessKeyId)
        // Your AccessKey secret.
        .setAccessKeySecret(accessKeySecret);
    // Specify the endpoint of the required service. Replace the variables with actual values.
    config.endpoint = "ebs.cn-beijing.aliyuncs.com";
    return new com.aliyun.ebs20210730.Client(config);
}
```

2. Create a request.

```
com.aliyun.ebs20210730.Client client = Sample.createClient("accessKeyId", "accessKeySecret");
DescribeDiskReplicaPairsRequest describeDiskReplicaPairsRequest = new DescribeDiskReplicaPairsRequest()
    // Specify the actual region ID.
    .setRegionId("cn-beijing");
```

3. Initiate the request and handle the response.

```
DescribeDiskReplicaPairsResponse resp = client.describeDiskReplicaPairs(describeDiskReplicaPairsRequest);
for (int i = 0; i < resp.body.getReplicaPairs().size(); i++) {
    DescribeDiskReplicaPairsResponseBody.DescribeDiskReplicaPairsResponseBody.ReplicaPairs pair
        = resp.body.getReplicaPairs().get(i);
    System.out.println(pair.getReplicaPairId());
    System.out.println(pair.getStatus());
}
```

3.3.2.5. Troubleshooting

If the server or SDK encounters an error when you use SDK for Java, it returns an error response. The error response contains an error code and an error message.

In most cases, you do not need to handle the errors returned by SDK for Java. You need to only handle the errors returned by the server based on error messages.

- `ServerException` indicates a server error.
- `ClientException` indicates an SDK error.

For example, when the following error occurs, you can modify the AccessKey ID based on the error message to solve the problem:

```
com.aliyuncs.exceptions.ClientException: InvalidAccessKeyId.NotFound : Specified access key is not found.
```

If you want to handle SDK errors, refer to the following example to write your code:

```
try {
    FooResponse response = client.getAcsResponse(request);
    // Handle the response.
    // ...
} catch (ServerException e) {
    // You can add your logic to handle errors.
    // For example, print the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
    // If the error persists, you can submit a ticket with the request ID included.
    System.out.println("ResponseId=" + e.getRequestId());
} catch (ClientException e) {
    // You can add your logic to handle errors.
    // For example, print the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
}
```

3.3.2.6. Use CommonRequest to call API operations

If you want to call an ECS or EBS API operation without an SDK, or the request and response of the API operation that you call by using an SDK cannot meet your business requirements, you can use `CommonRequest` to call the API operation. You can use `CommonRequest` to call all API operations.

Benefits

`CommonRequest` provides the following benefits:

- Lightweight: You need to only download the core package. You do not need to download and install the SDKs of services.
- Easy to use: You can call the newly released API operations of a service without the need to update the SDK of the service.
- Fast iteration.

Use CommonRequest to call an API operation

If you want to send a `CommonRequest` request, you must obtain the values of the following parameters. You can obtain the values of these parameters from the related topics about ECS API operations.

- Domain: the endpoint of the service.
- Version: the version number of the API. Specify the version number in the YYYY-MM-DD format.

You can obtain the API version from the sections about common parameters in the related topics about API operations.

- Action: the name of the API operation that you want to call.

You must obtain the value of the `Action` parameter and specify the API operation in the `request.setSysAction("<Action>")` format.

For example, if you want to use `CommonRequest` to call the `RunInstances` operation, you must specify the API operation in the

```
request.setSysAction("RunInstances") format.
```

Example: Call an RPC API operation

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify `2014-05-26` as the API version and `Ecs` as the service name.

- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify **2021-07-30** as the API version and **ebs** as the service name.

The following sample code provides an example in which the `DescribeDisks` operation is called to query the information about cloud disks:

 Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Variables:

- <yourRegionID>: Replace the variable with the actual region ID.
- <yourAccessKeyID>: Replace the variable with your AccessKey ID.
- <yourAccessKeySecret>: Replace the variable with your AccessKey secret.
- <Endpoint>: Replace the variable with the endpoint of the required service.
- <yourSecurityToken>: Replace the variable with your security token.
- <yourResourceGroupID>: Replace the variable with the resource set ID.
- <yourOrganizationID>: Replace the variable with the organization ID.
- <yourInstanceID>: Replace the variable with the instance ID.
- <http://192.168.XX.XX:33XX>: Replace the variable with the IP address and port of the proxy that you use.


```

import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;

public class Sample {
    public static void main(String[] args) {
        // Create a DefaultAcsClient instance and initialize the instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace the variable with the actual region ID.
            "<yourAccessKeyID>", // Replace the variable with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace the variable with your AccessKey secret.
        // Specify the endpoint of the required service. Replace the variables with actual values.
        profile.addEndpoint("<Endpoint>");
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setProtocolType(ProtocolType.HTTP);
        /** Optional. Specify the IP address and port of the proxy.
        clientConfig.setHttpProxy("<http://192.168.XX.XX:33XX>");
        Or
        clientConfig.setHttpsProxy("<https://192.168.XX.XX:33XX>");
        Specify the IP address and port of the proxy.
        */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and set the required parameters.
        CommonRequest request = new CommonRequest();
        // Specify the endpoint of the required service.
        request.setDomain("ecs.aliyuncs.com");
        request.setSysProduct("Ecs");
        request.setVersion("2014-05-26");
        request.setAction("DescribeDisks");
        request.putQueryParameter("PageNumber", "1");
        request.putQueryParameter("PageSize", "30");
        request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");// Optional. Specify the resource set ID.
        request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");// Required. Specify the organization ID.
        request.putHeadParameter("x-acs-caller-sdk-source", "ApiTestDemo");// Required. Specify the source of the API call.
        request.putHeadParameter("x-acs-regionid", "<yourRegionID>");// Optional. Specify the region ID. If you use Apsara Stack Enterprise V3.15.0 or earlier, you must replace <yourRegionID> with the actual region ID.
        // request.putHeadParameter("x-acs-instanceid", "<yourInstanceID>");// Optional. Specify the instance ID. If you want to modify, delete, or view the details of a specific instance, you must replace <yourInstanceID> with the actual instance ID.
        /**
        If you use the access credentials assigned by STS to call an RPC API operation, you must specify the SecurityToken parameter in the request.
        Format:
        request.putQueryParameter("SecurityToken", "<yourSecurityToken>");
        Replace <yourSecurityToken> with your security token.
        */
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ServerException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ClientException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

The following sample code provides an example in which the DescribeDiskReplicaPairs operation is called to query the information about replication pairs:

```

import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;

public class Sample {
    public static void main(String[] args) {
        // Create a DefaultAcsClient instance and initialize the instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace the variable with the actual region ID.
            "<yourAccessKeyID>", // Replace the variable with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace the variable with your AccessKey secret.
        // Specify the endpoint of the required service. Replace the variables with actual values.
        profile.addEndpoint("ebs.cn-beijing.aliyuncs.com");
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setProtocolType(ProtocolType.HTTP);
        /** Optional. Specify the IP address and port of the proxy.
        clientConfig.setHttpProxy("<http://192.168.XX.XX:33XX>");
        Or
        clientConfig.setHttpsProxy("<https://192.168.XX.XX:33XX>");
        Specify the IP address and port of the proxy.
        */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and set the required parameters.
        CommonRequest request = new CommonRequest();
        // Specify the endpoint of the required service.
        request.setDomain("ebs.aliyuncs.com");
        request.setSysProduct("ebs");
        request.setVersion("2021-07-30");
        request.setAction("DescribeDiskReplicaPairs");
        request.putQueryParameter("RegionId", "cn-beijing");
        request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");// Optional. Specify the resource set ID.
        request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");// Required. Specify the organization ID.
        request.putHeadParameter("x-acs-caller-sdk-source", "ApiTestDemo");// Required. Specify the source of the API call.
        request.putHeadParameter("x-acs-regionid", "<yourRegionID>");// Optional. Specify the region ID. If you use Apsara Stack Enterprise V3.15.0 or earlier, you must replace <yourRegionID> with the actual region ID.
        // request.putHeadParameter("x-acs-instanceid", "<yourInstanceID>");// Optional. Specify the instance ID. If you want to modify, delete, or view the details of a specific instance, you must replace <yourInstanceID> with the actual instance ID.
        /**
        If you use the access credentials assigned by STS to call an RPC API operation, you must specify the SecurityToken parameter in the request.
        Format:
        request.putQueryParameter("SecurityToken", "<yourSecurityToken>");
        Replace <yourSecurityToken> with your security token.
        */
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ServerException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ClientException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

3.4. Python SDK call example

3.4.1. Quick start

This topic describes how to obtain and use ECS SDK for Python and EBS SDK for Python to call API operations.

Environment requirements

- An Apsara Stack tenant account that has the required permissions and an AccessKey pair that consists of an AccessKey ID and an AccessKey secret are obtained. For more information, see [Obtain access credentials assigned by RAM](#).
- The endpoint of the corresponding service is obtained. The service varies depending on the feature that you want to use. For more information, see [Obtain an endpoint](#).

Install ECS or EBS SDK for Python

To use EBS features, you must install both the core library of Alibaba Cloud SDK for Python and ECS or EBS SDK for Python.

The SDK that you need to install varies depending on the EBS feature that you want to use. We recommend that you install both ECS SDK for Python and EBS SDK for Python.

- To use a feature related to cloud disks or snapshots, you must install the core library of Alibaba Cloud SDK for Python and ECS SDK for Python.

- To use the asynchronous replication or replication pair-consistent group feature, you must install the core library of Alibaba Cloud SDK for Python and EBS SDK for Python.

To install ECS or EBS SDK for Python, perform the following steps:

1. Install the core library of Alibaba Cloud SDK.

- If you use Python 2.x, run the following command:

```
pip install aliyun-python-sdk-core
```

- If you use Python 3.x, run the following command:

```
pip3 install aliyun-python-sdk-core-v3
```

2. Install ECS or EBS SDK for Python based on your business requirements.

- Install ECS SDK for Python.

- If you use Python 2.x, run the following command:

```
pip install aliyun-python-sdk-ecs
```

- If you use Python 3.x, run the following command:

```
pip3 install aliyun-python-sdk-ecs
```

- Install EBS SDK for Python.

- If you use Python 2.x, run the following command:

```
pip install alibabacloud_ebs20210730==1.0.3
```

- If you use Python 3.x, run the following command:

```
pip3 install alibabacloud_ebs20210730==1.0.3
```

Use ECS or EBS SDK for Python

To use ECS or EBS SDK for Python, perform the following steps:

1. Create a client instance. When you create a client instance, you must obtain a region ID, your AccessKey ID, and your AccessKey secret.
2. Create an API request and set the required parameters.
3. Send the API request and handle the response or exceptions.

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify **2014-05-26** as the API version and **Ecs** as the service name.
- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify **2021-07-30** as the API version and **ebs** as the service name.

Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Variables:

- <yourRegionID>: Replace the variable with the actual region ID.
- <yourAccessKeyID>: Replace the variable with your AccessKey ID.
- <yourAccessKeySecret>: Replace the variable with your AccessKey secret.
- <ecs.aliyuns.com>: Replace the variable with the endpoint of the required service.
- <http://192.168.XX:33XX>: Replace the variable with the IP address and port of the proxy that you use.

The following sample code provides an example in which the DescribeDisks operation is called to query the information about cloud disks:

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdkecs.request.v20140526 import DescribeDisksRequest
# global config
region_id = '<yourRegionID>' # Specify the region ID.
access_key = '<yourAccessKeyID>' # Specify your AccessKey ID.
access_secret = '<yourAccessKeySecret>' # Specify your AccessKey secret.
http_proxy='<http://192.168.XX:33XX>', # Specify the IP address and port of the HTTP proxy.
# product config
product_name = 'ECS' # Specify the name of the required service.
endpoint = '<ecs.aliyuns.com>' # Specify the endpoint of the required service.
# create client
region_provider.modify_point(product_name, region_id, endpoint)
clt = client.AcsClient(access_key, access_secret, region_id)
# setup request
request = DescribeDisksRequest.DescribeDisksRequest()
request.set_accept_format('json') # Specify the response format.
# get response
response = clt.do_action_with_exception(request)
print(response)
```

The following sample code provides an example in which the DescribeDiskReplicaPairs operation is called to query the information about replication pairs:

```
# -*- coding: utf-8 -*-
# This file is auto-generated, don't edit it. Thanks.
import sys

from typing import List

from alibabacloud_ebs20210730.client import Client as ebs20210730Client
from alibabacloud_tea_openapi import models as open_api_models
from alibabacloud_ebs20210730 import models as ebs_20210730_models

class Sample:
    def __init__(self):
        pass

    @staticmethod
    def create_client(
        access_key_id: str,
        access_key_secret: str,
    ) -> ebs20210730Client:
        """
        Use your AccessKey ID and AccessKey secret to initialize the client.
        @param access_key_id:
        @param access_key_secret:
        @return: Client
        @throws Exception
        """
        config = open_api_models.Config(
            # Your AccessKey ID.
            access_key_id=access_key_id,
            # Your AccessKey secret.
            access_key_secret=access_key_secret
        )
        # Specify the endpoint of the required service.
        config.endpoint = f'ebs.cn-beijing.aliyuncs.com'
        return ebs20210730Client(config)

    @staticmethod
    def main(
        args: List[str],
    ) -> None:
        client = Sample.create_client('accessKeyId', 'accessKeySecret')
        describe_disk_replica_pairs_request = ebs_20210730_models.DescribeDiskReplicaPairsRequest(
            region_id='cn-beijing'
        )
        # Run the following sample code to obtain the response parameter values of the API operation.
        client.describe_disk_replica_pairs(describe_disk_replica_pairs_request)

    @staticmethod
    def main_async(
        args: List[str],
    ) -> None:
        client = Sample.create_client('accessKeyId', 'accessKeySecret')
        describe_disk_replica_pairs_request = ebs_20210730_models.DescribeDiskReplicaPairsRequest(
            region_id='cn-beijing'
        )
        # Run the following sample code to obtain the response parameter values of the API operation.
        await client.describe_disk_replica_pairs_async(describe_disk_replica_pairs_request)

if __name__ == '__main__':
    Sample.main(sys.argv[1:])
```

3.4.2. User Manual

3.4.2.1. Install ECS or EBS SDK for Python

This topic describes how to install and use ECS SDK for Python and EBS SDK for Python.

Background information

Alibaba Cloud SDK for Python supports Python 2.6.x, Python 2.7.x, Python 3.x, and later versions. You can install Alibaba Cloud SDK for Python by using pip.

Note

To use Alibaba Cloud SDK for Python to access an Apsara Stack service, you must install the core library of Alibaba Cloud SDK for Python. For example, if you want to use ECS SDK for Python, you must install both the core library of Alibaba Cloud SDK for Python and ECS SDK for Python.

Install ECS or EBS SDK for Python

To use EBS features, you must install both the core library of Alibaba Cloud SDK for Python and ECS or EBS SDK for Python.

The SDK that you need to install varies depending on the EBS feature that you want to use. We recommend that you install both ECS SDK for Python and EBS SDK for Python.

- To use a feature related to cloud disks or snapshots, you must install the core library of Alibaba Cloud SDK for Python and ECS SDK for Python.
- To use the asynchronous replication or replication pair-consistent group feature, you must install the core library of Alibaba Cloud SDK for Python and EBS SDK for Python.

To install ECS or EBS SDK for Python, perform the following steps:

1. Install the core library of Alibaba Cloud SDK.

- If you use Python 2.x, run the following command:

```
pip install aliyun-python-sdk-core
```

- If you use Python 3.x, run the following command:

```
pip3 install aliyun-python-sdk-core-v3
```

2. Install ECS or EBS SDK for Python based on your business requirements.

- Install ECS SDK for Python.

- If you use Python 2.x, run the following command:

```
pip install aliyun-python-sdk-ecs
```

- If you use Python 3.x, run the following command:

```
pip3 install aliyun-python-sdk-ecs
```

- Install EBS SDK for Python.

- If you use Python 2.x, run the following command:

```
pip install alibabacloud_ebs20210730==1.0.3
```

- If you use Python 3.x, run the following command:

```
pip3 install alibabacloud_ebs20210730==1.0.3
```

3.4.2.2. Configure authentication credentials

When you use Alibaba Cloud SDKs to access Apsara Stack services, you must provide an Apsara Stack tenant account for authentication. This topic describes how to configure authentication credentials.

Alibaba Cloud SDK for Python allows you to complete the authentication process by using an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain access credentials assigned by RAM](#).

If you want to use your AccessKey pair to access Apsara Stack, you must configure the AccessKey pair as credentials when you initialize a client instance.

Note

Do not disclose any code that contains your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.

```
from aliysdkcore import client
clt = client.AcsClient(access_key, acces_secret, region_id)
```

3.4.2.3. Set HTTPS requests

Apsara Stack SDK for Python allows you to initiate API requests over HTTP and HTTPS.

You can specify the HTTP or HTTPS protocol for a specific request, or set a global default protocol.

Note

The default HTTP or HTTPS protocol of a service takes precedence over the global default protocol.

Add OpenSSL support

The HTTPS protocol support of Apsara Stack SDK for Python is based on the OpenSSL support in Python. To use Apsara Stack SDK for Python to send HTTPS requests, you must add OpenSSL support in Python. OpenSSL supports TLS 1.2 and later.

Run the `python -c "import ssl"` command to check whether your Python environment supports OpenSSL. If the error message

```
ImportError: No module named ssl
```

is not returned, OpenSSL is already supported.

Otherwise, run the following command to install it:

```
pip install pyopenssl
```

Set the protocol to HTTP or HTTPS for a single request

The following sample code provides an example on how to set HTTPS as the protocol for a request:

```
request = DescribeInstancesRequest()
request.set_protocol_type("https")
# Set the value to https or http.
```

Set a global default protocol

The following sample code provides an example on how to set a global default protocol:

```
import aliyunsdkcore.request
aliyun_sdkcore.request.set_default_protocol_type("https")
# Create a request and call client.do_action_with_exception() to send the request.
```

3.4.2.4. Initiate a request

This topic describes how to use ECS SDK for Python and EBS SDK for Python to initiate requests.

Use ECS SDK for Python to initiate a request

1. Import ECS SDK for Python.

```
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdkecs.request.v20140526 import DescribeDiskRequest
```

2. Create an AcsClient instance.

```
client.region_provider.modify_point(product_name, region_id, endpoint)
clt = client.AcsClient(access_key, access_secret, region_id)
```

3. Create a request object.

```
request = DescribeDiskRequest.DescribeDiskRequest()
request.set_accept_format('json')
```

4. Initiate the request and handle the response.

```
response = clt.do_action_with_exception(request)
print(response)
```

Use EBS SDK for Python to initiate a request

1. Import EBS SDK for Python.

```
from alibabacloud_ebs20210730.client import Client as ebs20210730Client
from alibabacloud_tea_openapi import models as open_api_models
from alibabacloud_ebs20210730 import models as ebs_20210730_models
```

2. Create a client instance.

```
@staticmethod
def create_client(
    access_key_id: str,
    access_key_secret: str,
) -> ebs20210730Client:
    """
    Use your AccessKey ID and AccessKey secret to initialize the client.
    @param access_key_id:
    @param access_key_secret:
    @return: Client
    @throws Exception
    """
    config = open_api_models.Config(
        # Your AccessKey ID.
        access_key_id=access_key_id,
        # Your AccessKey secret.
        access_key_secret=access_key_secret
    )
    # Specify the endpoint of the required service.
    config.endpoint = f'ebs.cn-beijing.aliyuncs.com'
    return ebs20210730Client(config)
```

3. Create a request.

```
describe_disk_replica_pairs_request = ebs_20210730_models.DescribeDiskReplicaPairsRequest(
    region_id='cn-beijing'
)
```

4. Initiate the request and handle the response.

```
# Run the following sample code to obtain the response parameter values of the API operation.
response = client.describe_disk_replica_pairs(describe_disk_replica_pairs_request)
```

3.4.2.5. Handle errors

If an error occurs on the server side or SDK side when you use the Python SDK, the Python SDK returns the corresponding exception message. An exception message contains the detailed error information, namely the error code and the error message.

Generally, you are not required to handle the errors returned by the Python SDK. You are only required to handle the errors on the server side according to the error message.

- `ServerException` is the error message on the server side.
- `ClientException` is the error message on the SDK side.

For example, if the following error occurs, modify the ID of `AccessKey` according to the detailed error message.

```
aliyunSdkcore.acs\_exception.exceptions.ServerException: HTTP Status: 404 Error:InvalidAccessKeyId.NotFound Specified access key is not found.
```

To handle the errors returned by the Python SDK, see the following code example for reference:

```
try:
    response = client.do_action_with_exception(request)
except ServerException as e:
    # Add your own error handling logic here.
    # For example, print the detailed error message.
    print e.get_http_status()
    print e.get_error_code()
    print e.get_error_msg()
```

3.4.2.6. Use CommonRequest to call API operations

If you want to call an ECS or EBS API operation without an SDK, or the request and response of the API operation that you call by using an SDK cannot meet your business requirements, you can use `CommonRequest` to call the API operation. You can use `CommonRequest` to call all API operations.

Benefits

`CommonRequest` provides the following benefits:

- **Lightweight:** You only need to download the core package. You do not need to download and install SDKs of Apsara Stack services.
- **Easy to use:** You can call the newly released API operations of a service without the need to update the SDK of the service.
- **Fast in iteration.**

Use CommonRequest to call an API operation

If you want to send a `CommonRequest` request, you must obtain the values of the following parameters. You can obtain the values of these parameters from the related topics about API operations.

- **Domain:** the endpoint of the service.
- **Version:** the version number of the API. Specify the version number in the YYYY-MM-DD format.
You can obtain the API version from the sections about common parameters in the related topics about API operations.
- **Action:** the name of the API operation that you want to call.

You must obtain the value of the `Action` parameter and specify the API operation in the `request.set_action_name('Action')` format.

For example, if you want to use `CommonRequest` to call the `RunInstances` operation, you must specify the API operation in the

```
request.set_action_name('RunInstances')
```

 format.

Example: Call an RPC API operation

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify **2014-05-26** as the API version and **Ecs** as the service name.
- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify **2021-07-30** as the API version and **ebs** as the service name.

The following sample code provides an example in which the `DescribeDisks` operation is called to query the information about cloud disks:

Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Variables:

- `<yourRegionID>`: Replace the variable with the actual region ID.
- `<yourAccessKeyID>`: Replace the variable with your AccessKey ID.
- `<yourAccessKeySecret>`: Replace the variable with your AccessKey secret.
- `<yourSecurityToken>`: Replace the variable with your security token.
- `<yourResourceGroupID>`: Replace the variable with the resource set ID.
- `<yourOrganizationID>`: Replace the variable with the organization ID.
- `<yourInstanceID>`: Replace the variable with the instance ID.
- `<ecs.aliyuncs.com>`: Replace the variable with the endpoint of the required service.

```
#!/usr/bin/env python
# coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.request import CommonRequest
# Specify the AccessKey ID, AccessKey secret, and region ID.
client = AcsClient('{<yourAccessKeyId>}', '{<yourAccessKeySecret>}', '{<yourRegionID>}')
request = CommonRequest()
request.set_product('Ecs')
request.set_action_name('DescribeDisks')
request.set_version('2014-05-26')
request.set_domain('ecs.aliyuncs.com') # Specify the endpoint of the required service.
# or:
# request = CommonRequest(product='Ecs', action_name='DescribeInstanceStatus', version='2014-05-26', domain='ecs.aliyuncs.com')
#####
# Specify the request protocol. Valid values: https and http. Default value: http.
# The following example shows how to set the request protocol to HTTPS:
# request.set_protocol_type('https')
#####
request.set_accept_format('json')
request.set_method('POST')
request.add_query_param('PageNumber', '1')
request.add_query_param('PageSize', '30')
#####
# If you use the access credentials assigned by STS to call an RPC API operation, you must specify the SecurityToken parameter in the request.
# Format:
# request.add_query_param("SecurityToken", "<yourSecurityToken>");
# Specify your security token.
#####
# Optional. Specify the resource set ID.
request.add_header("x-acs-resourcegroupid", "<yourResourceGroupID>")
# Required. Specify the organization ID.
request.add_header("x-acs-organizationid", "<yourOrganizationID>")
# Required. Specify the source of the API call.
request.add_header("x-acs-caller-sdk-source", "ApiTestDemo")
# Optional. Specify the region ID. If you use Apsara Stack Enterprise V3.15.0 or earlier, you must replace <yourRegionID> with the actual region ID.
request.add_header("x-acs-regionid", "<yourRegionID>")
# Optional. Specify the instance ID. If you want to modify, delete, or view the details of a specific instance, you must replace <yourInstanceID> with the actual instance ID.
# request.add_header("x-acs-instanceid", "<yourInstanceID>")

response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding = 'utf-8'))
```

The following sample code provides an example in which the DescribeDiskReplicaPairs operation is called to query the information about replication pairs:


```
#!/usr/bin/env python
# coding=utf-8
from aliynsdkcore.client import AcsClient
from aliynsdkcore.request import CommonRequest
# Specify the AccessKey ID, AccessKey secret, and region ID.
client = AcsClient('{<yourAccessKeyId>}', '{<yourAccessKeySecret>}', '{<yourRegionID>}')
request = CommonRequest()
request.set_product('ebs')
request.set_action_name('DescribeDiskReplicaPairs')
request.set_version('2021-07-30')
request.set_domain('<ebs.aliyuncs.com>') # Specify the endpoint of the required service.
# or:
# request = CommonRequest(product='Ecs', action_name='DescribeInstanceStatus', version='2014-05-26', domain='ecs.aliyuncs.com')
#####
# Specify the request protocol. Valid values: https and http. Default value: http.
# The following example shows how to set the request protocol to HTTPS:
# request.set_protocol_type('https')
#####
request.set_accept_format('json')
request.set_method('POST')
request.add_query_param('RegionId', 'cn-beijing')
#####
# If you use the access credentials assigned by STS to call an RPC API operation, you must specify the SecurityToken parameter in the request.
# Format:
# request.add_query_param("SecurityToken", "<yourSecurityToken>");
# Specify your security token.
#####
# Optional. Specify the resource set ID.
request.add_header("x-acs-resourcegroupid", "<yourResourceGroupID>")
# Required. Specify the organization ID.
request.add_header("x-acs-organizationid", "<yourOrganizationID>")
# Required. Specify the source of the API call.
request.add_header("x-acs-caller-sdk-source", "ApiTestDemo")
# Optional. Specify the region ID. If you use Apsara Stack Enterprise V3.15.0 or earlier, you must replace <yourRegionID> with the actual region ID.
request.add_header("x-acs-regionid", "<yourRegionID>")
# Optional. Specify the instance ID. If you want to modify, delete, or view the details of a specific instance, you must replace <yourInstanceID> with the actual instance ID.
# request.add_header("x-acs-instanceid", "<yourInstanceID>")

response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding = 'utf-8'))
```

3.5. Go SDK call example

3.5.1. Quick start

Alibaba Cloud SDK for Go allows you to access services without complex coding. This topic describes how to download and use ECS SDK for Go and EBS SDK for Go.

Make preparations

- Obtain an authorized account and an AccessKey pair, consisting of an AccessKey ID and an AccessKey secret. For more information, see [Obtain access credentials assigned by RAM](#).
- Obtain the endpoint of the service that is involved based on the feature that you want to use. For more information, see [Obtain an endpoint](#).

Install the core package of Alibaba Cloud SDK for Go

Alibaba Cloud SDK for Go supports Go 1.10.x and later. You can run the following command to install the core package:

```
go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

Use the SDKs for Go

To use an SDK for Go, perform the following steps:

1. Initialize a client instance.
2. Create an API request and set the required parameters.
3. Initiate the API request and process the response or exception.

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify **2014-05-26** as the API version and **Ecs** as the service name.
- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify **2021-07-30** as the API version and **ebs** as the service name.

Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Variables:

- <yourRegionID>: Replace the variable with the actual region ID.
- <yourAccessKeyID>: Replace the variable with your AccessKey ID.
- <yourAccessKeySecret>: Replace the variable with your AccessKey secret.
- <Endpoint>: Replace the variable with the endpoint of the required service.
- <http://192.168.XX:33XX>: Replace the variable with the IP address and port of the proxy that you use.

The following sample code provides an example in which the DescribeDisks operation is called to query the information about cloud disks:

```
package main
import (
    "github.com/aliyun/alibaba-cloud-sdk-go/services/ecs"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "fmt"
)
func main() {
    // Create an ecsClient instance.
    ecsClient, err := ecs.NewClientWithAccessKey(
        "<yourRegionID>",           // Replace the variable with the actual region ID.
        "<yourAccessKeyID>",       // Replace the variable with your AccessKey ID.
        "<yourAccessKeySecret>"     // Replace the variable with your AccessKey secret.
    )
    if err != nil {
        // Handle exceptions.
        panic(err)
    }
    // Configure the proxy.
    ecsClient.SetHttpProxy("https://192.168.XX:33XX") // Configure the HTTP proxy.
    ecsClient.GetHttpProxy()                          // Query the HTTP proxy.

    ecsClient.SetHttpsProxy("<https://192.168.XX:33XX>") // Configure the HTTPS proxy.
    ecsClient.GetHttpsProxy()                          // Query the HTTPS proxy.

    ecsClient.SetNoProxy("127.0.0.1,localhost")        // Configure the IP addresses or domain names that do not require proxies.
    ecsClient.GetNoProxy()                            // Query the IP addresses or domain names that do not require proxies.
    // Create an API request and set the parameters.
    request := ecs.DescribeDisksRequest()
    // Specify the endpoint of the service.
    request.Domain = "<Endpoint>"
    // Set request.PageSize to 10.
    request.PageSize = requests.NewInteger(10)
    // Initiate the request and handle exceptions.
    response, err := ecsClient.DescribeInstances(request)
    if err != nil {
        // Handle exceptions.
        panic(err)
    }
    fmt.Println(response)
}
```

The following sample code provides an example in which the DescribeDiskReplicaPairs operation is called to query the information about replication pairs:

```

package main

import (
    "os"
    ebs20210730 "github.com/alibabacloud-go/ebs-20210730/client"
    openapi "github.com/alibabacloud-go/darabonba-openapi/client"
    "github.com/alibabacloud-go/tea/tea"
)

/**
 * Initialize the client by using your AccessKey ID and AccessKey secret.
 * @param accessKeyId
 * @param accessKeySecret
 * @return Client
 * @throws Exception
 */
func CreateClient (accessKeyId *string, accessKeySecret *string) (_result *ebs20210730.Client, _err error) {
    config := &openapi.Config{
        // Your AccessKey ID.
        AccessKeyId: accessKeyId,
        // Your AccessKey secret.
        AccessKeySecret: accessKeySecret,
    }
    // The endpoint of the service.
    config.Endpoint = tea.String("ebs.cn-beijing.aliyuncs.com")
    _result = &ebs20210730.Client{}
    _result, _err = ebs20210730.NewClient(config)
    return _result, _err
}

func _main (args []*string) (_err error) {
    client, _err := CreateClient(tea.String("accessKeyId"), tea.String("accessKeySecret"))
    if _err != nil {
        return _err
    }

    describeDiskReplicaPairsRequest := &ebs20210730.DescribeDiskReplicaPairsRequest{
        RegionId: tea.String("cn-beijing"),
    }
    // Run the following sample code to obtain the response parameter values of the API operation.
    resp, _err = client.DescribeDiskReplicaPairs(describeDiskReplicaPairsRequest)
    if _err != nil {
        return _err
    }
    return _err
}

func main() {
    err := _main(tea.StringSlice(os.Args[1:]))
    if err != nil {
        panic(err)
    }
}

```

4. Optional. Switch to HTTP or HTTPS.

```
request.Scheme = "HTTPS"
```

3.5.2. User Manual

3.5.2.1. Install the core package of SDK for Go

This topic describes how to install the core package of SDK for Go.

Apsara Stack SDK for Go supports Go 1.10.x and later. You can run the following command to install the core package:

```
go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

3.5.2.2. Configure authentication credentials

When you use Alibaba Cloud SDKs to access Apsara Stack services, you must provide an Apsara Stack tenant account for authentication. This topic describes how to configure authentication credentials.

Alibaba Cloud SDK for Go allows you to complete the authentication process by using an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain access credentials assigned by RAM](#).

If you want to use your AccessKey pair to access Apsara Stack, you must configure the AccessKey pair as credentials when you initialize a client instance.

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify **2014-05-26** as the API version and **Ecs** as the service name. In this case, you need to initialize an ECS client instance.

- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify 2021-07-30 as the API version and ebs as the service name. In this case, you need to initialize an EBS client instance.

Note

- Do not disclose any code that contains your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.
- You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Initialize an ECS client instance

```
Initialize a client instance.
ecsClient, err := ecs.NewClientWithAccessKey(
    "<yourRegionID>",           // Replace the variable with the actual region ID.
    "<yourAccessKeyId>",        // Replace the variable with your AccessKey ID.
    "<yourAccessKeySecret>")    // Replace the variable with your AccessKey secret.
```

Initialize an EBS client instance

```
func CreateClient (accessKeyId *string, accessKeySecret *string) (_result *ebs20210730.Client, _err error) {
    config := &openapi.Config{
        // Your AccessKey ID.
        AccessKeyId: accessKeyId,
        // Your AccessKey secret.
        AccessKeySecret: accessKeySecret,
    }
    // The endpoint of the service.
    config.Endpoint = tea.String("ebs.cn-beijing.aliyuncs.com")
    _result = &ebs20210730.Client{}
    _result, _err = ebs20210730.NewClient(config)
    return _result, _err
}
```

3.5.2.3. Initiate a request

This topic describes how to use ECS SDK for Go and EBS SDK for Go to initiate requests.

Use ECS SDK for Go to initiate a request

1. Import ECS SDK for Go.

```
import "github.com/aliyun/alibaba-cloud-sdk-go/services/ecs"
```

2. Create an AcsClient instance that contains all API operations of the service.

```
ecsClient, err := <ecs>.NewClientWithAccessKey( //Replace the variable with the name of the service.
    "<yourRegionID>", // Replace the variable with the actual region ID.
    "<yourAccessKeyId>", // Replace the variable with your AccessKey ID.
    "<yourAccessKeySecret>") // Replace the variable with your AccessKey secret.
ecsClient.Domain = "ecs.aliyuncs.com" // Specify the endpoint of the service.
```

3. Use the `xxx.CreateXXXRequest` method to create a request.

The naming convention for this method is `${service}.Create${apiName}Request`.

- `${service}` specifies the service name in lowercase. Example: ecs. You can replace ecs with the actual service name when you call an SDK of another service.
- `${apiName}` specifies the API operation name. Example: DescribeDisks.

```
request := ecs.CreateDescribeDisksRequest()
request.InstanceIds = ""
```

4. Initiate the request and process the response.

```
response, err := ecsClient.DescribeDisks(request)
```

Use EBS SDK for Go to initiate a request

1. Import EBS SDK for Go.

```
import (
    "os"
    ebs20210730 "github.com/alibabacloud-go/ebs-20210730/client"
    openapi "github.com/alibabacloud-go/darabonba-openapi/client"
    "github.com/alibabacloud-go/tea/tea"
)
```

2. Create a client instance.

```
client, _err := CreateClient(tea.String("accessKeyId"), tea.String("accessKeySecret"))
```

3. Create a request.

```
describeDiskReplicaPairsRequest := &ebs20210730.DescribeDiskReplicaPairsRequest{
    RegionId: tea.String("cn-beijing"),
}
```

4. Initiate the request and process the response.

```
// Run the following sample code to obtain the response parameter values of the API operation.
resp, _err = client.DescribeDiskReplicaPairs(describeDiskReplicaPairsRequest)
if _err != nil {
    return _err
}
```

3.5.2.4. Handle errors

When you use Alibaba Cloud SDK for Go, errors may occur. If an error occurs on the server or in the SDK, a `ServerError` or a `ClientError` is returned. Both error types are included in `sdk.errors`. The errors conform to the standard Go error interface and can be handled in a similar way as standard Go errors.

Troubleshooting

You can use the following code to troubleshoot an error:

```
response, err := ecsClient.DescribeDisks(request)
if err != nil {
    // Obtain the error information.
    err.Error()
}
```

ClientError

If an error occurs in the SDK when you send a request and the error cannot be automatically handled, the SDK returns a `ClientError`.

The following sample code shows how to determine and obtain the error message in a `ClientError` returned by ECS SDK for Go:

Note

If you use a specific error struct, you must add the error package `github.com/aliyun/alibaba-cloud-sdk-go/sdk/errors` by using the following code:

```
package main
import (
    "github.com/aliyun/alibaba-cloud-sdk-go/services/ecs" // Replace ecs with the name of the service.
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    // The error package that you want to add.
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/errors"
    "fmt"
)
```

```
response, err := ecsClient.DescribeDisks(request)
if clientError, ok := err.(*errors.ClientError); ok {
    // Obtain the error code.
    clientError.ErrorCode()
    // Obtain the error message.
    clientError.Message()
    // Obtain the original error. The return value may be nil.
    clientError.OriginError()
}
```

ServerError

If the server returns an error response, the SDK wraps the response into a `ServerError` and returns it to the client. In this case, you can still obtain the original HTTP response from `Response`.

The following sample code shows how to determine and obtain the error message in a `ServerError` and `Response` returned by ECS SDK for Go:

Note

If you use a specific error struct, you must add the error package `github.com/aliyun/alibaba-cloud-sdk-go/sdk/errors` by using the following code:

```
import (
    "github.com/aliyun/alibaba-cloud-sdk-go/services/ecs" // Replace ecs with the name of the service.
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    // The error package that you want to add.
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/errors"
    "fmt"
)
```

```

response, err := ecsClient.DescribeDisks(request)
if serverError, ok := err.(*errors.ServerError); ok {
    // Obtain the error code.
    serverError.ErrorCode()
    // Obtain the error message.
    serverError.Message()
    // Obtain the original HTTP response.
    response.GetOriginHttpResponse()
}

```

3.5.2.5. Use CommonRequest to call API operations

If you want to call the API operations of a cloud service that does not provide SDKs, or the requests and responses of the API operations that you call by using an SDK cannot meet your business requirements, you can use CommonRequest to call API operations. You can use CommonRequest to call all API operations.

Benefits

CommonRequest provides the following benefits:

- **Lightweight:** You only need to download the core package. You do not need to download and install SDKs of Apsara Stack services.
- **Easy to use:** You can call the newly released API operations of a service without the need to update the SDK of the service.
- **Fast in iteration.**

Use CommonRequest to call an API operation

If you want to send a CommonRequest request, you must obtain the values of the following parameters. You can obtain the values of these parameters from ECS API references.

- **Domain:** the endpoint of the service.
- **Version:** the version of the API in the YYYY-MM-DD format.
You can obtain the API version from the common request parameters in the API references of cloud services.
- **Operation information:** the name of the API operation that you want to call.

You must obtain the value of the `Action` parameter and specify the API operation in the `request.setSysAction("<Action>")` format.

For example, if you want to use `CommonRequest` to call the `RunInstances` operation, you must specify the API operation in the `request.setSysAction("RunInstances")` format.

Example: Call an RPC API operation

The API version and service name that you must specify in the code vary depending on the EBS feature that you want to use.

- If you want to use a feature related to cloud disks or snapshots, you must specify **2014-05-26** as the API version and **Ecs** as the service name.
- If you want to use the asynchronous replication or replication pair-consistent group feature, you must specify **2021-07-30** as the API version and **ebs** as the service name.

The following sample code provides an example in which the DescribeDisks operation is called to query the information about cloud disks:

Note

You must replace the variables in the following code with actual values. Delete the angle brackets (<>) before you submit the code.

Variables:

- `<yourRegionID>`: Replace the variable with the actual region ID.
- `<yourAccessKeyID>`: Replace the variable with your AccessKey ID.
- `<yourAccessKeySecret>`: Replace the variable with your AccessKey secret.
- `<ecs.aliyuncs.com>`: Replace the variable with the endpoint of the service.
- `<yourSecurityToken>`: Replace the variable with your security token.
- `<yourResourceGroupID>`: Replace the variable with the resource set ID.
- `<yourOrganizationID>`: Replace the variable with the organization ID.
- `<yourInstanceID>`: Replace the variable with the instance ID.

```

package main
import (
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "fmt"
)

func main() {
    client, err := sdk.NewClientWithAccessKey(
        "<yourRegionID>", // Replace the variable with the actual region ID.
        "<yourAccessKeyID>", // Replace the variable with your AccessKey ID.
        "<yourAccessKeySecret>") // Replace the variable with your AccessKey secret.
    if err != nil {
        panic(err)
    }
    request := requests.NewCommonRequest()
    request.Method = "POST"
    request.Domain = "ecs.aliyuncs.com" // Specify the endpoint of the service.
    request.Product = "Ecs"
    request.Version = "2014-05-26"
    // This is an RPC API operation. Therefore, the ApiName parameter must be set to specify the operation.
    request.ApiName = "DescribeDisks"
    // Set service-related parameters.
    request.QueryParams["PageNumber"] = "1"
    request.QueryParams["PageSize"] = "30"
    /*
    If you use STS AccessKey credentials to call an RPC API operation, you must add the SecurityToken parameter in the request.
    Format:
    request.QueryParams["SecurityToken"]="<yourSecurityToken>"
    Replace <yourSecurityToken> with your security token.
    */
    request.Headers["x-acs-caller-sdk-source"] = "ApiTestDemo" // Required. Specify the source of the API call.
    request.Headers["x-acs-regionid"] = "<yourRegionID>" // Optional. Specify the region ID. This parameter is required if you use Apsara Stack Enterprise V3.15.0 or earlier.
    request.Headers["x-acs-resourcegroupid"] = "<yourResourceGroupID>" // Optional. Specify the resource set ID.
    request.Headers["x-acs-organizationid"] = "<yourOrganizationID>" // Required. Specify the organization ID.
    // request.Headers["x-acs-instanceid"] = "<yourInstanceID>" // Optional. Specify the instance ID. This parameter is required if you want to modify, delete, or view the details of a single instance.
    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Print(response.GetHttpContentString())
}

```

The following sample code provides an example in which the DescribeDiskReplicaPairs operation is called to query the information about replication pairs:

```

package main
import (
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "fmt"
)

func main() {
    client, err := sdk.NewClientWithAccessKey(
        "<yourRegionID>", // Replace the variable with the actual region ID.
        "<yourAccessKeyID>", // Replace the variable with your AccessKey ID.
        "<yourAccessKeySecret>") // Replace the variable with your AccessKey secret.
    if err != nil {
        panic(err)
    }
    request := requests.NewCommonRequest()
    request.Method = "POST"
    request.Domain = "ebs.aliyuncs.com" // Specify the endpoint of the service.
    request.Product = "ebs"
    request.Version = "2021-07-30"
    // This is an RPC API operation. Therefore, the ApiName parameter must be set to specify the operation.
    request.ApiName = "DescribeDiskReplicaPairs"
    // Set service-related parameters.
    request.QueryParams["RegionId"] = "cn-beijing"
    request.QueryParams["PageSize"] = "30"
    /*
    If you use STS AccessKey credentials to call an RPC API operation, you must add the SecurityToken parameter in the request.
    Format:
    request.QueryParams["SecurityToken"]="<yourSecurityToken>"
    Replace <yourSecurityToken> with your security token.
    */
    request.Headers["x-acs-caller-sdk-source"] = "ApiTestDemo" // Required. Specify the source of the API call.
    request.Headers["x-acs-regionid"] = "<yourRegionID>" // Optional. Specify the region ID. This parameter is required if you use Apsara Stack Enterprise V3.15.0 or earlier.
    request.Headers["x-acs-resourcegroupid"] = "<yourResourceGroupID>" // Optional. Specify the resource set ID.
    request.Headers["x-acs-organizationid"] = "<yourOrganizationID>" // Required. Specify the organization ID.
    // request.Headers["x-acs-instanceid"] = "<yourInstanceID>" // Optional. Specify the instance ID. This parameter is required if you want to modify, delete, or view the details of a single instance.
    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Println(response.GetHttpContentString())
}

```

3.6. API references

3.6.1. Make API requests

3.6.1.1. Overview

To call an EBS API operation, you can send an HTTP GET request to the endpoint of the EBS server. You must set the relevant request parameters in the request based on the operation description. Then, the system returns the processing result of the request. All requests and responses are UTF-8 encoded.

3.6.1.2. Request syntax

Endpoint

For information about how to obtain the endpoints of services, see [Obtain an endpoint](#).

Request method

You can call EBS API operations by sending HTTP GET requests. If you use this method, you must include request parameters in request URLs.

Protocols

You can send API requests over HTTP or HTTPS. We recommend that you use HTTPS to ensure higher security.

Request parameters

In each request, you must set the Action parameter to specify the operation that you want to perform. For example, you can set the Action parameter to CreateDisk. You must also specify the common request parameters and operation-specific parameters.

Encoding format

Requests and responses are encoded in UTF-8.

3.6.1.3. API authorization

To ensure the security of your account, we recommend that you use a RAM user to call API operations. Before you call an EBS API operation as a RAM user, you must grant the RAM user the required permissions by creating a policy and attaching the policy to the RAM user.

In RAM, you can authorize a RAM user to call the following API operations on EBS resources:

Category	Operation
Cloud disk	• CreateDisk • DescribeDisks • AttachDisk • DetachDisk • ModifyDiskAttribute • ReInitDisk • ResetDisk • DeleteDisk
Snapshot	• CreateSnapshot • DeleteSnapshot • DescribeSnapshots • CopySnapshot • DescribeSnapshotLinks • DescribeSnapshotsUsage • CreateAutoSnapshotPolicy • DeleteAutoSnapshotPolicy • ModifyAutoSnapshotPolicyEx • ModifySnapshotAttribute • DescribeAutoSnapshotPolicyEX • ApplyAutoSnapshotPolicy • CancelAutoSnapshotPolicy • CreateSnapshotGroup • DescribeSnapshotGroups • DeleteSnapshotGroup • ModifySnapshotGroup
Asynchronous replication	• CreateDiskReplicaPair • StartDiskReplicaPair • DescribeDiskReplicaPairs • DescribeDiskReplicaPairProgress • ModifyDiskReplicaPair • FailoverDiskReplicaPair • ReprotectDiskReplicaPair • StopDiskReplicaPair • DeleteDiskReplicaPair
Replication pair-consistent group	• CreateDiskReplicaGroup • AddDiskReplicaPair • StartDiskReplicaGroup • DescribeDiskReplicaGroups • ModifyDiskReplicaGroup • FailoverDiskReplicaGroup • ReprotectDiskReplicaGroup • RemoveDiskReplicaPair • StopDiskReplicaGroup • DeleteDiskReplicaGroup

3.6.1.4. Request signatures

EBS verifies the identity of the sender of each access request. Therefore, all API requests must be signed regardless of whether the requests are sent over HTTP or HTTPS.

EBS implements symmetric encryption by using an AccessKey pair to verify the identity of the request sender. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. They are the identity credentials issued by Alibaba Cloud. The AccessKey ID is used to verify the identity of the request sender, and the AccessKey secret is used to encrypt and verify the signature string. You must keep your AccessKey secret strictly confidential. For more information about how to obtain an AccessKey pair, see [Obtain access credentials assigned by RAM](#).

When you call an EBS API operation, you must perform the following steps to sign the request:

1. Create a canonicalized query string by arranging the request parameters.
 - i. Sort the request parameters. Sort the request parameters in alphabetical order. All common request parameters except Signature and all operation-specific request parameters are included.

 **Note** If you use the GET method to send the request, these parameters are included in the request URI and connected by ampersands (&). The first request parameter follows the question mark (?).

- ii. Encode the request parameters. Encode the name and value of each request parameter. The names and values must be URL encoded by using the UTF-8 character set based on the following rules:
 - Uppercase letters, lowercase letters, digits, and specific special characters do not need to be encoded. The specific special characters are hyphens (-), underscores (_), periods (.), and tildes (~).
 - Other characters must be percent encoded in the %XY format. XY represents the ASCII code of the characters in hexadecimal notation. For example, double quotation marks (") are encoded as %22.
 - Extended UTF-8 characters must be encoded in the %XY%ZA... format.
 - Spaces must be encoded as %20 instead of plus signs (+).

This encoding method is similar to but slightly different from the `application/x-www-form-urlencoded` Multipurpose Internet Mail Extension (MIME) encoding algorithm, such as `java.net.URLEncoder` in the Java standard library. You can encode the parameter names and values by using the encoding method in the Java standard library. Then, in the encoded string, replace the plus signs (+) with %20, asterisks (*) with %2A, and tildes (~) with %7E to obtain an encoded string that complies with the preceding encoding rules. You can use the following `percentEncode()` method to perform the encoding:

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedOperationException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A").replace("%7E", "~") : null;
}
```

- iii. Use equal signs (=) to connect the encoded parameter names and values.
 - iv. Sort the parameters in the order specified in Step 1.i and connect the parameter name-parameter value pairs by using ampersands (&) to generate a canonicalized query string.
2. Construct a string-to-sign based on the canonicalized query string and the following rules:

```
StringToSign=
    HTTPMethod + "&" +
    percentEncode("/") + "&" +
    percentEncode(CanonicalizedQueryString)
```

- HTTPMethod specifies the HTTP method that you want to use to send the request, such as GET.
 - `percentEncode("/")` specifies the encoded value for the forward slash (/) based on the URL encoding rules described in Step 1.ii. The value is %2F.
 - `percentEncode(CanonicalizedQueryString)` specifies the canonicalized query string generated in Step 1. The encoding follows the URL encoding rules described in Step 1.ii.
3. Calculate the Hash-based Message Authentication Code (HMAC) value of the string-to-sign as defined in [RFC 2104](#).

Note Use the SHA1 algorithm to calculate the HMAC value of the string-to-sign. The AccessKey secret to which an ampersand (&) (ASCII:38) is appended is used as the key for HMAC calculation.

4. Encode the HMAC value in Base64 to obtain the signature string.
5. Add the signature string to the request as the Signature parameter. This way, the API request is signed.

Note When the signature string is submitted to ECS as the final request parameter value, the value must be URL-encoded like other parameters based on rules defined in [RFC 3986](#).

3.6.1.5. Responses

API responses use the HTTP response format. A 2xx status code indicates a successful call, and a 4xx or 5xx status code indicates a failed call. Success responses can be returned in the JSON or XML format. You can specify the response format in the request. By default, responses are returned in the XML format.

Note The sample responses provided in this topic have been formatted with line breaks and indentations for better readability. The actual responses are not properly formatted as these examples.

Sample success responses

• XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Result root node-->
<Operation name + Response>
  <!--Return request tag-->
  <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
  <!--Return result data-->
</Operation name + Response>
```

• JSON format

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
  /* Return result data */
}
```

Sample error responses

If an error occurs when you call an API operation, no result data is returned. You can troubleshoot the error based on the error code returned. For an error that occurs when you call an API operation, a 4XX or 5XX HTTP status code is returned. The response body contains the specific error code and error message. The response body also contains the request ID and ID of the host that you wanted to access with this request. The request ID is globally unique.

• XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <HostId>ebs.example.com</HostId>
  <Code>InvalidOwnerId</Code>
  <Message>The specified OwnerId is not valid.</Message>
</Error>
```

- JSON format

```
{
  "RequestId": "8906582E-6722-409A-A6C4-0E7863B733A5",
  "HostId": "ebs.example.com",
  "Code": "InvalidOwnerId",
  "Message": "The specified OwnerId is not valid."
}
```

3.6.2. RAM authentication

When you call API operations to access EBS resources within an Apsara Stack tenant account as a RAM user, EBS checks whether the RAM user is granted the required permissions.

Background information

You can use your Apsara Stack tenant account or a RAM user account to manage your EBS resources by using the EBS console or by calling API operations. Specific permissions are required in the following scenarios:

- A RAM user is newly created and does not have permissions to manage resources within the Apsara Stack tenant account.
- You want to access EBS resources from other Apsara Stack services, or access other Apsara Stack services from EBS.
- Before you can manage a resource that is configured with access control, you must be granted the required permissions on the resource and the relevant API operations by the resource owner.

When an Apsara Stack tenant account requests access to EBS resources within your Apsara Stack tenant account by calling EBS API operations, EBS instructs RAM to perform a permission check to ensure that the requester account is granted the required permissions. Required permissions vary based on the requested EBS resources and API operations.

Rule details

The permissions that EBS checks vary based on the resources that are requested by the API operation and the syntax of the API operation. The following table describes the authentication rules for each API operation.

API operation	Authentication rule	Description
CreateDisk	- acs:ecs:\$regionid:\$accountid:disk/* - acs:ecs:\$regionid:\$accountid:snapshot/\$snapshotid  Note If you specify the Snapshotid parameter in the API request, you must specify the snapshotid variable.	Creates a disk.
DescribeDisks	acs:ecs:\$regionid:\$accountid:disk/*	Queries one or more cloud disks and local disks that you have created.
AttachDisk	- acs:ecs:\$regionid:\$accountid:disk/\$diskid - acs:ecs:\$regionid:\$accountid:instance/\$instanceid	Attaches a disk to an ECS instance.
DetachDisk	- acs:ecs:\$regionid:\$accountid:disk/\$diskid - acs:ecs:\$regionid:\$accountid:instance/\$instanceid	Detaches a disk from an ECS instance.
ModifyDiskAttribute	acs:ecs:\$regionid:\$accountid:disk/\$diskid	Modifies the attributes of a cloud disk.
ReInitDisk	acs:ecs:\$regionid:\$accountid:disk/\$diskid	Re-initializes a cloud disk to the state that it was in when it was created.
ResetDisk	- acs:ecs:\$regionid:\$accountid:snapshot/\$snapshotid - acs:ecs:\$regionid:\$accountid:disk/\$diskid	Rolls back a cloud disk to a specific point in time by using a snapshot of the cloud disk.
ReplaceSystemDisk	- acs:ecs:\$regionid:\$accountid:instance/\$instanceid - acs:ecs:\$regionid:\$accountid:image/\$imageNo	Replaces the system disk or the operating system of an ECS instance.
ResizeDisk	acs:ecs:\$regionid:\$accountid:disk/\$diskid	Resizes a cloud disk. You can resize a system disk or a data disk.
DeleteDisk	acs:ecs:\$regionid:\$accountid:disk/\$diskid	Releases a disk.
CreateSnapshot	- acs:ecs:\$regionid:\$accountid:disk/\$diskid - acs:ecs:\$regionid:\$accountid:snapshot/*	Creates a snapshot for a cloud disk.
DescribeSnapshots	- acs:ecs:\$regionid:\$accountid:snapshot/\$snapshotid - acs:ecs:\$regionid:\$accountid:snapshot/*	Queries all the snapshots of an ECS instance or a cloud disk.

API operation	Authentication rule	Description
DeleteSnapshot	<code>acs:ecs:\$regionid:\$accountid:snapshot/\$snapshotid</code>	Deletes a specific snapshot.
DescribeSnapshotLinks	<code>acs:ecs:\$regionid:\$accountid:disk/\$diskid</code> <code>acs:ecs:\$regionid:\$accountid:disk/*</code>	Queries the snapshot chains of one or more cloud disks.
CreateAutoSnapshotPolicy	<code>acs:ecs:*:\$accountid:snapshot/*</code>	Creates an automatic snapshot policy.
DeleteAutoSnapshotPolicy	<code>acs:ecs:*:\$accountid:snapshot/*</code>	Deletes an automatic snapshot policy.
ModifySnapshotAttribute	<code>acs:ecs:\$regionid:\$accountid:snapshot/\$snapshotid</code>	Modifies the name or description of a snapshot.
ApplyAutoSnapshotPolicy	<code>acs:ecs:*:\$accountid:snapshot/*</code>	Applies an automatic snapshot policy to one or more cloud disks or replaces the existing automatic snapshot policy of the cloud disks.
CancelAutoSnapshotPolicy	<code>acs:ecs:*:\$accountid:snapshot/*</code>	Disables an automatic snapshot policy for one or more cloud disks.
CreateDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/*</code>	Creates a replication pair.
StartDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Enables the asynchronous replication feature for a replication pair.
DescribeDiskReplicaPairs	<code>acs:ebs:\$regionid:\$accountid:pair/*</code>	Queries the information of one or more replication pairs.
DescribeDiskReplicaPairProgress	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Queries the replication progress of a replication pair.
ModifyDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Modifies the information of a replication pair.
FailoverDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Enables the failover feature for a replication pair.
ReprotectDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Enables the reverse replication feature for a replication pair.
StopDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Disables the asynchronous replication feature for a replication pair.
DeleteDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pair/\$pairid</code>	Deletes a replication pair.
CreateDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/*</code>	Creates a replication pair-consistent group.
AddDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Adds a replication pair to a replication pair-consistent group.
StartDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Enables the asynchronous replication feature for a replication pair-consistent group.
DescribeDiskReplicaGroups	<code>acs:ebs:\$regionid:\$accountid:pg/*</code>	Queries the information of one or more replication pair-consistent groups.
ModifyDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Modifies the information of a replication pair-consistent group.
FailoverDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Enables the failover feature for a replication pair-consistent group.
ReprotectDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Enables the reverse replication feature for a replication pair-consistent group.
RemoveDiskReplicaPair	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Removes a replication pair from a replication pair-consistent group.
StopDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Disables the asynchronous replication feature for a replication pair-consistent group.
DeleteDiskReplicaGroup	<code>acs:ebs:\$regionid:\$accountid:pg/\$groupid</code>	Deletes a replication pair-consistent group.

3.6.3. Common parameters

Common parameters include common request parameters, common headers, and common response parameters.

Common request parameters

Common request parameters must be included in all API requests.

Common request parameters

Parameter	Type	Required	Description
Format	String	No	The format of the response. Set the value to JSON.
Version	String	Yes	The version number of the API. Specify the version number in the YYYY-MM-DD format. The version of the API varies based on the feature you want to use. - If you want to use a feature related to cloud disks or snapshots, specify 2014-05-26 as the API version. - If you want to use the asynchronous replication or replication pair-consistent group feature, specify 2021-07-30 as the API version.
AccessKeyId	String	Yes	The AccessKey ID provided to you by Apsara Stack.
AccessKeySecret	String	Yes	The AccessKey secret provided to you by Apsara Stack.
RegionId	String	Yes	The region ID provided to you by Apsara Stack.
Product	String	Yes	The name of the service to which the API operation belongs. The version of the API varies based on the feature you want to use. - If you want to use a feature related to cloud disks or snapshots, specify ecs as the service name. - If you want to use the asynchronous replication or replication pair-consistent feature, specify ebs as the service name.

To obtain the values of the `Product` and `Version` parameters in the Apsara Uni-manager Management Console, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Others > API Tool**.
3. In the left-side navigation pane, click **API Directory**.
4. In the **Search by operation name** search box, enter the name of an operation to query its information.

Common headers

Header	Description
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information, see Obtain an organization ID .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information, see Obtain a resource set ID .
x-acs-instanceid	The ID of the instance on which you want to perform the operation. For more information, see Obtain an instance ID .

Note When you call the API operations of cloud services, you must specify the x-acs-instanceid, x-acs-organizationid, and x-acs-resourcegroupid headers. These headers specify the ID of the instance on which you want to perform the operation, and the organization and resource set to which the instance belongs. If you call an API operation to create an instance or query the information about one or more instances, you do not need to specify the x-acs-instanceid header. If you do not specify the headers when you call an API operation, the information of the root organization and root resource set is used by default.

Common response parameters

API responses use the HTTP response format:

- A 2xx status code indicates a successful call.
- A 4xx or 5xx status code indicates a failed call.

Responses are returned in the JSON format. Every response returns a unique request ID regardless of whether the call is successful.

3.6.4. Disk

3.6.4.1. CreateDisk

Creates a data disk. The disk can be an ultra disk, a shared ultra disk, a standard SSD, or a shared standard SSD.

Description

- By default, the DeleteAutoSnapshot parameter is set to true when a disk is created. This indicates that when the disk is released, its automatic snapshots are deleted. You can call the ModifyDiskAttribute operation to modify this parameter.
- You must specify one of the Size and SnapshotId parameters. The Size parameter specifies the size of the disk and the SnapshotId parameter specifies the snapshot that is used to create the disk.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDisk	The operation that you want to perform. If you use a custom HTTP URL or HTTPS URL to make an API request, you must specify the <code>Action</code> parameter. Set the value to CreateDisk.
RegionId	String	Yes	cn-qingdao-env17-d01	The ID of the region in which to create the disk. You can call the DescribeRegions operation to query the most recent region list.
ZoneId	String	No	cn-qingdao-env17-amtest11001-a	The ID of the zone in which to create the disk. This parameter is required.
SnapshotId	String	No	s-bp67acfmazb4p****	The ID of the snapshot used to create the disk. If this parameter is specified, the Size parameter is ignored, and the disk is created with the size of the specified snapshot. Snapshots that were created on or before July 15, 2013 cannot be used to create disks.
DiskName	String	No	test-DiskName	The name of the disk. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, colons (:), underscores (_), and hyphens (-). This parameter is empty by default.
Size	Integer	No	2000	The size of the disk. Unit: GB. The value of this parameter must be greater than or equal to the size of the specified snapshot. Valid values: - Valid values when DiskCategory is set to cloud_efficiency: 20 to 32768 - Valid values when DiskCategory is set to san_efficiency: 20 to 32768 - Valid values when DiskCategory is set to cloud_ssd: 20 to 32768 - Valid values when DiskCategory is set to san_ssd: 20 to 32768
DiskCategory	String	No	cloud_ssd	The category of the disk. Valid values: - cloud_efficiency: ultra disk - san_efficiency: shared ultra disk - cloud_ssd: standard SSD - san_ssd: shared standard SSD Default value: cloud_efficiency.
Description	String	No	This is a description example.	The description of the disk. The description must be 2 to 256 characters in length and cannot start with http:// or https://. This parameter is empty by default.
Encrypted	Boolean	No	false	Specifies whether to encrypt the disk. Default value: false.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate a value, but you must make sure that it is unique among different requests. The ClientToken value can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence in ECS Developer Guide.
InstanceId	String	No	i-bp1g6zv0ce8ogp12****	 Note This parameter is unavailable.
Tag.N.value	String	No	TestValue	The tag value of the disk.
Tag.N.key	String	No	TestKey	The tag key of the disk.
Tag.N.Key	String	No	TestKey	The key of tag N of the disk. Valid values of N: 1 to 20. The tag key cannot be an empty string. It can be up to 128 characters in length and cannot start with acs: or aliyun. It cannot contain http:// or https://.
Tag.N.Value	String	No	TestValue	The value of tag N of the disk. Valid values of N: 1 to 20. The tag value can be an empty string. It can be up to 128 characters in length and cannot start with acs: or contain http:// or https://.

Parameter	Type	Required	Example	Description
ResourceGroupId	String	No	rg-bp67acfmxazb4p****	The ID of the resource group to which the disk belongs.
KMSKeyId	String	No	0e478b7a-4262-4802-b8cb-00d3fb40826X	The ID of the Key Management Service (KMS) key used by the disk.
PerformanceLevel	String	No	PL1	 Note This parameter is unavailable.
StorageSetId	String	No	ss-bp67acfmxazb4p****	The ID of the storage set.
EncryptAlgorithm	String	No	aes-256	The encryption algorithm. Valid values: - aes-256: AES-256 algorithm - sm4-128: SM4 encryption algorithm
StorageSetPartitionNumber	Integer	No	3	The number of partitions in the storage set.

Response parameters

Parameter	Type	Example	Description
DiskId	String	d-bp131n0q38u3a4zi1****	The ID of the disk.
OrderId	String	20413515388****	 Note This parameter is unavailable.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=CreateDisk
&RegionId=cn-qingdao-env17-d01
&ZoneId=cn-qingdao-env17-amtest11001-a
&SnapshotId=s-bp67acfmxazb4p****
&DiskName=test-DiskName
&Size=2000
&DiskCategory=cloud_ssd
&Description=This is a description example.
&Encrypted=false
&ClientToken=123e4567-e89b-12d3-a456-426655440000
&Tag.1.Key=TestKey
&Tag.1.Value=TestValue
&<Common request parameters>
```

Sample success responses

XML **format**

```
<CreateDiskResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E</RequestId>
  <DiskId>d-bp131n0q38u3a4zi1****</DiskId>
</CreateDiskResponse>
```

JSON **format**

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E",
  "DiskId": "d-bp131n0q38u3a4zi1****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidSize.ValueNotSupported	The specified parameter Size is not valid.	The error message returned because the specified Size parameter is invalid.
403	InvalidDataDiskCategory.NotSupported	Specified disk category is not supported.	The error message returned because the specified DiskCategory parameter is not supported.
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified RegionId parameter does not exist.

HTTP status code	Error code	Error message	Description
404	InvalidZoneId.NotFound	The specified zone does not exist.	The error message returned because the specified ZoneId parameter does not exist.
404	InvalidSnapshotId.NotFound	The specified SnapshotId does not exist.	The error message returned because the specified snapshot does not exist. Check whether the snapshot ID is correct.
400	InvalidDiskName.Malformed	The specified disk name is wrongly formed.	The error message returned because the specified DiskName parameter is invalid. The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). It must start with a letter and cannot start with http:// or https://.
400	InvalidDescription.Malformed	The specified description is wrongly formed.	The error message returned because the specified Description parameter is invalid. The description must be 2 to 256 characters in length and cannot start with http:// or https://.
403	InstanceDiskCategoryLimitExceed	The total size of specified disk category in an instance exceeds.	The error message returned because the total size of disks of the specified category exceeds the maximum capacity allowed for an instance.
403	InvalidSnapshot.NotReady	The specified snapshot creation is not completed yet.	The error message returned because the specified snapshot is not created.
403	InvalidSnapshot.TooOld	This operation is forbidden because the specified snapshot is created before 2013-07-15.	The error message returned because the specified snapshot was created on or before July 15, 2013 and cannot be used to create disks.
403	InvalidSnapshot.TooLarge	The capacity of snapshot exceeds 2000GB.	The error message returned because the size of the specified snapshot exceeds 2,000 GB.
403	OperationDenied	The specified snapshot is not allowed to create disk.	The error message returned because the specified snapshot cannot be used to create disks.
403	QuotaExceed.PortableCloudDisk	The quota of portable cloud disk exceeds.	The error message returned because the maximum number of removable disks has been reached.
400	MissingParameter	The input parameter either "SnapshotId" or "Size" should be specified.	The error message returned because the SnapshotId and Size parameters are not specified. You must specify at least one of them.
403	InvalidDiskCategory.ValueUnauthorized	The disk category is not authorized.	The error message returned because you are not authorized to manage the specified disk category.
403	InvalidSnapshotId.NotReady	The specified snapshot has not completed yet.	The error message returned because the specified snapshot is being created.
403	InvalidSnapshotId.NotDataDiskSnapshot	The specified snapshot is system disk snapshot.	The error message returned because the specified snapshot is a system disk snapshot.
403	InvalidDiskSize.TooSmall	Specified disk size is less than the size of snapshot.	The error message returned because the size of the specified disk is smaller than that of the snapshot.
403	OperationDenied	The type of the disk does not support the operation.	The error message returned because the specified disk category does not support this operation.
403	InvalidDataDiskCategory.NotSupported	%s	The error message returned because the specified DiskCategory parameter is invalid.
403	InvalidDiskSize.NotSupported	disk size is not supported.	The error message returned because the specified Size parameter is invalid.
400	InvalidDiskCategory.NotSupported	The specified disk category is not support.	The error message returned because the specified DiskCategory parameter is not supported.
400	Account.Arrearage	Your account has an outstanding payment.	The error message returned because you have overdue payments within your account.
400	InvalidDiskCategory.ValueNotSupported	The specified parameter "DiskCategory" is not valid.	The error message returned because the specified SystemDisk.Category parameter is invalid.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient. Add funds to your account and try again.
403	InvalidAccountStatus.SnapshotServiceUnavailable	Snapshot service has not been opened yet.	The error message returned because the operation is not supported while the snapshot service is not activated.
400	InvalidDataDiskCategory.ValueNotSupported	%s	The error message returned because the specified DiskCategory parameter is invalid.

HTTP status code	Error code	Error message	Description
400	InvalidParameter.Conflict	%s	The error message returned because a specified parameter is invalid. Check whether parameter conflicts exist.
400	RegionUnauthorized	%s	The error message returned because you are not authorized to create disks in the specified region.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
400	Zone.NotOnSale	%s	The error message returned because the requested resources are unavailable in the specified zone.
400	InvalidDataDiskSize.ValueNotSupported	%s	The error message returned because the specified Size parameter is invalid.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the specified payment method is invalid.
400	OperationDenied	The specified Zone is not available or not authorized.	The error message returned because the specified zone is unavailable or you are not authorized to access it.
404	InvalidResourceGroup.NotFound	The ResourceGroup provided does not exist in our records.	The error message returned because the specified ResourceGroup parameter does not exist.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk category does not support the operation.
403	InvalidDiskSize.NotSupported	The specified disk size is not supported.	The error message returned because the specified disk size is not supported.
400	InvalidDiskSize.NotSupported	The specified parameter size is not valid.	The error message returned because the specified Size parameter is invalid.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are authorized to manage the disk. Try again when you are authorized.
400	InvalidDiskSizeOrCategory	The specified disk category or size is invalid.	The error message returned because the specified DiskCategory or Size parameter is invalid.
400	InvalidParameter.EncryptedIllegal	%s	The error message returned because the specified parameter is invalid. Check whether your encryption operation is valid.
400	InvalidParameter.EncryptedNotSupported	%s	The error message returned because the specified parameter is invalid and your encryption operation is not supported.
400	EncryptedOption.Conflict	%s	The error message returned because the parameter used to encrypt disks is not supported.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
403	QuotaExceed.PostPaidDisk	Living postPaid disks quota exceeded.	The error message returned because the maximum number of pay-as-you-go disks has been reached.
403	InvalidRegion.NotSupport	The specified region does not support byok.	The error message returned because the bring your own key (BYOK) feature is not supported in the region.
403	UserNotInTheWhiteList	The user is not in byok white list.	The error message returned because you are not in the BYOK whitelist. Try again after you are in the whitelist.
400	InvalidParameter.EncryptedIllegal	The specified parameter Encrypted must be true when kmsKeyId is not empty.	The error message returned because the encryption feature is not enabled after a KMS key ID is specified.
403	InvalidParameter.KMSKeyId.CMKNotEnabled	The CMK needs to be enabled.	The error message returned because the customer master key (CMK) is not enabled when a KMS key ID is specified for a disk. You can call the DescribeKey operation of KMS to query the information of the specified CMK.
404	InvalidParameter.KMSKeyId.NotFound	The specified KMSKeyId does not exist.	The error message returned because the specified KMSKeyId parameter does not exist.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service have no right to access your KMS.	The error message returned because ECS is not authorized to access your KMS resources.
403	SecurityRisk.3DVerification	We have detected a security risk with your default credit or debit card. Please proceed with verification via the link in your email.	The error message returned because risks are detected in your default credit card or debit card. Click the URL in the email for verification.

HTTP status code	Error code	Error message	Description
400	Duplicate.TagKey	The Tag.N.Key contain duplicate key.	The error message returned because the specified tag key already exists. Tag keys must be unique.
400	InvalidTagKey.Malformed	The specified Tag.n.Key is not valid.	The error message returned because the specified Tag.N.Key parameter is invalid.
400	InvalidTagValue.Malformed	The specified Tag.n.Value is not valid.	The error message returned because the specified Tag.N.Value parameter is invalid.
404	InvalidInstanceId.NotFound	The InstanceId provided does not exist in our records.	The error message returned because the specified InstanceId parameter does not exist. Check whether the instance ID is correct.
403	InvalidStatus.Updating	The instance is upgrading; please try again later.	The error message returned because the configurations of the specified instance are being upgraded. Try again later.
400	InvalidPerformanceLevel.Malformed	The specified parameter PerformanceLevel is not valid.	The error message returned because the specified PerformanceLevel parameter is invalid.
403	QuotaExceed.Tags	%s	The error message returned because the maximum number of tags has been reached.
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified InstanceId parameter does not exist. Check whether the instance ID is correct.
403	OperationDenied.SnapshotNotAllowed	The specified snapshot is not allowed to create disk.	The error message returned because the specified snapshot cannot be used to create disks.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the last token request is being processed. Try again later.

3.6.4.2. DescribeDisks

You can call this operation to query one or more Elastic Block Storage (EBS) devices that you have created, including cloud disks and local disks.

Description

- You can specify multiple request parameters such as `RegionId`, `ZoneId`, `DiskIds`, and `InstanceId` to be queried. Specified parameters have logical AND relations. Only the specified parameters are included in the filter conditions.
- The `DiskIds` value is a JSON array. If `DiskIds` is not specified, it is not used as a filter condition. If `DiskIds` is set to an empty JSON array, it is regarded as a valid filter condition and an empty result is returned.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDisks	The operation that you want to perform. Set the value to DescribeDisks.
RegionId	String	Yes	cn-qingdao-env17-d01	The region ID of the disk. You can call the DescribeRegions operation to query the most recent region list.
ZoneId	String	No	cn-qingdao-env17-amtest11001-a	The zone ID of the disk.
DiskIds	String	No	["d-bp67acfmazb4p****", "d-bp67acfmazb4p****", ... "d-bp67acfmazb4p****"]	The IDs of disks. The value is a JSON array that consists of up to 100 disk IDs. Separate multiple disk IDs with commas (,).
InstanceId	String	No	i-bp67acfmazb4p****	The ID of the instance to which the disk is attached.
DiskType	String	No	all	The type of the disk. Valid values: - all: system disk and data disk - system: system disk - data: data disk Default value: all.

Parameter	Type	Required	Example	Description
Category	String	No	all	The category of the disk. Valid values: - all: all disk categories - cloud: basic disk - cloud_efficiency: ultra disk - cloud_ssd: standard SSD - cloud_essd: enhanced SSD (ESSD) - local_ssd_pro: I/O-intensive local disk - local_hdd_pro: throughput-intensive local disk - ephemeral: retired local disk - ephemeral_ssd: retired local SSD Default value: all.
Status	String	No	All	The status of the disk. Valid values: - In_use - Available - Attaching - Detaching - Creating - Relining - All Default value: All.
SnapshotId	String	No	s-bp67acfmazb4p****	The ID of the snapshot used to create the disk.
Portable	Boolean	No	false	Specifies whether the disk is removable. Valid values: - true: The disk is removable. A removable disk can exist independently and can be attached to or detached from an instance within the same zone. - false: The disk is not removable. A disk that is not removable cannot exist independently or be detached from one instance and then attached to another instance within the same zone. To attach or detach a disk, make sure that the disk is removable. When basic disks, ultra disks, and standard SSDs are used as system disks, they are not removable. This attribute cannot be modified.
DeleteWithInstance	Boolean	No	false	Specifies whether the disk is released when its associated instance is released. Valid values: - true: The disk is released when its associated instance is released. - false: The disk is not released but is retained when its attached instance is released. Default value: false.
DeleteAutoSnapshot	Boolean	No	false	Specifies whether the automatic snapshots of the disk are deleted when the disk is released. Default value: false.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
NextToken	String	No	AAAAAdWBF2****	The query token. Set the value to the <code>NextToken</code> value returned in the last call to the <code>DescribeDisks</code> operation.
MaxResults	Integer	No	50	The maximum number of entries to return on each page. Valid values: 1 to 500. Default value: 10.
DiskName	String	No	test-DiskName	The name of the disk.

Parameter	Type	Required	Example	Description
AutoSnapshotPolicyId	String	No	sp-m5e2w2jutw8bv31a****	The ID of the automatic snapshot policy applied to the disk.
EnableAutomatedSnapshotPolicy	Boolean	No	false	Specifies whether to query disks that have automatic snapshot policies applied. Default value: false.
Filter.1.Key	String	No	CreationStartTime	The key of filter 1 used to query resources. Set the value to CreationStartTime.
Filter.2.Key	String	No	CreationEndTime	The key of filter 2 used to query resources. Set the value to CreationEndTime.
Filter.1.Value	String	No	2017-12-05T22:40:00Z	The value of filter 1 used to query resources. Set the value to the beginning of the time range to query.
Filter.2.Value	String	No	2017-12-06T22:40:00Z	The value of filter 2 used to query resources. Set the value to the end of the time range to query.
Tag.N.Key	String	No	TestKey	The key of tag N of the disk. Valid values of N: 1 to 20.
Tag.N.Value	String	No	TestValue	The value of tag N of the disk. Valid values of N: 1 to 20.
ResourceGroupId	String	No	rg-bp67acfmxazb4p****	The ID of the resource group to which the disk belongs.
EnableShared	Boolean	No	false	Specifies whether the disk is a Shared Block Storage device.
Encrypted	Boolean	No	false	Specifies whether to query only encrypted disks. Default value: false.
AdditionalAttributes.N	RepeatList	No	null	Attribute N of the disk.  Note This parameter will be removed in the future. We recommend that you use other parameters to ensure future compatibility.
DryRun	Boolean	No	false	Specifies whether to check the validity of the request without actually making the request. Valid values: - true: The validity of the request is checked but the request is not made. The system checks whether your AccessKey pair is valid, whether RAM users are authorized, and whether the required parameters are specified. If the check fails, the corresponding error is returned. If the check succeeds, the <code>DryRunOperation</code> error code is returned. - false: The validity of the request is checked. If the check succeeds, a 2XX HTTP status code is returned and the request is made. Default value: false.
KMSKeyId	String	No	0e478b7a-4262-4802-b8cb-00d3fb408***	The ID of the KMS key used by the disk.

Response parameters

Parameter	Type	Example	Description
Disks	Array of Disk		Details about the disks.
Disk			
AttachedTime	String	2018-01-01T01:04:22Z	The time when the disk was attached. The time follows the ISO 8601 standard in the yyyy-MM-ddThh:mmZ format. The time is displayed in UTC. This parameter is valid only when the value of <code>Status</code> is <code>Available</code> .
AutoSnapshotPolicyId	String	sp-bp67acfmxazb4p****	The ID of the automatic snapshot policy applied to the disk.

Parameter	Type	Example	Description
Bdfid	String	test	 Note This parameter is not supported.
Category	String	cloud_ssd	The category of the disk. Valid values: - cloud: basic disk - cloud_efficiency: ultra disk - cloud_ssd: standard SSD - cloud_essd: ESSD - local_ssd_pro: I/O-intensive local disk - local_hdd_pro: throughput-intensive local disk - ephemeral: retired local disk - ephemeral_ssd: retired local SSD
CreationTime	String	2018-01-01T01:01:22Z	The time when the disk was created.
DeleteAutoSnapshot	Boolean	false	Indicates whether the automatic snapshots of the disk are deleted when the disk is released. Valid values: - true: The automatic snapshots of the disk are deleted when the disk is released. - false: The automatic snapshots of the disk are retained when it is released. The snapshots created by calling the CreateSnapshot operation or by using the ECS console are not affected by this parameter.
DeleteWithInstance	Boolean	true	Indicates whether the disk is released when its associated instance is released. Valid values: - true: The disk is released when its associated instance is released. - false: The disk is retained when its associated instance is released.
Description	String	This is a description example.	The description of the disk.
DetachedTime	String	2018-01-08T01:01:22Z	The time when the disk was detached. This parameter is valid only when the value of <code>Status</code> is <code>Available</code>.
Device	String	/dev/xvdb	The device name of the disk on its associated instance. Example: <code>/dev/xvdb</code>. This parameter has a value only when the value of <code>Status</code> is <code>In use</code>.  Note This parameter will be removed in the future. We recommend that you use other parameters to ensure future compatibility.
DiskChargeType	String	PostPaid	 Note This parameter is not supported.
DiskId	String	d-bp18um4r4f2fve24****	The ID of the disk.
DiskName	String	test-DiskName	The name of the disk.
EnableAutoSnapshot	Boolean	false	 Note This parameter is not supported.
EnableAutomatedSnapshotPolicy	Boolean	false	Indicates whether the disk has an automatic snapshot policy applied.
Encrypted	Boolean	false	Indicates whether the disk was encrypted.
ExpiredTime	String	2018-01-10T01:01:22Z	 Note This parameter is not supported.

Parameter	Type	Example	Description
IOPS	Integer	4000	The number of input/output operations per second (IOPS).
IOPSRead	Integer	2000	The number of I/O reads per second.
IOPSWrite	Integer	2000	The number of I/O writes per second.
ImageId	String	m-bp13aqm171qynt3****	The ID of the image used to create the instance. This parameter is empty unless the disk was created from an image. The value of this parameter remains unchanged throughout the lifecycle of the disk.
InstanceId	String	i-bp1j4izjdf3owhe****	The ID of the instance to which the disk is attached. This parameter has a value only when the value of <code>Status</code> is <code>In_use</code> .
KMSKeyId	String	0e478b7a-4262-4802-b8cb-00d3fb40****	The ID of the KMS key used by the disk.
MountInstanceNum	Integer	1	The number of instances to which the Shared Block Storage device is attached.
MountInstances	Array of MountInstance		The attaching information of the disk.
MountInstance			
AttachedTime	String	2017-12-05T2340:00Z	The time when the disk was attached. The time follows the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time is displayed in UTC.
Device	String	/dev/xvda	The mount point of the disk.
InstanceId	String	i-bp1j4izjdf3owhe****	The ID of the instance to which the disk is attached.
OperationLocks	Array of OperationLock		The reasons why the disk was locked.
OperationLock			
LockReason	String	security	The security reason why the disk was locked.
PerformanceLevel	String	PL2	The performance level of the ESSD. Valid values: - PL1: A single enhanced SSD can deliver up to 50,000 random read/write IOPS. - PL2: A single enhanced SSD can deliver up to 100,000 random read/write IOPS. - PL3: A single ESSD can deliver up to 1,000,000 random read/write IOPS.
Portable	Boolean	false	Indicates whether the disk is removable.
ProductCode	String	test	 Note This parameter is not supported.
RegionId	String	cn-qingdao-env17-d01	The region ID of the disk.
ResourceGroupId	String	rg-bp67acfmazb4p****	The ID of the resource group to which the disk belongs.
SerialNumber	String	bp18um4r4f2fve2****	The serial number of the disk.
Size	Integer	2000	The size of the disk. Unit: GiB.
SourceSnapshotId	String	s-bp67acfmazb4p****	The ID of the snapshot used to create the disk. This parameter is empty unless the disk was created from a snapshot. The value of this parameter remains unchanged throughout the lifecycle of the disk.

Parameter	Type	Example	Description
Status	String	Available	The status of the disk. Valid values: • In_use • Available • Attaching • Detaching • Creating • Relniting
StorageSetId	String	ss-i-bp1j4i2jdf3owlhe****	 Note This parameter is not supported.
StorageSetPartitionNumber	Integer	11	 Note This parameter is not supported.
Tags	Array of Tag		The tags of the disk.
Tag			
TagKey	String	TestKey	The tag key of the disk.
TagValue	String	TestValue	The tag value of the disk.
Type	String	data	The type of the disk. Valid values: • system: system disk • data: data disk
ZoneId	String	cn-qingdao-env17-amtest11001-a	The zone ID of the disk.
NextToken	String	AAAAAdWBF2****	The query token returned in this call.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	1	The number of entries returned per page.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3CB3E	The ID of the request.
TotalCount	Integer	76	The total number of entries returned.

Examples

Sample requests

```
https://ecs.example.com/?Action=DescribeDisks
&RegionId=cn-qingdao-env17-d01
&PageNumber=1
&PageSize=1
&<Common request parameters>
```

Sample success responses

XML **format**

```

<DescribeDisksResponse>
  <PageNumber>1</PageNumber>
  <TotalCount>76</TotalCount>
  <PageSize>1</PageSize>
  <RequestId>C74847CB-9B69-4360-9969-3595F2B6B9C1</RequestId>
  <Disks>
    <Disk>
      <ImageId>centos_7_06_64_20G_alibase_20190711.vhd</ImageId>
      <Device>/dev/xvda</Device>
      <DetachedTime></DetachedTime>
      <Type>system</Type>
      <InstanceId>i-bp1j4i2jdf3owlhe****</InstanceId>
      <Encrypted>>false</Encrypted>
      <ZoneId>cn-qingdao-env17-amtest11001-a</ZoneId>
      <EnableAutoSnapshot>>true</EnableAutoSnapshot>
      <AttachedTime>2019-11-11T08:35:32Z</AttachedTime>
      <PerformanceLevel></PerformanceLevel>
      <SourceSnapshotId></SourceSnapshotId>
      <DeleteAutoSnapshot>>false</DeleteAutoSnapshot>
      <KMSKeyId></KMSKeyId>
      <Size>40</Size>
      <Description></Description>
      <BdfId></BdfId>
      <ProductCode></ProductCode>
      <Portable>>false</Portable>
      <EnableAutomatedSnapshotPolicy>>false</EnableAutomatedSnapshotPolicy>
      <ResourceGroupId></ResourceGroupId>
      <DiskName></DiskName>
      <StorageSetId></StorageSetId>
      <AutoSnapshotPolicyId></AutoSnapshotPolicyId>
      <CreationTime>2019-11-11T08:35:29Z</CreationTime>
      <MountInstances>
        </MountInstances>
      <Status>In_use</Status>
      <Tags>
        <Tag>
          <TagValue>TestValue</TagValue>
          <TagKey>TestKey</TagKey>
        </Tag>
      </Tags>
      <Category>cloud_efficiency</Category>
      <RegionId>cn-qingdao-env17-d01</RegionId>
      <DeleteWithInstance>>true</DeleteWithInstance>
      <OperationLocks>
        </OperationLocks>
      <DiskId>d-bp18um4r4f2fve24****</DiskId>
    </Disk>
  </Disks>
</DescribeDisksResponse>

```

JSON **format**


```

{
  "PageNumber": 1,
  "TotalCount": 76,
  "PageSize": 1,
  "RequestId": "C74847CB-9B69-4360-9969-3595F2B6B9C1",
  "Disks": {
    "Disk": [
      {
        "ImageId": "centos_7_06_64_20G_alibase_20190711.vhd",
        "Device": "/dev/xvda",
        "DetachedTime": "",
        "Type": "system",
        "InstanceId": "i-bplj4i2jdf3owlhe****",
        "Encrypted": false,
        "ZoneId": "cn-qingdao-env17-amtest11001-a",
        "EnableAutoSnapshot": true,
        "AttachedTime": "2019-11-11T08:35:32Z",
        "PerformanceLevel": "",
        "SourceSnapshotId": "",
        "DeleteAutoSnapshot": false,
        "KMSKeyId": "",
        "Size": 40,
        "Description": "",
        "BdfId": "",
        "ProductCode": "",
        "Portable": false,
        "EnableAutomatedSnapshotPolicy": false,
        "ResourceGroupId": "",
        "DiskName": "",
        "StorageSetId": "",
        "AutoSnapshotPolicyId": "",
        "CreationTime": "2019-11-11T08:35:29Z",
        "MountInstances": {
          "MountInstance": []
        },
        "Status": "In_use",
        "Tags": {
          "Tag": [
            {
              "TagValue": "TestValue",
              "TagKey": "TestKey"
            }
          ]
        },
        "Category": "cloud_efficiency",
        "RegionId": "cn-qingdao-env17-d01",
        "DeleteWithInstance": true,
        "OperationLocks": {
          "OperationLock": []
        },
        "DiskId": "d-bp18um4r4f2fve24****"
      }
    ]
  }
}

```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidDiskType.ValueNotSupported	The specified disk type is not supported.	The error message returned because the specified DiskType parameter is invalid.
400	InvalidCategory.ValueNotSupported	The specified disk category is not supported.	The error message returned because the specified Category parameter is invalid.
400	InvalidStatus.ValueNotSupported	The specified disk status is not supported.	The error message returned because the operation is not supported while the disk is in the current state.
403	InvalidDiskIds.Malformed	The amount of specified disk ids exceeds the limit.	The error message returned because the specified DiskIds parameter is invalid.
404	InvalidDiskChargeType.NotFound	The DiskChargeType does not exist in our records	The error message returned because the specified DiskChargeType parameter does not exist.
404	InvalidLockReason.NotFound	The specified LockReason is not found	The error message returned because the specified LockReason parameter does not exist.
404	InvalidFilterKey.NotFound		The error message returned because the specified start time or end time is invalid.
400	InvalidTag.Mismatch	The specified Tag.n.Key and Tag.n.Value are not match.	The error message returned because the specified Tag.N.Key and Tag.N.Value parameters do not match.

HTTP status code	Error code	Error message	Description
400	InvalidTagCount	The specified tags are beyond the permitted range.	The error message returned because the number of specified tags exceeds the upper limit.
400	InvalidRegion.NotFound	The specified parameter RegionId is not valid.	The error message returned because the specified RegionId parameter is invalid.
500	InternalServerError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
400	InvalidZoneId.NotFound	The zoneId provided does not exist in our records.	The error message returned because the specified ZoneId parameter does not exist.
400	MissingParamter.RegionId	The regionId should not be null.	The error message returned because the RegionId parameter is not specified.
400	InvalidParameter.DiskIds	The specified parameter diskIds is not valid.	The error message returned because the specified DiskIds parameter is invalid.
400	IncompleteParamter	Some fields can not be null in this request.	The error message returned because a required parameter is not specified.
400	InvalidParamter	Some parameters are invalid in this request.	The error message returned because the request contains invalid parameters.
400	InvalidSnapshot.NotFound	The specified parameter SnapshotId is not valid.	The error message returned because the specified SnapshotId parameter is invalid.
403	UserNotInTheWhiteList	The user is not in volume white list.	The error message returned because you are not authorized to manage Shared Block Storage devices. Submit a ticket to apply for the authorization.
404	InvalidDiskIds.ValueNotSupported	The specified parameter "DiskIds" is not supported.	The error message returned because the specified DiskIds parameter is invalid.

3.6.4.3. AttachDisk

Attaches a system disk or data disk to an instance. The instance and the disk must reside in the same zone.

Description

When you call this operation, take note of the following items:

- The disk to be attached must be in the **Unattached** (`Available`) state.
- When the disk is attached as a data disk, take note of the following items:
The instance must be in the **Running** (`Running`) or **Stopped** (`Stopped`) state.
- When the disk is attached as a system disk, take note of the following items:
 - The instance must be the original instance from which the system disk is detached.
 - The instance must be in the **Stopped** (`Stopped`) state.
 - The logon credentials must be configured.
- When you query the instance information and the responses contain `{"OperationLocks": {"LockReason": "security"}}`, the instance is locked for security reasons and all operations cannot take effect on the instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AttachDisk	The operation that you want to perform. Set the value to AttachDisk.
DiskId	String	Yes	d-bp198f7xn673w2rw****	The ID of the disk to be attached. The disk specified by the <code>DiskId</code> parameter and the instance specified by the <code>InstanceId</code> parameter must reside in the same zone. Note For more information about the limits on attaching a data disk and system disk, see the "Description" section of this topic.
InstanceId	String	Yes	i-bp1ahcvgu9z55pk7****	The ID of the instance to which to attach the disk.

Parameter	Type	Required	Example	Description
Device	String	No	testName	The device name of the disk. Note This parameter will be removed in the future. We recommend that you use other parameters to ensure future compatibility.
DeleteWithInstance	Boolean	No	false	Specifies whether to release the disk along with the instance to which it is attached. Valid values: - true: releases the disk when the instance is released. - false: does not release the disk when the instance is released. Default value: false. Note If <code>OperationLocks</code> in the <code>DescribeInstances</code> response contains <code>"LockReason" : "security"</code> for the instance to which the disk is attached, the instance is locked for security reasons. In this case, the disk is released along with the instance even if <code>DeleteWithInstance</code> is set to <code>false</code>.
Bootable	Boolean	No	false	Specifies whether to attach the disk as a system disk. Default value: false. Note You can set <code>Bootable</code> to true only when the instance is in the No System Disk state.
Password	String	No	EcsV587!	The password set when you attach the system disk. The password is applicable only to the administrator and root users. The password must be 8 to 30 characters in length and contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include: <pre>()`~!@#%\$^&*~_+= {}[];':"<>.,?/</pre> For Windows instances, the password cannot start with a forward slash (/). Note For security reasons, we recommend that you use HTTPS to send requests if the <code>Password</code> parameter is specified.
KeyPairName	String	No	test-KeyPairName	The name of the SSH key pair that you bind to the Linux instance when you attach the system disk. - Windows instances do not support logons based on SSH key pairs. The <code>Password</code> parameter takes effect even if the <code>KeyPairName</code> parameter is specified. - For Linux instances, the username and password logon method is disabled by default.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3CB3E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=AttachDisk
&DiskId=d-bp198f7xn673w2rw****
&InstanceId=i-bp1ahcvgu9z55pk7****
&DeleteWithInstance=false
&<Common request parameters>
```

Sample success responses

XML **format**

```
<AttachDiskResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E</RequestId>
</AttachDiskResponse>
```

JSON **format**

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified InstanceId parameter does not exist. Check whether the instance ID is correct.
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
400	InvalidDevice.Malformed	The specified device is not valid.	The error message returned because the specified Device parameter does not exist.
403	InstanceDiskLimitExceeded	The amount of the disk on instance in question reach its limits.	The error message returned because the maximum number of disks that can be attached to the specified instance has been reached.
403	InvalidDevice.InUse	The specified device has been occupied.	The error message returned because the specified device already has a disk attached.
403	IncorrectDiskStatus	The operation is not supported in this status.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
403	DiskNotPortable	The specified disk is not a portable disk.	The error message returned because the specified disk is not removable.
403	InstanceLockedForSecurity	The instance is locked due to security.	The error message returned because the operation is not supported while the instance is locked for security reasons.
403	ResourcesNotInSameZone	The specified instance and disk are not in the same zone.	The error message returned because the specified instance and disk are not in the same zone.
403	InstanceExpiredOrInArrears	The specified operation is denied as your prepay instance is expired (prepay mode) or in arrears (afterpay mode).	The error message returned because the subscription instance has expired. Renew the instance and try again.
403	DiskInArrears	The specified operation is denied as your disk owing fee.	The error message returned because you have an overdue payment for the specified disk.
400	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the instance is in the current state.
403	DiskError	IncorrectDiskStatus.	The error message returned because the state of the specified disk is invalid.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
400	InvalidParameter	The input parameter is mandatory for processing this request is empty.	The error message returned because a required parameter is not specified.
403	DiskId.ValueNotSupported	The specified parameter diskid is not supported.	The error message returned because the category of the specified Elastic Block Storage (EBS) device does not support this operation.
403	DiskId.StatusNotSupported	The specified disk status is not supported.	The error message returned because this operation is not supported while the disk is in the current state.
404	InvalidDisk.InUse	The specified disk has been occupied.	The error message returned because the specified disk is in use.
404	DiskAttachedNumberExceeded	The attaching times of the specified disk exceeded.	The error message returned because the maximum number of instances to which the specified disk can be attached has been reached.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are not authorized to manage the disk. Try again when you are authorized.

HTTP status code	Error code	Error message	Description
400	InvalidRegionId.MalFormed	The specified RegionId is not valid	The error message returned because the specified RegionId parameter is invalid.
400	InvalidOperation.InstanceTypeNotSupport	The instance type of the specified instance does not support hot attach disk.	The error message returned because the instance to which the disk is attached does not support the hot swapping of disks.
400	DiskCategory.OperationNotSupported	The operation is not supported to the specified disk due to its disk category	The error message returned because the specified disk category does not support the operation.
400	InvalidOperation.InstanceTypeNotSupport	The specified disk which has kms key should only attach to ioOptimized instance.	The error message returned because the specified instance is not supported. You can set Key Management Service (KMS) keys only for I/O optimized instances.
403	InvalidParameter.KMSKeyId.CMKNotEnabled	The CMK needs to be enabled.	The error message returned because the customer master key (CMK) is not enabled when a KMS key ID is specified for a disk. You can call the DescribeKey operation of KMS to query the information of the specified CMK.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service have no right to access your KMS.	The error message returned because Elastic Compute Service (ECS) is not authorized to access your KMS resources.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
400	InvalidParameter.AllEmpty	%s	The error message returned because the required parameters are not specified.
403	InvalidInstanceType.NotSupportDiskCategory	The instanceType of the specified instance does not support this disk category.	The error message returned because the instance type specified by the InstanceType parameter does not support the disk category of the instance. Try another instance type. For information about the disk categories supported by instance types, see the instance family documentation.

3.6.4.4. DetachDisk

You can call this operation to detach a data disk or system disk from an instance.

Description

When you call this operation, take note of the following items:

- The disk to be detached must be in the **Running** (`InUse`) state.
- The instance from which you want to detach a data disk must be in the **Running** (`Running`) or **Stopped** (`Stopped`) state.
- The instance from which you want to detach the system disk must be in the **Stopped** (`Stopped`) state.
- If the API response contains `{"OperationLocks": {"LockReason": "security"}}` when you query information of an instance, the instance is locked for security reasons and all operations are prohibited on it.
- DetachDisk is an asynchronous operation. It takes about one minute for the disk to be detached after this operation is called.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DetachDisk	The operation that you want to perform. Set the value to DetachDisk.
DiskId	String	Yes	d-bp67acfmazb4p****	The ID of the disk to be detached.
InstanceId	String	Yes	i-bp67acfmazb4p****	The ID of the instance from which the disk is to be detached.
DeleteWithInstance	Boolean	No	false	The automatic release attribute of the system disk. That is, this parameter specifies whether the system disk is released when its attached instance is released. Valid values: - true: The system disk is released when its attached instance is released. - false: The system disk is not released but is retained when its attached instance is released. Default value: true.  Note If a data disk is to be detached, the default value is <code>false</code>.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0E3C83E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=DetachDisk
&DiskId=d-bp67acfmazb4p****
&InstanceId=i-bp67acfmazb4p****
&<Common request parameters>
```

Sample success responses

XML **format**

```
<DetachDiskResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0E3C83E</RequestId>
</DetachDiskResponse>
```

JSON **format**

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0E3C83E"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified InstanceId parameter does not exist.
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Ensure that the disk is available and you have no overdue payment for it.
403	DiskNotPortable	The specified disk is not a portable disk.	The error message returned because the specified disk is not removable.
403	InstanceLockedForSecurity	The instance is locked due to security.	The error message returned because the operation is not supported while the resource is locked for security reasons.
403	DependencyViolation	The specified disk has not been attached on the specified instance.	The error message returned because the operation is not supported while the resource has other dependencies. For example, the specified disk is not attached to the specified instance, or the specified security group cannot be deleted while it contains instances.
403	DiskTypeViolation	The specified disk is a system disk and cannot support the operation.	The error message returned because the specified disk is a system disk and cannot be detached from the associated instance.
403	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the resource is in the current state.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
400	InvalidParameter	The input parameter is mandatory for processing this request is empty.	The error message returned because a required parameter is not specified.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are not authorized to manage the disk.
400	InvalidRegionId.MalFormed	The specified RegionId is not valid	The error message returned because the specified region ID is invalid.
404	InvalidDisk.AlreadyDetached	The specified disk has been detached.	The error message returned because the specified disk has been detached.

HTTP status code	Error code	Error message	Description
400	InvalidOperation.InstanceTypeNotSupport	The instance type of the specified instance does not support hot detach disk.	The error message returned because the instance to which the disk is attached does not support the hot swapping of disks.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.

3.6.4.5. ModifyDiskAttribute

You can call this operation to modify the attributes of one or more Block Storage devices, including their names, descriptions, and whether they are released along with their attached instances.

Description

If you set `DeleteWithInstance` to false for a disk and the instance to which the disk is attached is locked for security reasons, the `DeleteWithInstance` parameter will be ignored and the disk will be released along with the instance. (If the value of `LockReason` is security in `OperationLocks` of the API response when you query information of an instance, the instance is locked for security reasons.)

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyDiskAttribute	The operation that you want to perform. Set the value to <code>ModifyDiskAttribute</code> .
DiskId	String	No	d-bp198f7xn673w2rw****	The ID of the disk to be modified. Note You can specify either <code>DiskId</code> or <code>DiskIds.N</code> but you cannot specify both of them.
DiskIds.N	RepeatList	No	d-bp198f7xn673w2rw****	The ID of disk N to be modified. Valid values of N: 0 to 100. Note You can specify either <code>DiskId</code> or <code>DiskIds.N</code> but you cannot specify both of them.
DiskName	String	No	test-DiskName	The name of the disk. The name must be 2 to 128 characters in length, and can contain letters, digits, colons (:), underscores (_), and hyphens (-). It must start with a letter and cannot start with <code>http://</code> or <code>https://</code> .
Description	String	No	This is a description example.	The description of the disk. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
DeleteWithInstance	Boolean	No	false	Specifies whether the disk is released with the instance. This parameter is empty by default, which indicates that the current value remains unchanged. An error will be returned if you set this parameter to false in the following cases: - The specified disk is a local disk. - The specified disk is a basic disk and is not removable.
DeleteAutoSnapshot	Boolean	No	false	Specifies whether the automatic snapshots of the disk are deleted when it is released. This parameter is empty by default, which indicates that the current value remains unchanged.
EnableAutoSnapshot	Boolean	No	true	Specifies whether to apply a created automatic snapshot policy to the disk. This parameter is empty by default, which indicates that the current value remains unchanged.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ModifyDiskAttribute
&DiskId=d-bp198f7xn673w2rw***
&DiskName=test-DiskName
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifyDiskAttributeResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E</RequestId>
</ModifyDiskAttributeResponse>
```

JSON format

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified disk does not exist. Check whether the disk ID is correct.
400	InvalidDiskName.Malformed	The specified disk name is wrongly formed.	The error message returned because the specified DiskName parameter is invalid. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter and cannot start with http:// or https://.
404	InvalidDescription.Malformed	The specified description is wrongly formed.	The error message returned because the specified Description parameter is invalid. The description must be 2 to 256 characters in length and cannot start with http:// or https://.
400	NoAttributeToModify	No attribute to be modified in this request.	The error message returned because no attributes of the specified disk are modified.
403	QuotaExceed.Snapshot	The snapshot quota exceeds.	The error message returned because the maximum number of snapshots has been reached. To store new snapshots, delete snapshots that are no longer needed.
403	DiskNotPortable	The specified disk is not a portable disk.	The error message returned because the specified disk is not removable.
403	IncorrectDiskStatus	The operation is not supported in this status.	The error message returned because the operation is not supported while the disk is in the current state. Ensure that the disk is available and you have no overdue payment for it.
400	IncompleteParamter	Some fields can not be null in this request.	The error message returned because some required parameters are not specified.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are not authorized to manage the disk.
400	InvalidRegionId.MalFormed	The specified RegionId is not valid	The error message returned because the specified region ID is invalid.
404	InvalidInstanceId.NotFound	Specified attached instance does not exist.	The error message returned because the specified instance ID does not exist.

3.6.4.6. ReInitDisk

Re-initializes a disk.

Description

When you call this operation, take note of the following items:

- The disk to be re-initialized must be in the **In Use** (In_use) state and the instance to which the disk is attached must be in the **Stopped** (Stopped) state.
- If the instance has never been started since it was created, the disks attached to it cannot be re-initialized.
- When a system disk is re-initialized, it is restored to the state of the image from which it was created. If the image has been deleted, the disk cannot be re-initialized.
- When a separately created data disk is re-initialized, it is restored to an empty data disk.
- When a data disk that was created from a snapshot is re-initialized, the disk is restored to the state of the snapshot. If the snapshot has been deleted, the disk cannot be re-initialized and an error is returned.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ReInitDisk	The operation that you want to perform. Set the value to ReInitDisk.
DiskId	String	Yes	d-bp67acfmazb4p****	The ID of the disk.
Password	String	No	EcsV587!	Specifies whether to reset the password of the instance when you re-initialize its system disk. The password must be 8 to 30 characters in length and contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include: <pre>() ~ ! @ # \$ % ^ & * _ + = { } [] : ; ' < > , . ? /</pre> For Windows instances, the password cannot start with a forward slash (/). Note For security reasons, we recommend that you use HTTPS to send requests if the <code>Password</code> parameter is specified.
KeyPairName	String	No	test-KeyPairName	The name of the key pair. Note The parameter is applicable only to Linux instances. You can bind an SSH key pair to an instance as the logon credential when you re-initialize its system disk. After the SSH key pair is bound, the username and password logon method is disabled for the instance.
AutoStartInstance	Boolean	No	true	Specifies whether to automatically start the instance after the disk is re-initialized. Default value: false.
SecurityEnhancementStrategy	String	No	Active	Specifies whether to use Security Center after the system disk is re-initialized. Valid values: - Active: uses Security Center after the system disk is re-initialized. This value is applicable only to public images. - Deactive: does not use Security Center after the system disk is re-initialized. This value is applicable to all images. Default value: Deactive.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3****	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ReInitDisk
&DiskId=d-bp67acfmazb4p****
&KeyPairName=test-KeyPairName
&AutoStartInstance=true
&SecurityEnhancementStrategy=Active
&<Common request parameters>
```

Sample success responses

XML format

```
<ReInitDiskResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3****</RequestId>
</ReInitDiskResponse>
```

JSON format

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
403	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the instance is in the current state.
403	InstanceLockedForSecurity	The instance is locked due to security.	The error message returned because the operation is not supported while the instance is locked for security reasons.
403	InvalidSnapshot.TooOld	The disk is created from a snapshotId made before 2013-07-15, it cannot be re-initiated the specified disk any more since the detached first time.	The error message returned because snapshots created before July 15, 2013 do not support this operation.
403	OperationDenied	The snapshot which is used to create the specified disk has been deleted.	The error message returned because the snapshot used to create the specified disk does not exist.
403	InstanceExpiredOrInArrears	The specified operation is denied as your prepay instance is expired (prepay mode) or in arrears (afterpay mode).	The error message returned because the subscription instance has expired. Renew the instance and try again.
403	DiskCreatingSnapshot	The operation is denied due to a snapshot of the specified disk is not completed yet.	The error message returned because a snapshot is being created for the specified disk.
403	InvalidSourceSnapshot	The snapshot which is used to create the specified disk has been deleted.	The error message returned because the snapshot used to create the specified disk has been deleted.
404	InvalidImageId.NotFound	The specified ImageId does not exist.	The error message returned because the specified image does not exist in this account. Check whether the image ID is correct.
403	SharedImageDeleted	The specified image by others shared is deleted.	The error message returned because the specified shared image has been deleted.
400	InvalidPassword.Malformed	The specified parameter "Password" is not valid.	The error message returned because the specified Password parameter is invalid.
400	DiskCategory.OperationNotSupported	The operation is not supported to the specified disk due to its disk category	The error message returned because the specified disk category does not support this operation.
400	InvalidKeyPairName.NotFound	The specified KeyPairName does not exist.	The error message returned because the specified KeyPairName parameter does not exist.
400	DependencyViolation.IoOptimize	The specified parameter InstanceId is not valid.	The error message returned because the I/O optimization configuration of the instance is invalid.
403	UserNotInTheWhiteList	The user is not in volume white list.	The error message returned because you are not authorized to manage the Shared Block Storage device.
400	InvalidRegionId.Malformed	The specified RegionId is not valid	The error message returned because the specified RegionId parameter is invalid.
404	InvalidDiskId.OperationNotSupported	The operation is not supported due to image not exist.	The error message returned because the specified ImageId parameter does not exist.
403	InvalidParameter.KMSKeyId.CMKNotEnabled	The CMK needs to be enabled.	The error message returned because the customer master key (CMK) is not enabled when a KMS key ID is specified for a disk. You can call the DescribeKey operation of KMS to query the information of the specified CMK.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service have no right to access your KMS.	The error message returned because Elastic Compute Service (ECS) is not authorized to access your KMS resources.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.

3.6.4.7. ResetDisk

Restores a disk to a specific point in time from a snapshot of the disk.

Description

When you call this operation, take note of the following items:

- The disk must be in the In Use (In_Use) state.

- The instance to which the disk is attached must be in the Stopped (Stopped) state.
- The snapshot specified by the SnapshotId parameter must be created from the disk specified by the DiskId parameter.
- When you call the DescribeInstances operation to query the instance information and the responses contain `{"OperationLocks": {"LockReason": "security"}}`, the instance is locked for security reasons and all operations cannot take effect on the instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ResetDisk	The operation that you want to perform. Set the value to ResetDisk.
DiskId	String	Yes	d-bp198f7xn673w2rw****	The ID of the disk.
SnapshotId	String	Yes	s-bp1btgnlb4l1cd41****	The ID of the snapshot that is used to restore the disk.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3CB3E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ResetDisk
&DiskId=d-bp198f7xn673w2rw****
&SnapshotId=s-bp1btgnlb4l1cd41****
&<Common request parameters>
```

Sample success responses

XML **format**

```
<ResetDiskResponse>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
</ResetDiskResponse>
```

JSON **format**

```
{
  "RequestId": "F3CD6886-D8D0-4FEE-B93E-1B73239673DE"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
404	Disk.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
404	InvalidSnapshotId.NotFound	The specified SnapshotId does not exist.	The error message returned because the specified SnapshotId parameter does not exist. Check whether the snapshot ID is correct.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
403	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the instance is in the current state.
403	InstanceLockedForSecurity	The instance is locked due to security.	The error message returned because the operation is not supported while the instance is locked for security reasons.
403	InvalidParameter.Mismatch	The specified snapshot is not created from the specified disk.	The error message returned because an encrypted snapshot cannot be used to roll back an unencrypted disk.
403	InvalidSnapshot.TooOld	The snapshotId is created before 2013-07-15, it cannot be restored since the first time the disk detached.	The error message returned because snapshots created before July 15, 2013 do not support this operation.

HTTP status code	Error code	Error message	Description
403	InstanceExpiredOrInArrears	The specified operation is denied as your prepaid instance is expired (prepay mode) or in arrears (afterpay mode).	The error message returned because the subscription instance has expired. Renew the instance and try again.
403	OperationDenied	The specified snapshot does not support ResetDisk.	The error message returned because the specified snapshot does not support this operation.
403	InvalidSnapshotId.NotReady	The specified snapshot has not completed yet.	The error message returned because the specified snapshot is being created.
500	InternalServerError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
400	DiskCategory.OperationNotSupported	The operation is not supported to the specified disk due to its disk category.	The error message returned because the specified disk category does not support this operation.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient. Add funds to your account and try again.
403	InvalidAccountStatus.SnapshotServiceUnavailable	Snapshot service has not been opened yet.	The error message returned because the operation is not supported while the snapshot service is not activated.
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified InstanceId parameter does not exist. Check whether the instance ID is correct.
403	Operation.Conflict	The operation may conflict with others.	The error message returned because the operation conflicts with other operations.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are not authorized to manage the disk. Try again when you are authorized.
400	InvalidRegionId.Malformed	The specified RegionId is not valid.	The error message returned because the specified RegionId parameter is invalid.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
403	InvalidParameter.KMSKeyId.CMKNotEnabled	The CMK needs to be enabled.	The error message returned because the customer master key (CMK) is not enabled when a Key Management Service (KMS) key ID is specified for a disk. You can call the DescribeKey operation of KMS to query the information of the specified CMK.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service has no right to access your KMS.	The error message returned because Elastic Compute Service (ECS) is not authorized to access your KMS resources.

3.6.4.8. ReplaceSystemDisk

You can call this operation to replace the system disk or operating system of an ECS instance. After the system disk is replaced, the ID of the system disk changes and the original disk is released.

Description

When you call this operation, take note of the following items:

- You cannot change the category of the system disk.
- The instance to which the system disk is to be replaced is attached must be in the Stopped (Stopped) state.
- If OperationLocks in the DescribeInstances response contains "LockReason" : "security" for an instance, the instance is locked for security reasons and cannot be restarted.
- You can use the SystemDisk.Size parameter to specify the capacity of the new system disk.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ReplaceSystemDisk	The operation that you want to perform. Set the value to ReplaceSystemDisk.
InstanceId	String	Yes	i-bp67acfmazb4p****	The ID of the instance.
ImageId	String	No	m-bp67acfmazb4p****	The ID of the image to be used to replace the system disk.

Parameter	Type	Required	Example	Description
SystemDisk.Size	Integer	No	80	The capacity of the new system disk. Unit: GiB. Valid values: Max(20, the size of the image specified by ImageId) to 500. Default value: 40 or the size of the specified image, whichever is greater.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that it is unique among different requests. The ClientToken value can only contain ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence in ECS Developer Guide .
UseAdditionalService	Boolean	No	true	Specifies whether to use the virtual machine system configuration provided by Apsara Stack (Windows: NTP and KMS. Linux: NTP and YUM). Note This parameter takes effect only when you attach a system disk, whose device name is /dev/xvda.
Password	String	No	EcsV587!	The new password of the instance. The password must be 8 to 30 characters in length. It must include at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include: () ^ ~ ! @ # \$ % ^ & * - _ + = { } [] : ; ' < > , . ? / The password of a Windows instance cannot start with a forward slash (/). This parameter is empty by default, which indicates that the current password remains unchanged. Note If the <code>Password</code> parameter is specified, we recommend that you send requests over HTTPS to avoid password leaks.
PasswordInherit	Boolean	No	false	Specifies whether to use the password preset in the image. Default value: false. Note If the PasswordInherit parameter is set to true, you must not specify the Password parameter and make sure that the selected image has a password preset.
KeyPairName	String	No	test-KeyPairName	The name of the key pair. Note This parameter is applicable only to Linux instances. You can bind an SSH key pair to the instance as a logon credential. However, the password logon method is disabled after you bind the SSH key pair.
Platform	String	No	CentOS	The distribution of the operating system. Valid values: - CentOS - Ubuntu
Architecture	String	No	i386	The system architecture. Valid values: - i386 - x86_64
SecurityEnhancementStrategy	String	No	Active	Specifies whether to use Security Center after you replace the system disk. Valid values: - Active: Security Center is used for free after the system disk is replaced. This value is applicable only to public images. - Deactive: Security Center is not used for free after the system disk is replaced. This value is applicable to all images. Default value: Deactive.

Response parameters

Parameter	Type	Example	Description
DiskId	String	d-bp67acfmazb4p****	The ID of the new system disk.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ReplaceSystemDisk
&InstanceId=i-bp67acfmazb4p****
&ImageId=m-bp67acfmazb4p****
&SystemDisk.Size=80
&ClientToken=123e4567-e89b-12d3-a456-426655440000
&PasswordInherit=false
&KeyPairName=test-KeyPairName
&SecurityEnhancementStrategy=Active
&<Common request parameters>
```

Sample success responses

XML format

```
<ReplaceSystemDiskResponse>
  <DiskId>d-bp67acfmazb4p****</DiskId>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
</ReplaceSystemDiskResponse>
```

JSON format

```
{
  "RequestId": "337568C5-64F3-4B76-8CDD-D3D8C57B5B8C",
  "DiskId": "d-bp67acfmazb4p****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified InstanceId parameter does not exist.
400	InvalidSystemDiskSize.ValueNotSupported	The specified parameter SystemDisk.Size is invalid.	The error message returned because the specified SystemDisk.Size parameter is invalid.
404	InvalidInstanceId.NotFound	The specified instance does not exist.	The error message returned because the specified InstanceId parameter does not exist.
403	InvalidSystemDiskStatus.IsTransferring	The current status of the resource does not support this operation, system disk is transferring.	The error message returned because the operation is not supported while the resource is in the current state. Try again after the system disk stops transmitting data.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
404	InvalidImageId.NotFound	The specified ImageId does not exist.	The error message returned because the specified image does not exist in this account. Check whether the image ID is correct.
403	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the resource is in the current state.
403	InstanceLockedForSecurity	The instance is locked due to security.	The error message returned because the operation is not supported while the resource is locked for security reasons.
403	ImageNotSubscribed	The specified image has not be subscribed.	The error message returned because you have not subscribed to the specified Alibaba Cloud Marketplace image.
403	ImageRemovedInMarket	The specified market image is not available, Or the specified user defined image includes product code because it is based on an image subscribed from marketplace, and that image in marketplace including exact the same product code has been removed.	The error message returned because the specified Alibaba Cloud Marketplace image is unavailable, or because the specified custom image contains the product code of the Alibaba Cloud Marketplace image from which the custom image is derived and the Alibaba Cloud Marketplace image has been removed from Alibaba Cloud Marketplace.

HTTP status code	Error code	Error message	Description
500	OperationDenied	Internal Error.	The error message returned because an internal error has occurred.
400	InvalidParameter.Conflict	The specified image does not support the specified instance type.	The error message returned because the specified image does not support the specified instance type.
404	InvalidSystemDiskSize.MoreThanMaxSize	The specified SystemDisk.Size parameter exceeds the maximum size.	The error message returned because the specified system disk size exceeds the allowed maximum size.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
403	InstanceExpiredOrInArrears	The specified operation is denied as your prepaid instance is expired (prepay mode) or in arrears (afterpay mode).	The error message returned because the subscription instance has expired. Renew the instance first.
403	ChargeTypeViolation	The operation is not permitted due to charge type of the instance.	The error message returned because the billing method of the instance does not support this operation.
403	DiskCreatingSnapshot	The operation is denied due to a snapshot of the specified disk is not completed yet.	The error message returned because a snapshot is being created for the specified disk.
403	IoOptimized.NotSupported	The specified image is not support IoOptimized Instance.	The error message returned because the specified image does not support I/O optimized instances.
403	ImageNotSupportInstanceType	The specified image does not support the InstanceType instance.	The error message returned because the specified image does not support the specified instance type.
403	QuotaExceed.BuyImage	The specified image is from the image market, You have not bought it or your quota has been exceeded.	The error message returned because you have not purchased the specified Alibaba Cloud Marketplace image or your image quota has been used up.
404	InvalidSystemDiskSize.LessThanImageSize	The specified parameter SystemDisk.Size is less than the image size.	The error message returned because the specified system disk size is smaller than the image size.
404	InvalidSystemDiskSize.LessThanMinSize	The specified parameter SystemDisk.Size is less than the min size.	The error message returned because the specified system disk size is smaller than the allowed minimum size.
404	NoSuchResource	The specified resource is not found.	The error message returned because the specified resource does not exist.
400	InvalidSystemDiskSize.ImageNotSupportResize	The specified image does not support resize.	The error message returned because the specified image does not support resizing.
400	InvalidSystemDiskSize	The specified parameter SystemDisk.Size is invalid.	The error message returned because the specified SystemDisk.Size parameter is invalid.
403	INST_HAS_UNPAID_ORDER	The instance has unpaid order.	The error message returned because you have unpaid orders for the instance.
400	InvalidSystemDiskSize.ValueNotSupported	The specified parameter SystemDisk.Size is invalid	The error message returned because the specified SystemDisk.Size parameter is invalid.
400	InvalidPassword.Malformed	The specified parameter "Password" is not valid.	The error message returned because the specified Password parameter is invalid.
400	InvalidPasswordParam.Mismatch	The input password should be null when passwdInherit is true.	The error message returned because the Password parameter is specified when PasswdInherit is set to true.
400	OperationDenied	The specified image contains the snapshot of the data disk, does not support this operation.	The error message returned because the operation is not supported while the image contains data disk snapshots.
400	InvalidDiskCategory.ValueNotSupported	The specified parameter "DiskCategory" is not valid.	The error message returned because the specified disk category is invalid.
403	OperationDenied.InstanceCreating	The specified instance is creating.	The error message returned because the specified instance is being created.
400	InvalidParameter.Conflict	%s	The error message returned because a specified parameter is invalid. Check whether parameter conflicts exist.
400	InvalidSystemDiskSize.ValueNotSupported	%s	The error message returned because the specified SystemDisk.Size parameter is invalid.
400	OperationDenied	%s	The error message returned because the operation is denied.
400	InvalidKeyPairName.NotFound	The specified KeyPairName does not exist.	The error message returned because the specified KeyPairName parameter does not exist.

HTTP status code	Error code	Error message	Description
400	DependencyViolation.IoOptimize	The specified parameter InstanceId is not valid.	The error message returned because the specified InstanceId parameter is invalid.
403	InvalidParameter.NotMatch	%s	The error message returned because a specified parameter is invalid. Check whether parameter conflicts exist.
400	MissingParameter.Architecture	Architecture should not be null.	The error message returned because the Architecture parameter is not specified.
400	InvalidArchitecture.Malformed	Architecture is not valid.	The error message returned because the specified Architecture parameter is invalid.
400	MissingParameter.Platform	Platform should not be null.	The error message returned because the Platform parameter is not specified.
400	InvalidPlatform.Malformed	Platform is not valid.	The error message returned because the specified Platform parameter is invalid.
400	InvalidParameter.AllEmpty	%s	The error message returned because the required parameters are not specified.
400	InvalidDiskId.NotFound	The specified disk do not exist.	The error message returned because the specified disk does not exist.
400	InvalidDataDisk.DiskStatusViolation	The operation is not permitted due to status of the DataDisk.	The error message returned because the operation is not supported while the data disk is in the current state.
400	InvalidDataDisk.DiskCategoryViolation	The operation is not permitted due to category of the DataDisk.	The error message returned because the category of the data disk does not support the operation.
403	ResourcesNotInSameZone	The specified instance and disk are not in the same zone.	The error message returned because the specified instance and disk are not in the same zone.
400	InvalidSystemDiskSize.ValueNotSupported	The specified SystemDiskSize is not valid.	The error message returned because the specified SystemDisk.Size parameter is invalid.
400	MissingParameter	The input parameter "ImageId" that is mandatory for processing this request is not supplied.	The error message returned because the ImageId parameter is not specified.
403	ImageNotSupportInstanceType	The specified instanceType is not supported by instance with marketplace image.	The error message returned because the specified Alibaba Cloud Marketplace image does not support the instance type.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
403	OperationDenied.ImageNotValid	%s	The error message returned because the image does not support the operation.

3.6.4.9. ResizeDisk

You can call this operation to resize a system disk or data disk.

Description

Note Before you call this operation to resize a disk, you must check the partition format of the disk. You cannot resize an MBR disk to larger than 2 TiB in size without data loss. To resize an MBR disk to larger than 2 TiB in size, we recommend that you create and attach a new data disk with the desired size. Then, we recommend that you use the GPT partition format to partition and format the new data disk and copy data from the MBR-formatted data disk to the new GPT-formatted data disk. For more information, see [Expand a disk in ECS User Guide](#).

- You can resize the disks of different categories, including basic disks (`cloud`), ultra disks (`cloud_efficiency`), and standard SSDs (`cloud_ssd`).
- You cannot resize a disk when a snapshot is being created for the disk.
- The instance to which the disk to be resized is attached must be in the **Running** (`Running`) or **Stopped** (`Stopped`) state.
- After you resize a disk, its partitions and file systems are not changed. You must manually allocate the storage space on the disk after it is resized.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ResizeDisk	The operation that you want to perform. If you use a custom HTTP URL or HTTPS URL to make an API request, you must specify the <code>Action</code> parameter. Set the value to <code>ResizeDisk</code> .
DiskId	String	Yes	d-bp67acfmazb4p****	The ID of the disk.

Parameter	Type	Required	Example	Description
NewSize	Integer	Yes	1900	The new disk capacity. Unit: GiB. Valid values: - For ultra disks (cloud_efficiency): 20 to 32768 - For standard SSDs (cloud_efficiency): 20 to 32768 - For basic disks (cloud): 5 to 2000 The new disk capacity must be greater than the original disk capacity.
Type	String	No	offline	The method used to resize the disk. Default value: offline. Valid values: - offline: resizes the disk offline. After you resize a disk offline, you must restart its associated instance by using the ECS console or by calling the RebootInstance operation to make the resizing operation take effect. - online: resizes the disk online. After you resize a disk online, the operation takes effect immediately without the need to restart the instance. You can resize ultra disks and standard SSDs online.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that it is unique among different requests. The ClientToken value can only contain ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence in ECS Developer Guide.

Response parameters

Parameter	Type	Example	Description
RequestId	String	F3CD6886-D8D0-4FEE-B93E-1B73239673DE	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ResizeDisk
&DiskId=d-bp67acfmxb4p****
&NewSize=1900
&ClientToken=123e4567-e89b-12d3-a456-426655440000
&<Common request parameters>
```

Sample success responses

XML **format**

```
<ResizeDiskResponse>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
</ResizeDiskResponse>
```

JSON **format**

```
{
  "RequestId": "F3CD6886-D8D0-4FEE-B93E-1B73239673DE"
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	InvalidDataDiskSize.ValueNotSupported	The specified DataDisk.N.Size beyond the permitted range, or the capacity of snapshot exceeds the size limit of the specified disk category.	The error message returned because the specified DataDisk.N.Size parameter is invalid or the snapshot size exceeds the maximum capacity that is allowed for the specified category.
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified instance does not exist.
403	OperationDenied	The type of the disk does not support the operation.	The error message returned because the specified disk category does not support the operation.
403	OperationDenied	The status of the disk or the instance that the disk is attaching with does not support the operation.	The error message returned because the operation is not supported while the disk or instance is in the current state.

HTTP status code	Error code	Error message	Description
403	InvalidDiskSize.TooSmall	Specified new disk size is less than the original disk size.	The error message returned because the specified new disk capacity is smaller than the original disk capacity.
403	InvalidDiskSize.TooLarge	Specified new disk size is beyond the permitted range.	The error message returned because the specified new disk capacity exceeds the capacity limit.
403	InstanceExpiredOrInArrears	The specified operation is denied as your prepay instance is expired (prepay mode) or in arrears (afterpay mode).	The error message returned because the subscription instance has expired. Renew the instance first.
403	DiskError	IncorrectDiskStatus	The error message returned because the status of the specified disk is invalid.
403	DiskInArrears	The specified operation is denied as your disk owing fee.	The error message returned because you have an overdue payment for the disk.
403	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the resource is in the current state.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
403	DiskCreatingSnapshot	The operation is denied due to a snapshot of the specified disk is not completed yet.	The error message returned because a snapshot is being created for the specified disk.
400	InvalidDataDiskCategory.ValueNotSupported	%s	The error message returned because the specified data disk category is invalid.
400	InvalidParameter.Conflict	%s	The error message returned because a specified parameter is invalid. Check whether parameter conflicts exist.
400	InvalidDataDiskSize.ValueNotSupported	%s	The error message returned because the specified data disk capacity is invalid.
400	InvalidSystemDiskSize.ValueNotSupported	%s	The error message returned because the specified system disk size is invalid.
403	InvalidDiskSize	Specified new disk size is less than or equal the original disk size.	The error message returned because the specified new disk capacity is smaller than or equal to the original disk capacity.
400	IncompleteParameter	Some fields can not be null in this request.	The error message returned because some required parameters are not specified.
403	Operation.Conflict	The operation may conflicts with others.	The error message returned because the operation conflicts with other operations.
403	InstanceLockedForSecurity	The instance is locked due to security.	The error message returned because the operation is not supported while the resource is locked for security reasons.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payment for it.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are authorized to manage the disk. Try again when you are authorized.
400	InvalidRegionId.Malformed	The specified RegionId is not valid	The error message returned because the specified RegionId parameter is invalid.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk category does not support the operation.
400	InvalidParam.Type	The specified type is not supported.	The error message returned because the specified Type parameter is invalid.
403	InvalidRegion.NotSupport	The specified region does not support resize online.	The error message returned because online resizing is not supported in the specified region.
403	InvalidDiskCategory.NotSupported	The specified disk category does not support resize online.	The error message returned because the specified disk category does not support online resizing.
403	IncorrectInstanceStatus	The current status of the resource does not support resize online.	The error message returned because this operation is not supported while the resource is in the current state.
403	InvalidInstanceStatus.NotRunning	The status of instance to which the disk attaches must be running when resizing online.	The error message returned because the instance to which the disk is attached is not in the Running state when you resize the disk online.

HTTP status code	Error code	Error message	Description
403	IncorrectDiskStatus	The current status of the resource does not support resize online	The error message returned because online resizing is not supported while the resource is in the current state.
400	LastOrderProcessing	The previous order is still processing, please try again later.	The error message returned because the order is being processed. Try again later.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service have no right to access your KMS.	The error message returned because ECS is not authorized to access your KMS resources.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
403	SecurityRisk.3DVerification	We have detected a security risk with your default credit or debit card. Please proceed with verification via the link in your email.	The error message returned because risks are detected in your default credit card or debit card. Click the link in the email for verification.
400	InvalidSystemDiskSize.ImageNotSupportResize	The specified image does not support resize.	The error message returned because the specified image does not support resizing.

3.6.4.10. DeleteDisk

Releases a data disk.

Description

When you call this operation to release a data disk, take note of the following items:

- Manual snapshots of the disk are retained.
- You can call the `ModifyDiskAttribute` operation to configure whether to retain automatic snapshots of the disk or release them along with the disk. We recommend that you delete the snapshots that are no longer needed to maintain a sufficient quota for new automatic snapshots.
- The disk must be in the `Unattached (Available)` state.
- If the specified `DiskId` parameter does not exist, the request is ignored.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteDisk	The operation that you want to perform. Set the value to <code>DeleteDisk</code> .
DiskId	String	Yes	d-bp198f7xn673w2rw****	The ID of the disk that you want to release.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3****	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=DeleteDisk
&DiskId=d-bp198f7xn673w2rw****
&<Common request parameters>
```

Sample success responses

XML format

```
<DeleteDiskResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3****</RequestId>
</DeleteDiskResponse>
```

JSON format

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3****"
}
```

Error codes

HTTP status code	Error code	Error message	Description

HTTP status code	Error code	Error message	Description
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
403	DiskNotPortable	The specified disk is not a portable disk.	The error message returned because the specified disk is not removable.
403	DiskTypeViolation	The specified disk is a system disk and cannot support the operation.	The error message returned because the specified disk is a system disk and cannot be detached from the instance to which it is attached.
403	DiskCreatingSnapshot	The operation is denied due to a snapshot of the specified disk is not completed yet.	The error message returned because a snapshot of the specified disk is being created.
403	IncorrectDiskStatus.Initializing	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
404	InvalidDiskId.NotFound	The specified disk does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
403	DiskStillAttached	The disk is still attached to the disk.	The error message returned because the specified disk is still attached to an instance.
403	UserNotInTheWhiteList	The user is not in disk white list.	The error message returned because you are not authorized to manage the disk. Try again when you are authorized.
400	InvalidRegionId.MalFormed	The specified parameter RegionId is not valid.	The error message returned because the specified RegionId parameter is invalid.
403	InvalidDiskStatus.DeleteProtection	The specified disk is under delete protection, can not be deleted.	The error message returned because deletion protection is enabled for the specified disk and the disk cannot be released. Possible cause: The disk was detached from an instance to which it was attached as the system disk, and the detached system disk was retained as a data disk. The detached system disk cannot be released before the instance is released.

3.6.5. Snapshots

3.6.5.1. CreateSnapshot

Creates a snapshot for a disk.

Description

In the following scenarios, you cannot create snapshots for a disk:

- The number of manual snapshots (also called user-created snapshots) of the disk has reached 256.
- A snapshot is being created for the disk.
- The Elastic Compute Service (ECS) instance to which the disk is attached has never been started.
- The instance to which the disk is attached is not in the **Stopped** (`Stopped`) or **Running** (`Running`) state.
- When you query the instance information and the responses contain `{"OperationLocks": {"LockReason": "security"}}`, the instance is locked for security reasons and all operations cannot take effect on the instance.

When you call this operation, take note of the following items:

- About 1 hour after an instance is created or the system disk of an instance is replaced, you can create a snapshot for a disk that is attached to the instance. After a data disk is attached to an instance, you must wait a period of time before you can create a snapshot for the data disk. The amount of time to wait depends on the volume of data stored on the disk.
- If a snapshot is being created, you cannot use this snapshot to create a custom image by calling the `CreateImage` operation.
- If the disk for which you want to create a snapshot is attached to an instance, do not change the instance state during snapshot creation.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateSnapshot	The operation that you want to perform. Set the value to <code>CreateSnapshot</code> .
DiskId	String	Yes	d-bp1s5fnnk4gn2tws0****	The ID of the disk.

Parameter	Type	Required	Example	Description
SnapshotName	String	No	test-Snapshot****	The name of the snapshot. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, colons (:), underscores (_), and hyphens (-). It cannot start with auto because snapshots whose names start with auto are recognized as automatic snapshots.
Description	String	No	This is a description example.	The description of the snapshot. The description must be 2 to 256 characters in length and cannot start with http:// or https://. This parameter is empty by default.
RetentionDays	Integer	No	30	The retention period of the snapshot. Valid values: 1 to 65536. Unit: days. The snapshot is automatically released when the retention period expires. This parameter is empty by default, which indicates that the snapshot is not automatically released.
Category	String	No	Standard	The type of the snapshot. Valid values: - Standard: normal snapshot - Flash: local snapshot
ClientToken	String	No	123e4567-e89b-12d3-a456-42665544****	The client token that is used to ensure the idempotence of the request. You can use the client to generate a value, but you must make sure that it is unique among different requests. The ClientToken value can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence in ECS Developer Guide.
Tag.N.Key	String	No	TestKey	The key of tag N to be added to the snapshot. Valid values of N: 1 to 20. The tag key cannot be an empty string. It can be up to 128 characters in length and cannot start with acs: or aliyun. It cannot contain http:// or https://.
Tag.N.Value	String	No	TestValue	The value of tag N to be added to the snapshot. Valid values of N: 1 to 20. The tag value can be an empty string. It can be up to 128 characters in length and cannot start with acs: or contain http:// or https://.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.
SnapshotId	String	s-bp17441ohwka0yuh****	The ID of the snapshot.

Examples

Sample requests

```
https://ecs.example.com/?Action=CreateSnapshot
&DiskId=d-bp1s5fnvk4gn2tws0****
&SnapshotName=test-SnapshotName
&Description=This is a description example.
&ClientToken=123e4567-e89b-12d3-a456-42665544****
&Tag.1.Key=TestKey
&Tag.1.Value=TestValue
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateSnapshotResponse>
  <RequestId>C8B26B44-0189-443E-9816-D951F59623A9</RequestId>
  <SnapshotId>s-bp17441ohwka0yuh****</SnapshotId>
</CreateSnapshotResponse>
```

JSON format

```
{
  "RequestId": "C8B26B44-0189-443E-9816-D951F59623A9",
  "SnapshotId": "s-bp17441ohwka0yuh*****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidDiskId.NotFound	The specified DiskId does not exist.	The error message returned because the specified DiskId parameter does not exist. Check whether the disk ID is correct.
400	InvalidSnapshotName.Malformed	The specified SnapshotName is wrongly formed.	The error message returned because the specified SnapshotName parameter is invalid.
404	InvalidDescription.Malformed	The specified description is wrongly formed.	The error message returned because the specified Description parameter is invalid. The description must be 2 to 256 characters in length and cannot start with http:// or https://.
400	IncorrectInstanceState	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the instance is in the current state.
403	IncorrectDiskStatus.CreatingSnapshot	A previous snapshot creation is in process.	The error message returned because another snapshot is being created for the disk. Wait until the snapshot is created and try again.
403	InstanceLockedForSecurity	The disk attached instance is locked due to security.	The error message returned because the instance to which the disk is attached is locked for security reasons.
403	IncorrectDiskStatus.NeverAttached	The specified disk has never been attached to any instance.	The error message returned because the removable disk has never been attached to instances and its content remains unchanged.
403	QuotaExceed.Snapshot	The snapshot quota exceeds.	The error message returned because the maximum number of snapshots has been reached. To create new snapshots, delete snapshots that are no longer needed.
403	IncorrectDiskStatus.NeverUsed	The specified disk has never been used after creating.	The error message returned because the specified disk has never been used and its content remains unchanged.
403	CreateSnapshot.Failed	The process of creating snapshot is failed.	The error message returned because the snapshot fails to be created.
403	DiskInArrears	The specified operation is denied as your disk has expired.	The error message returned because the disk has expired due to an overdue payment.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
403	DiskId.ValueNotSupported	The specified parameter diskid is not supported.	The error message returned because the category of the specified Elastic Block Storage (EBS) device does not support this operation.
400	DiskCategory.OperationNotSupported	The operation is not supported to the specified disk due to its disk category	The error message returned because the specified disk category does not support this operation.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is available and you have no overdue payments for it.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient. Add funds to your account and try again.
403	InvalidAccountStatus.SnapshotServiceUnavailable	Snapshot service has not been opened yet.	The error message returned because the operation is not supported while the snapshot service is not activated.
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified InstanceId parameter does not exist. Check whether the instance ID is correct.
403	IncorrectVolumeStatus	The current volume status does not support this operation.	The error message returned because the operation is not supported while the Shared Block Storage device is in the current state.
404	InvalidVolumeId.NotFound	The specified volume does not exist.	The error message returned because the specified Shared Block Storage device does not exist. Check whether the Shared Block Storage device ID is correct.

HTTP status code	Error code	Error message	Description
403	IdempotentParameterMismatch	The specified clientToken is used.	The error message returned because the specified client token is already in use.
403	IncorrectDiskStatus.Invalid	The specified device status invalid, restart instance and try again.	The error message returned because the state of the specified disk is invalid. Restart the instance and try again.
403	IncorrectDiskType.NotSupport	The specified device type is not supported.	The error message returned because the specified disk type does not support the operation.
403	IncorrectDiskStatus.Transferring	The specified device is transferring, you can retry after the process is finished.	The error message returned because the specified disk is being migrated. Wait until the disk is migrated and try again.
403	InvalidParameter.KMSKeyId.CMKNotEnabled	The CMK needs to be enabled.	The error message returned because the customer master key (CMK) is not enabled when a Key Management Service (KMS) key ID is specified for a disk. You can call the DescribeKey operation of KMS to query the information of the specified CMK.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service have no right to access your KMS.	The error message returned because ECS is not authorized to access your KMS resources.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
400	Duplicate.TagKey	The Tag.N.Key contain duplicate key.	The error message returned because the specified tag key already exists. Tag keys must be unique.
400	InvalidTagKey.Malformed	The specified Tag.n.Key is not valid.	The error message returned because the specified Tag.N.Key parameter is invalid.
400	InvalidTagValue.Malformed	The specified Tag.n.Value is not valid.	The error message returned because the specified Tag.N.Value parameter is invalid.
403	IdempotentProcessing	The previous idempotent request(s) is still processing.	The error message returned because a previous idempotent request is being processed. Try again later.
400	InvalidRetentionDays.Malformed	The specified RetentionDays is not valid.	The error message returned because the specified RetentionDays parameter is invalid.
403	QuotaExceed.Tags	%s	The error message returned because the maximum number of tags has been reached.
403	InvalidSnapshotCategory.Malformed	The specified Category is not valid.	The error message returned because the specified Category parameter is invalid.

3.6.5.2. DeleteSnapshot

Deletes a snapshot. If you call this operation to delete a snapshot that is being created, the snapshot creation task is canceled.

Description

When you call this operation, take note of the following items:

- If the specified SnapshotId parameter does not exist, the request is ignored.
- A snapshot that has already been used to create custom images cannot be deleted. The snapshot can be deleted only after the created custom images are deleted (DeleteImage).

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteSnapshot	The operation that you want to perform. Set the value to DeleteSnapshot.
SnapshotId	String	Yes	s-bp1btgnlb4licd41****	The ID of the snapshot.
Force	Boolean	No	false	Specifies whether to forcibly delete the snapshot that has been used to create disks.  Note After a snapshot is deleted, the disks created from this snapshot cannot be re-initialized.

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0E3CB3E	The ID of the request. Every response returns a unique request ID regardless of whether the call is successful.

Examples

Sample requests

```
https://ecs.example.com/?Action=DeleteSnapshot
&SnapshotId=s-bp1btgnlb4licd41****
&<Common request parameters>
```

Sample success responses

XML format

```
<DeleteSnapshotResponse>
  <RequestId>CEF72CEB-54B6-4AE8-B225-F876FF7BA984</RequestId>
</DeleteSnapshotResponse>
```

JSON format

```
{
  "RequestId": "CEF72CEB-54B6-4AE8-B225-F876FF7BA984"
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	SnapshotCreatedImage	The snapshot has been used to create user defined image(s).	The error message returned because the specified snapshot has been used to create custom images and cannot be deleted. The snapshot can be deleted only after the custom images created from the snapshot are deleted (DeleteImage).
403	SnapshotCreatedDisk	The snapshot has been used to create disk(s).	The error message returned because the specified snapshot has been used to create disks and cannot be deleted when the Force parameter is not specified or is set to false. If you do want to delete the snapshot, you must set the Force parameter to true to force delete the snapshot. After the snapshot is force deleted, the disks created from the snapshot cannot be re-initialized.
400	MissingParameter	The input parameter SnapshotId that is mandatory for processing this request is not supplied.	The error message returned because the required SnapshotId parameter is not specified.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
404	InvalidSnapshotId.NotFound	The specified snapshot is not found	The error message returned because the specified SnapshotId parameter does not exist.
403	Operation.Conflict	The operation may conflicts with others, please retry later.	The error message returned because the operation conflicts with other operations. Try again later.

3.6.5.3. DescribeSnapshots

You can call this operation to query all snapshots of an ECS instance or a disk.

Description

You can specify multiple request parameters such as InstanceId, DiskId, and SnapshotIds to be queried. Specified parameters have logical AND relations. Only specified parameters are included in the filter conditions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeSnapshots	The operation that you want to perform. Set the value to DescribeSnapshots.
RegionId	String	Yes	cn-qingdao-env17-d01	The region ID of the instance or disk. You can call the DescribeRegions operation to query the most recent region list.
InstanceId	String	No	i-bp67acfmazb4p****	The ID of the instance.

Parameter	Type	Required	Example	Description
DiskId	String	No	d-bp67acfmazb4p****	The ID of the disk.
SnapshotLinkId	String	No	sp-bp67acfmazb4p****	The ID of the snapshot chain.
SnapshotIds	String	No	["s-bp67acfmazb4p****", "s-bp67acfmazb4p****", ... "s-bp67acfmazb4p****"]	The IDs of snapshots. The value can be a JSON array that consists of up to 100 snapshot IDs. Separate multiple snapshot IDs with commas (,).
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
SnapshotName	String	No	test-SnapshotName	The name of the snapshot.
Status	String	No	all	The status of the snapshot. Default value: all. Valid values: - progressing: The snapshot is being created. - accomplished: The snapshot is created. - failed: The snapshot failed to be created. - all: all snapshot states.
SnapshotType	String	No	all	The type of the snapshot. Default value: all. Valid values: - auto: automatic snapshot - user: manual snapshot - all: all snapshot types
Filter.1.Key	String	No	CreationStartTime	The key of filter condition 1 used to query resources. Set the value to CreationStartTime.
Filter.2.Key	String	No	CreationEndTime	The key of filter condition 2 used to query resources. Set the value to CreationEndTime.
Filter.1.Value	String	No	2019-12-13T17:00Z	The value of filter condition 1 used to query resources. The value must be the beginning of the time range in which to query created resources.
Filter.2.Value	String	No	2019-12-13T22:00Z	The value of filter condition 2 used to query resources. The value must be the end of the time range in which to query created resources.
Usage	String	No	none	Specifies whether the snapshot has been used to create images or disks. Valid values: - image: The snapshot has been used to create custom images. - disk: The snapshot has been used to create disks. - image_disk: The snapshot has been used to create custom images and data disks. - none: The snapshot has not been used to create custom images or disks.
SourceDiskType	String	No	Data	The category of the disk for which the snapshot was created. Valid values: - System: system disk - Data: data disk
Tag.N.Key	String	No	TestKey	The key of tag N of the snapshot. Valid values of N: 1 to 20.
Tag.N.Value	String	No	TestValue	The value of tag N of the snapshot. Valid values of N: 1 to 20.
Encrypted	Boolean	No	false	Specifies whether the snapshot is encrypted. Default value: false.
ResourceGroupId	String	No	rg-bp67acfmazb4p****	The ID of the resource group to which the snapshot belongs.

Parameter	Type	Required	Example	Description
DryRun	Boolean	No	false	Specifies whether to check the validity of the request without actually making the request. Default value: false. Valid value: - true: The validity of the request is checked but the request is not made. Check items include whether your AccessKey pair is valid, whether RAM users are authorized, and whether the required parameters are specified. If the check fails, the corresponding error is returned. If the check succeeds, the DryRunOperation error code is returned. - false: The validity of the request is checked. If the check succeeds, a 2XX HTTP status code is returned and the request is made.
KMSKeyId	String	No	0e478b7a-4262-4802-b8cb-00d3fb40****	The ID of the KMS key used by the data disk.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3CB3E	The ID of the request.
Snapshots	Array		The collections of details about snapshots.
Snapshot			
Category	String	standard	The type of the snapshot. Valid values: - standard: normal snapshot - flash: local snapshot
CreationTime	String	2019-12-24T13:00:52Z	The time when the snapshot was created. The time follows the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time is displayed in UTC.
Description	String	FinanceDept	The description of the snapshot.
Encrypted	Boolean	false	Indicates whether the snapshot was encrypted.
KMSKeyId	String	0e478b7a-4262-4802-b8cb-00d3fb40826X	The ID of the KMS key used by the data disk.
LastModifiedTime	String	test	 Note This parameter is not supported.
ProductCode	String	jxsc000***	 Note This parameter is not supported.
Progress	String	100	The progress of the snapshot creation task. Unit: percent (%).
RemainTime	Integer	38	The time remaining for the snapshot creation task. Unit: seconds.
ResourceGroupId	String	rg-bp67acfmazb4p****	The ID of the resource group to which the snapshot belongs.
RetentionDays	Integer	30	The retention period of the automatic snapshot. Unit: days.
SnapshotId	String	s-bp67acfmazb4p****	The ID of the snapshot.
SnapshotName	String	test-SnapshotName	The name of the snapshot. This parameter is returned only if a name was specified during creation of the snapshot.
SnapshotSN	String	64472-116742336-61976****	The serial number of the snapshot.

Parameter	Type	Example	Description
SnapshotType	String	all	The type of the snapshot. Default value: all. Valid values: - auto: automatic snapshot - user: manual snapshot - all: all snapshot types
SourceDiskId	String	d-bp67acfmazb4p****	The ID of the disk for which the snapshot was created. This parameter is retained even after the disk is released.
SourceDiskSize	String	2000	The size of the disk for which the snapshot was created. Unit: GiB.
SourceDiskType	String	Data	The type of the disk for which the snapshot was created. Valid values: - System - Data
SourceStorageType	String	disk	The category of the disk for which the snapshot was created.  Note This parameter will be removed in the future. We recommend that you use other parameters to ensure compatibility.
Status	String	accomplished	The status of the snapshot. Valid values: - progressing - accomplished - failed
Tags	Array		The tags of the snapshot.
Tag			
TagKey	String	TestKey	The tag key of the snapshot.
TagValue	String	TestValue	The tag value of the snapshot.
Usage	String	none	Indicates whether the snapshot had been used to create images or disks. Valid values: - image - disk - image_disk - none
TotalCount	Integer	36	The total number of snapshots.

Examples

Sample requests

```
https://ecs.example.com/?Action=DescribeSnapshots
&RegionId=cn-qingdao-env17-d01
&InstanceId=i-bp67acfmazb4p****
&DiskId=d-bp67acfmazb4p****
&SnapshotIds=["s-bp67acfmazb4p****", "s-bp67acfmazb4p****", ... "s-bp67acfmazb4p****"]
&PageNumber=1
&PageSize=10
&SnapshotName=test-SnapshotName
&Status=all
&SnapshotType=all
&Usage=none
&SourceDiskType=Data
&Tag.1.Key=TestKey
&Tag.1.Value=TestValue
&Encrypted=false
&DryRun=false
&<Common request parameters>
```

Sample success responses

XML [format](#)

```
<DescribeSnapshotsResponse>
  <PageNumber>1</PageNumber>
  <PageSize>2</PageSize>
  <RequestId>659F91C6-1949-43B0-90C4-B6342CA757D5</RequestId>
  <Snapshots>
    <Snapshot>
      <CreationTime>2015-07-30T05:00:14Z</CreationTime>
      <Progress>100%</Progress>
      <SnapshotId>s-bp67acfmazb4ph****</SnapshotId>
      <SnapshotName>auto_20150730_3</SnapshotName>
      <SourceDiskId>d-bp67acfmazb4p****</SourceDiskId>
      <SourceDiskSize>20</SourceDiskSize>
      <SnapshotType>user</SnapshotType>
      <SourceDiskType>system</SourceDiskType>
      <Status>accomplished</Status>
      <Usage>none</Usage>
    </Snapshot>
    <Snapshot>
      <CreationTime>2015-07-30T05:00:14Z</CreationTime>
      <Progress>100%</Progress>
      <SnapshotId>s-bp67acfmazb4pk****</SnapshotId>
      <SnapshotName>auto_20150730_3</SnapshotName>
      <SourceDiskId>d-bp67acfmazb4p****</SourceDiskId>
      <SourceDiskSize>20</SourceDiskSize>
      <SourceDiskType>system</SourceDiskType>
      <Status>accomplished</Status>
      <Usage>none</Usage>
    </Snapshot>
  </Snapshots>
  <TotalCount>36</TotalCount>
</DescribeSnapshotsResponse>
```

JSON **format**

```
{
  "PageNumber": 1,
  "PageSize": 2,
  "RequestId": "659F91C6-1949-43B0-90C4-B6342CA757D5",
  "Snapshots": [
    "Snapshot": [
      {
        "CreationTime": "2015-07-30T05:00:14Z",
        "Progress": "100%",
        "SnapshotId": "s-bp67acfmazb4ph****",
        "SnapshotName": "auto_20150730_3",
        "SourceDiskId": "d-bp67acfmazb4p****",
        "SourceDiskSize": 20,
        "SnapshotType": "user",
        "SourceDiskType": "system",
        "Status": "accomplished",
        "Usage": "none"
      },
      {
        "CreationTime": "2015-07-30T05:00:14Z",
        "Progress": "100%",
        "SnapshotId": "s-bp67acfmazb4pk****",
        "SnapshotName": "auto_20150730_3",
        "SourceDiskId": "d-bp67acfmazb4p****",
        "SourceDiskSize": 20,
        "SourceDiskType": "system",
        "Status": "accomplished",
        "Usage": "none"
      }
    ]
  },
  "TotalCount": 36
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	InvalidSnapshotIds.Malformed	The amount of specified specified snapshot ids exceeds the limit.	The error message returned because the specified SnapshotIds parameter is invalid.
404	InvalidFilterKey.NotFound		The error message returned because the specified start time or end time is invalid.
404	InvalidUsage	The specified Usage is not valid	The error message returned because the specified Usage parameter is invalid.
404	InvalidSourceDiskType	The specified SourceDiskType is not valid	The error message returned because the specified SourceDiskType parameter is invalid.

HTTP status code	Error code	Error message	Description
404	InvalidStatus.NotFound	The specified Status is not found	The error message returned because the specified Status parameter does not exist.
404	InvalidSnapshotType.NotFound	The specified SnapshotType is not found	The error message returned because the specified SnapshotType parameter does not exist.
400	InvalidTag.Mismatch	The specified Tag.N.Key and Tag.N.Value are not match.	The error message returned because the specified Tag.N.Key and Tag.N.Value parameters do not match.
400	InvalidTagCount	The specified tags are beyond the permitted range.	The error message returned because the number of specified tags exceeds the upper limit.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.
404	InvalidSnapshotLinkId.NotFound	The specified snapshot link is not found.	The error message returned because the specified snapshot chain does not exist.

3.6.5.4. DescribeSnapshotLinks

You can call this operation to query the snapshot chains of one or more disks. A snapshot chain is a chain of all the snapshots created for a disk. A disk corresponds to a chain of snapshots.

Description

When you call this operation, take note of the following items:

- You can specify multiple request parameters such as `RegionId`, `DiskIds`, and `InstanceId` to be queried. Specified parameters have logical AND relations.
- Only the specified parameters are used as filter conditions. If the `DiskIds` and `SnapshotLinkIds` parameters are set to empty JSON arrays, they are regarded as valid filter conditions and an empty result is returned.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeSnapshotLinks	The operation that you want to perform. Set the value to DescribeSnapshotLinks.
RegionId	String	Yes	cn-qingdao-env17-d01	The region ID of the disk. You can call the DescribeRegions operation to query the most recent region list.
InstanceId	String	No	i-bp1ahcvgu9z55pk7****	The ID of the instance.
DiskIds	String	No	["d-bp198f7xn673w2rw****", "d-bp12te2l9m0170wc****", ... "d-bp1btgnlb4lgdzo****"]	The IDs of disks. The value is a JSON array that consists of up to 100 disk IDs. Separate multiple disk IDs with commas (,).
SnapshotLinkIds	String	No	["sl-bp14r3b4xczuenhr****", "sl-bp1eftye2796vi2a****", ... "sl-bp1aixnfbjo290a8****"]	The IDs of snapshot chains. The value is a JSON array that consists of up to 100 snapshot chain IDs. Separate multiple snapshot chain IDs with commas (,).
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	No	50	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	50	The number of entries returned per page.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.
SnapshotLinks	Array of SnapshotLink		Details about the snapshot chains.
SnapshotLink			

Parameter	Type	Example	Description
Category	String	test	 Note This parameter is not supported.
InstanceId	String	i-bp1ahcvgu9z55pk7****	The ID of the instance.
InstanceName	String	test-InstanceName	The name of the instance.
RegionId	String	cn-qingdao-env17-d01	The ID of the region where the source disk of the snapshot chain is located.
SnapshotLinkId	String	sl-bp14r3b4xczuenhr****	The ID of the snapshot chain.
SourceDiskId	String	d-bp198f7xn673w2rw****	The ID of the source disk. This parameter is retained even if the source disk is deleted.
SourceDiskName	String	test-SourceDiskName	The name of the source disk.
SourceDiskSize	Integer	40	The size of the source disk. Unit: GiB.
SourceDiskType	String	cloud_ssd	The category of the source disk. Valid values: - cloud_efficiency: ultra disk - cloud_ssd: standard SSD - cloud_essd: enhanced SSD (ESSD) - ephemeral_ssd: local SSD - cloud: basic disk
TotalCount	Integer	1	The total number of snapshots.
TotalSize	Long	2097152	The total size of all snapshots in the snapshot chain. Unit: byte.
TotalCount	Integer	9	The total number of snapshot chains.

Examples

Sample requests

```
https://ecs.example.com/?Action=DescribeSnapshotLinks
&RegionId=cn-qingdao-env17-d01
&InstanceId=i-bp1ahcvgu9z55pk7****
&DiskIds=["d-bp198f7xn673w2rw****", "d-bp12te2l9m0170wc****", ... "d-bp1btgnlb4lgdzoi****"]
&SnapshotLinkIds=["sl-bp14r3b4xczuenhr****", "sl-bp1eftye2796vi2a****", ... "sl-bplaixnfbjo290a8****"]
&PageNumber=1
&PageSize=10
&<Common request parameters>
```

Sample success responses

XML

format

```
<DescribeSnapshotLinksResponse>
  <SnapshotLinks>
    <SnapshotLink>
      <SourceDiskSize>20</SourceDiskSize>
      <TotalSize>2097152</TotalSize>
      <SourceDiskType>data</SourceDiskType>
      <RegionId>cn-qingdao-env17-d01</RegionId>
      <SnapshotLinkId>s1-bp14r3b4xczuenhr****</SnapshotLinkId>
      <SourceDiskId>d-bp198f7xn673w2rw****</SourceDiskId>
      <TotalCount>1</TotalCount>
    </SnapshotLink>
    <SnapshotLink>
      <SourceDiskSize>40</SourceDiskSize>
      <TotalSize>2097152</TotalSize>
      <SourceDiskType>data</SourceDiskType>
      <RegionId>cn-qingdao-env17-d01</RegionId>
      <SnapshotLinkId>s1-bp14r3b4xczuenhr****</SnapshotLinkId>
      <SourceDiskId>d-bp198f7xn673w2rw****</SourceDiskId>
      <TotalCount>1</TotalCount>
    </SnapshotLink>
  </SnapshotLinks>
  <PageNumber>1</PageNumber>
  <PageSize>2</PageSize>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E</RequestId>
  <TotalCount>9</TotalCount>
</DescribeSnapshotLinksResponse>
```

JSON **format**

```
{
  "SnapshotLinks": {
    "SnapshotLink": [
      {
        "SourceDiskSize": 20,
        "TotalSize": 2097152,
        "SourceDiskType": "data",
        "RegionId": "cn-qingdao-env17-d01",
        "SnapshotLinkId": "s1-bp14r3b4xczuenhr****",
        "SourceDiskId": "d-bp198f7xn673w2rw****",
        "TotalCount": 1
      },
      {
        "SourceDiskSize": 40,
        "TotalSize": 2097152,
        "SourceDiskType": "data",
        "RegionId": "cn-qingdao-env17-d01",
        "SnapshotLinkId": "s1-bp14r3b4xczuenhr****",
        "SourceDiskId": "d-bp198f7xn673w2rw****",
        "TotalCount": 1
      }
    ]
  },
  "PageNumber": 1,
  "PageSize": 2,
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E",
  "TotalCount": 9
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidRegionId	The specified regionId is invalid.	The error message returned because the specified RegionId parameter is invalid.
400	InvalidSnapshotLinkIds	The specified snapshotLinkIds is invalid.	The error message returned because the specified SnapshotLinkIds parameter is invalid.
400	InvalidDiskIds	The specified diskIds is invalid.	The error message returned because the specified DiskIds parameter is invalid.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the problem persists, submit a ticket.

3.6.5.5. DescribeSnapshotsUsage

You can call this operation to query the number of snapshots stored in a region and the total size of the snapshots.

If you want to view the snapshot usage information on each disk in the current region, we recommend that you call the DescribeSnapshotLinks operation to query snapshot chain information.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeSnapshotsUsage	The operation that you want to perform. Set the value to DescribeSnapshotsUsage.
RegionId	String	Yes	cn-qingdao-env17-d01	The region ID of the snapshot. You can call the DescribeRegions operation to query the most recent region list.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.
SnapshotCount	Integer	5	The number of snapshots stored in the current region.
SnapshotSize	Long	122	The total size of snapshots stored in the current region. Unit: byte.

Examples

Sample requests

```
https://ecs.example.com/?Action=DescribeSnapshotsUsage
&RegionId=cn-qingdao-env17-d01
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeSnapshotsUsageResponse>
  <SnapshotCount>5</SnapshotCount>
  <SnapshotSize>122</SnapshotSize>
  <RequestId>ED5CF6DD-71CA-462C-9C94-A61A78A01479</RequestId>
</DescribeSnapshotsUsageResponse>
```

JSON format

```
{
  "SnapshotCount": "5",
  "SnapshotSize": "122",
  "RequestId": "ACD9BBB0-A9D1-46D7-9630-B7A69889E110"
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	InvalidParam.RegionId	The specified region is not exist.	The error message returned because the specified RegionId parameter does not exist.

3.6.5.6. CreateAutoSnapshotPolicy

You can call this operation to create an automatic snapshot policy.

Description

After an automatic snapshot policy is created, you can call the ApplyAutoSnapshotPolicy operation to apply the policy to disks. You can also call the ModifyAutoSnapshotPolicyEx operation to modify the automatic snapshot policy.

When you call this operation, take note of the following items:

- A maximum of 100 automatic snapshot policies can be created in each region per Apsara Stack tenant account.
- If an automatic snapshot is being created when the time scheduled for a new automatic snapshot is due, the new snapshot task is skipped. This may occur when a disk contains a large volume of data. For example, an automatic snapshot policy is applied to a disk that contains a large volume of data, and the policy specifies that snapshots are created at 09:00, 10:00, 11:00, and 12:00. The system starts to create a snapshot for the disk at 09:00. The snapshot creation process takes 80 minutes and ends at 10:20. The system skips the automatic snapshot task scheduled for 10:00 and creates the next automatic snapshot for the disk at 11:00.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateAutoSnapshotPolicy	The operation that you want to perform. Set the value to CreateAutoSnapshotPolicy.

Parameter	Type	Required	Example	Description
regionId	String	Yes	cn-qingdao-env17-d01	The ID of the region in which to create an automatic snapshot policy. You can call the DescribeRegions operation to query the most recent region list.
repeatWeekdays	String	Yes	["1","2"]	The days of the week on which to create automatic snapshots. Valid values: 1 to 7, which correspond to Monday to Sunday. 1 indicates Monday. To schedule multiple automatic snapshots to be created for a disk in a week, you can set this parameter to a JSON array. - You can specify up to seven values in the array. - Separate the values in the array with commas (.). Example: <code>["1","2"...,"7"]</code>.
retentionDays	Integer	Yes	30	The number of days for which to retain automatic snapshots. Valid values: -1: Automatic snapshots are permanently retained. 1 to 65535: Automatic snapshots are retained for a specified number of days. Default value: -1.
timePoints	String	Yes	["0","1",... "23"]	The point in time at which to create automatic snapshots. Unit: hours. Valid values: 0 to 23. These values correspond to the hours of the day from 00:00 to 23:00. 1 indicates 01:00. To schedule multiple automatic snapshots to be created for a disk in a day, you can set this parameter to a JSON array that consists of multiple values. You can specify up to 24 values in the array.
autoSnapshotPolicyName	String	No	test-autoSnapshotPolicyName	The name of the automatic snapshot policy. The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). The name must start with a letter and cannot start with http:// or https://. This parameter is empty by default.

Response parameters

Parameter	Type	Example	Description
AutoSnapshotPolicyId	String	sp-bp12m37ccmxvbm5****	The ID of the automatic snapshot policy.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=CreateAutoSnapshotPolicy
&regionId=cn-qingdao-env17-d01
&repeatWeekdays=["1"]
&retentionDays=30
&timePoints=["0","19"]
&autoSnapshotPolicyName=test-autoSnapshotPolicyName
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateAutoSnapshotPolicyResponse>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
  <AutoSnapshotPolicyId>sp-bp12m37ccmxvbm5****</AutoSnapshotPolicyId>
</CreateAutoSnapshotPolicyResponse>
```

JSON format

```
{
  "RequestId": "F3CD6886-D8D0-4FEE-B93E-1B73239673DE",
  "AutoSnapshotPolicyId": "sp-bp12m37ccmxvbm5****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	ParameterInvalid	The specified RegionId parameter is invalid.	The error message returned because the specified regionId parameter is invalid.
403	ParameterInvalid	The specified RepeatWeekDays parameter is invalid.	The error message returned because the specified repeatWeekDays parameter is invalid.
403	ParameterInvalid	The specified TimePoints parameter is invalid.	The error message returned because the specified timePoints parameter is invalid.
403	ParameterInvalid	The specified RetentionDays parameter is invalid.	The error message returned because the specified retentionDays parameter is invalid.
403	AutoSnapshotPolicy.QuotaExceed	The maximum number of automatic snapshot policy has been reached.	The error message returned because the maximum number of automatic snapshot policies has been reached.
400	DiskCategory.OperationNotSupported	The type of the specified disk does not support creating a snapshot.	The error message returned because the operation is not supported by the current disk category.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient. Add funds to your account and try again.
403	InvalidAccountStatus.SnapshotServiceUnavailable	Snapshot service has not been opened yet.	The error message returned because the operation is not supported while the snapshot service is not activated.

3.6.5.7. DeleteAutoSnapshotPolicy

You can call this operation to delete an automatic snapshot policy. After an automatic snapshot policy is deleted, the policy will no longer be applied to the disks that it previously took effect on.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteAutoSnapshotPolicy	The operation that you want to perform. Set the value to DeleteAutoSnapshotPolicy.
autoSnapshotPolicyId	String	Yes	sp-bp12m37ccmxvbm5****	The ID of the automatic snapshot policy. You can call the DescribeAutoSnapshotPolicies operation to query available automatic snapshot policies.
regionId	String	Yes	cn-qingdao-env17-d01	The region ID of the automatic snapshot policy. You can call the DescribeRegions operation to query the most recent region list.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request. Every response returns a unique request ID regardless of whether the call is successful.

Examples

Sample requests

```
https://ecs.example.com/?Action=DeleteAutoSnapshotPolicy
&autoSnapshotPolicyId=sp-bp12m37ccmxvbm5****
&regionId=cn-qingdao-env17-d01
&<Common request parameters>
```

Sample success responses

XML format

```
<DeleteAutoSnapshotPolicyResponse>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
</DeleteAutoSnapshotPolicyResponse>
```

JSON format

```
{
  "RequestId": "F3CD6886-D8D0-4FEE-B93E-1B73239673DE"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	ParameterInvalid	The specified automatic snapshot policy does not exist.	The error message returned because the specified autoSnapshotPolicyID parameter does not exist.
404	ParameterInvalid	The specified automatic snapshot policy does not exist in the region.	The error message returned because the specified automatic snapshot policy does not exist in the current region.

3.6.5.8. ModifyAutoSnapshotPolicyEx

You can call this operation to modify an automatic snapshot policy. After an automatic snapshot policy is modified, the modifications take effect immediately on the disks to which the policy is applied.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyAutoSnapshotPolicyEx	The operation that you want to perform. Set the value to ModifyAutoSnapshotPolicyEx.
autoSnapshotPolicyId	String	Yes	sp-bp12m37ccmxvbm5****	The ID of the automatic snapshot policy. You can call the DescribeAutoSnapshotPolicies operation to query available automatic snapshot policies.
regionId	String	Yes	cn-qingdao-env17-d01	The region ID of the automatic snapshot policy. You can call the DescribeRegions operation to query the most recent region list.
autoSnapshotPolicyName	String	No	test-autoSnapshotPolicyName	The name of the automatic snapshot policy. If this parameter is not specified, the original name is retained.
timePoints	String	No	["0", "1"]	The point in time at which to create automatic snapshots. Unit: hours. Valid values: 0 to 23. These values correspond to the hours of the day from 00:00 to 23:00. 1 indicates 01:00. To schedule multiple automatic snapshots to be created for a disk in a day, you can set timePoints to a JSON array that consists of multiple values. - You can specify up to 24 values in the array. - Separate the values in array with commas (.). Example: ["0", "1", ... "23"].
repeatWeekdays	String	No	["1", "7"]	The days of the week on which to create automatic snapshots. Valid values: 1 to 7, which correspond to Monday to Sunday. 1 indicates Monday. To schedule multiple automatic snapshots to be created for a disk in a week, you can set repeatWeekdays to a JSON array that consists of multiple values. - You can specify up to seven values in the array. - Separate the values in the array with commas (.). Example: ["1", "2", ... "7"].
retentionDays	Integer	No	30	The number of days for which to retain automatic snapshots. Valid values: -1: Automatic snapshots are permanently retained. 1 to 65536: Automatic snapshots are retained for a specified number of days.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0E3CB3E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ModifyAutoSnapshotPolicyEx
&autoSnapshotPolicyId=sp-bp12m37ccmxvbm5****
&regionId=cn-qingdao-env17-d01
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifyAutoSnapshotPolicyResponse>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
</ModifyAutoSnapshotPolicyResponse>
```

JSON **format**

```
{
  "RequestId": "F3CD6886-D8D0-4FEE-B93E-1B73239673DE"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	ParameterInvalid	The specified automatic snapshot policy does not exist.	The error message returned because the specified automatic snapshot policy does not exist.
404	ParameterInvalid	The specified automatic snapshot policy does not exist in the region.	The error message returned because the specified automatic snapshot policy does not exist in the current region.
403	ParameterInvalid	The specified RegionId parameter is invalid.	The error message returned because the specified regionId parameter is invalid.
403	ParameterInvalid	The specified RepeatWeekDays parameter is invalid.	The error message returned because the specified repeatWeekDays parameter is invalid.
403	ParameterInvalid	The specified TimePoints parameter is invalid.	The error message returned because the specified timePoints parameter is invalid.
403	ParameterInvalid	The specified AutoSnapshotPolicyId is invalid.	The error message returned because the specified autoSnapshotPolicyId parameter is invalid.
403	ParameterInvalid	The specified RetentionDays parameter is invalid.	The error message returned because the specified retentionDays parameter is invalid.

3.6.5.9. ModifySnapshotAttribute

You can call this operation to modify the name or description of a snapshot.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifySnapshotAttribute	The operation that you want to perform. If you use a custom HTTP URL or HTTPS URL to make an API request, you must specify the <code>Action</code> parameter. Set this parameter to <code>ModifySnapshotAttribute</code> .
SnapshotId	String	Yes	s-923FE2B***	The ID of the snapshot.
SnapshotName	String	No	NewSnapshotName-EcsGuideTest	The new name of the snapshot. The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). The name must start with a letter and cannot start with <code>http://</code> or <code>https://</code> . It also cannot start with <code>auto</code> because snapshot names that start with <code>auto</code> are recognized as automatic snapshots.
Description	String	No	NewSnapshotDescription-EcsGuideTest	The new description of the snapshot. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ModifySnapshotAttribute
&SnapshotName=NewSnapshotName-EcsGuideTest
&Description=NewSnapshotDescription-EcsGuideTest
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifySnapshotAttributeResponse>
  <RequestId>C8B26B44-0189-443E-9816-D951F59623A9</RequestId>
</ModifySnapshotAttributeResponse>
```

JSON format

```
{
  "RequestId": "C8B26B44-0189-443E-9816-D951F59623A9"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidDescription.Malformed	The specified description is wrongly formed.	The error message returned because the specified Description parameter is invalid. The description must be 2 to 256 characters in length and cannot start with http:// or https://.
400	InvalidSnapshotName.Malformed	The specified SnapshotName is wrongly formed.	The error message returned because the specified SnapshotName parameter is invalid.
400	NoAttributeToModify	No attribute to be modified in this request.	The error message returned because no attributes of the specified snapshot are modified.
404	InvalidSnapshotId.NotFound	The specified SnapshotId does not exist.	The error message returned because the specified SnapshotId parameter does not exist.

3.6.5.10. DescribeAutoSnapshotPolicyEX

You can call this operation to query the automatic snapshot policies that you have created.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeAutoSnapshotPolicyEx	The operation that you want to perform. Set the value to DescribeAutoSnapshotPolicyEx.
RegionId	String	Yes	cn-qingdao-env17-d01	The region ID of the automatic snapshot policy. You can call the DescribeRegions operation to query the most recent region list.
AutoSnapshotPolicyId	String	No	sp-bp67acfmxazb4p****	The ID of the automatic snapshot policy.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Default value: 1
PageSize	Integer	No	1	The number of entries to return on each page. Valid values: 1 to 100 Default value: 10
Tag.N.Key	String	No	TestKey	The key of tag N of the automatic snapshot policy. Valid values of N: 1 to 20. The tag key cannot be an empty string. It can be up to 128 characters in length and cannot contain http:// or https://. It cannot start with acs: or aliyun.
Tag.N.Value	String	No	TestValue	The value of tag N of the automatic snapshot policy. Valid values of N: 1 to 20. The tag value can be an empty string. It can be up to 128 characters in length. It cannot start with acs: or contain http:// or https://.

Response parameters

Parameter	Type	Example	Description
AutoSnapshotPolicies	Array		The collection of details about automatic snapshot policies.
AutoSnapshotPolicy			
AutoSnapshotPolicyId	String	sp-bp67acfmxazb4p****	The ID of the automatic snapshot policy.

Parameter	Type	Example	Description
AutoSnapshotPolicyName	String	test-AutoSnapshotPolicyName	The name of the automatic snapshot policy.
CopiedSnapshotsRetentionDays	Integer	0	 Note This parameter is not supported.
CreationTime	String	2014-04-21T12:08:52Z	The time when the automatic snapshot policy was created. The time follows the ISO 8601 standard in the yyyy-MM-ddThh:mm:ssZ format. The time is displayed in UTC.
DiskNums	Integer	1	The number of disks to which the automatic snapshot policy was applied.
EnableCrossRegionCopy	Boolean	false	 Note This parameter is not supported.
RegionId	String	cn-qingdao-env17-d01	The region ID of the automatic snapshot policy.
RepeatWeekdays	String	["6"]	The days of the week on which to create automatic snapshots. Valid values: 1 to 7, which correspond to the days of the week. 1 indicates Monday. One or more days can be specified.
RetentionDays	Integer	30	The number of days for which to retain an automatic snapshot. Valid values: -1: The automatic snapshot is retained permanently. 1 to 65536: The automatic snapshot is retained for the specified number of days.
Status	String	Normal	The status of the automatic snapshot policy. Valid values: - Normal - Expire
Tags	Array		The tags of the automatic snapshot policy.
Tag			
TagKey	String	TestKey	The tag key of the automatic snapshot policy.
TagValue	String	TestValue	The tag value of the automatic snapshot policy.
TargetCopyRegions	String	test	 Note This parameter is not supported.
TimePoints	String	["4", "19"]	The points in time at which to create automatic snapshots. The least interval at which snapshots can be created is one hour. Valid values: 0 to 23, which correspond to the hours of the day from 00:00 to 23:00. 1 indicates 01:00. The value is a JSON array that consists of multiple points in time. Example: ["0", "1", ... "23"]. You can specify a maximum of 24 points in time. Separate the points in time with commas (,).
VolumeNums	Integer	2	The number of extended volumes to which this policy is applied.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	1	The number of entries returned per page.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	The ID of the request.
TotalCount	Integer	2	The total number of automatic snapshot policies.

Examples

Sample requests

```
https://ecs.example.com/?Action=DescribeAutoSnapshotPolicyEx
&RegionId=cn-qingdao-env17-d01
&AutoSnapshotPolicyId=sp-bp67acfmazb4p****
&PageNumber=1
&PageSize=1
&<Common request parameters>
```

Sample success responses

XML **format**

```
<DescribeAutoSnapshotPolicyResponse>
  <PageNumber>1</PageNumber>
  <TotalCount>2</TotalCount>
  <AutoSnapshotPolicies>
    <AutoSnapshotPolicy>
      <CreationTime>2019-01-22 11:38:08</CreationTime>
      <Status>Normal</Status>
      <DiskNums>1</DiskNums>
      <AutoSnapshotPolicyName>test-AutoSnapshotPolicyName</AutoSnapshotPolicyName>
      <RegionId>cn-qingdao-env17-d01</RegionId>
      <VolumeNums>2</VolumeNums>
      <RetentionDays>30</RetentionDays>
      <TimePoints>["0"]</TimePoints>
      <AutoSnapshotPolicyId>sp-bp67acfmazb4p****</AutoSnapshotPolicyId>
      <RepeatWeekdays>["7"]</RepeatWeekdays>
      <Tags>
        <Tag>
          <TagKey>Test</TagKey>
          <TagValue>Test</TagValue>
        </Tag>
      </Tags>
    </AutoSnapshotPolicy>
  </AutoSnapshotPolicies>
  <PageSize>1</PageSize>
  <RequestId>CC323EDF-BBB7-420F-BCD1-D9B9D2E9D424</RequestId>
</DescribeAutoSnapshotPolicyResponse>
```

JSON **format**

```
{
  "PageNumber": 1,
  "TotalCount": 2,
  "AutoSnapshotPolicies": {
    "AutoSnapshotPolicy": [
      {
        "CreationTime": "2019-01-22 11:38:08",
        "Status": "Normal",
        "DiskNums": 1,
        "AutoSnapshotPolicyName": "test-AutoSnapshotPolicyName",
        "RegionId": "cn-qingdao-env17-d01",
        "VolumeNums": 2,
        "RetentionDays": 30,
        "TimePoints": "[\"0\"]",
        "AutoSnapshotPolicyId": "sp-bp67acfmazb4p****",
        "RepeatWeekdays": "[\"7\"]",
        "Tags": {
          "Tag": [
            {
              "TagKey": "Test",
              "TagValue": "Test"
            }
          ]
        }
      }
    ]
  },
  "PageSize": 1,
  "RequestId": "CC323EDF-BBB7-420F-BCD1-D9B9D2E9D424"
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified RegionId parameter does not exist.
400	InvalidTag.Mismatch	The specified Tag.n.Key and Tag.n.Value are not match.	The error message returned because the specified Tag.N.Key and Tag.N.Value parameters do not match.
400	InvalidTagCount	The specified tags are beyond the permitted range.	The error message returned because the number of specified tags exceeds the upper limit.

3.6.5.11. ApplyAutoSnapshotPolicy

Applies an automatic snapshot policy to one or more disks. If you apply an automatic snapshot policy to a disk that already has an automatic snapshot policy, the new policy replaces the original policy to take effect on the disk.

Description

- Each disk can have only one automatic snapshot policy applied.
- A single automatic snapshot policy can be applied to multiple disks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ApplyAutoSnapshotPolicy	The operation that you want to perform. Set the value to ApplyAutoSnapshotPolicy.
regionId	String	Yes	cn-qingdao-****-d01	The region ID of the automatic snapshot policy and the disks. You can call the DescribeRegions operation to query the most recent region list.
autoSnapshotPolicyId	String	Yes	sp-bp12m37ccmxvbm5****	The ID of the automatic snapshot policy to apply.
diskIds	String	Yes	["d-bp198f7xn673w2rw****", "d-bp12te2l9m0170wc****", "d-bp1btgnlb4lgdzo****"]	The IDs of one or more disks. The value is a JSON array that consists of disk IDs. Separate multiple disk IDs with commas (,).

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3****	The ID of the request.

Examples

Sample requests

```
https://ecs.example.com/?Action=ApplyAutoSnapshotPolicy
&autoSnapshotPolicyId=sp-bp12m37ccmxvbm5****
&diskIds=["d-bp198f7xn673w2rw****", "d-bp12te2l9m0170wc****", "d-bp1btgnlb4lgdzo****"]
&regionId=cn-qingdao-****-d01
&<Common request parameters>
```

Sample success responses

XML format

```
<ApplyAutoSnapshotPolicyResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3****</RequestId>
</ApplyAutoSnapshotPolicyResponse>
```

JSON format

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	ParameterInvalid	The specified automatic snapshot policy does not exist.	The error message returned because the specified autoSnapshotPolicyId parameter does not exist.
404	ParameterInvalid	The specified automatic snapshot policy does not exist in the region.	The error message returned because the specified autoSnapshotPolicyId parameter does not exist.
404	InvalidDiskId.NotFound	The specified disk does not exist in the region.	The error message returned because the specified disk does not exist in the current region.
403	ParameterInvalid	The specified RegionId parameter is invalid.	The error message returned because the specified regionId parameter is invalid.
403	ParameterInvalid	The specified AutoSnapshotPolicyId parameter is invalid.	The error message returned because the specified autoSnapshotPolicyId parameter is invalid.
403	ParameterInvalid	The specified DiskIds are invalid.	The error message returned because the specified diskIds parameter is invalid.

HTTP status code	Error code	Error message	Description
400	DiskCategory.OperationNotSupported	The type of the specified disk does not support creating a snapshot.	The error message returned because the operation is not supported by the current disk category.
400	PartOfDiskCategory.OperationNotSupported	The types of some disks in the disk list do not support creating snapshots.	The error message returned because some of the specified disks do not support this operation.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient. Add funds to your account and try again.
403	InvalidAccountStatus.SnapshotServiceUnavailable	Snapshot service has not been opened yet.	The error message returned because the operation is not supported while the snapshot service is not activated.

3.6.5.12. CancelAutoSnapshotPolicy

You can call this operation to disable the automatic snapshot policy for one or more disks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CancelAutoSnapshotPolicy	The operation that you want to perform. Set the value to CancelAutoSnapshotPolicy.
diskIds	String	Yes	["d-bp198f7xn673w2rw****", "d-bp12te2l9m0170wc****", "d-bp1btgnlb4lgdzo****"]	The IDs of disks. To disable an automatic snapshot policy for multiple disks, you can set this parameter to a JSON array that consists of multiple disk IDs in the format of ["dxxxxxxx", "dyyyyyyyy", ... "dzzzzzzz"]. Separate multiple disk IDs with commas (,).
regionId	String	Yes	cn-qingdao-env17-d01	The ID of the region to which the automatic snapshot policy and the disks belong. You can call the DescribeRegions operation to query the most recent region list.

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0E3CB3E	The ID of the request. Every response returns a unique request ID regardless of whether the call is successful.

Examples

Sample requests

```
https://ecs.example.com/?Action=CancelAutoSnapshotPolicy
&diskIds=["d-bp198f7xn673w2rw****", "d-bp12te2l9m0170wc****", "d-bp1btgnlb4lgdzo****"]
&regionId=cn-qingdao-env17-d01
&<Common request parameters>
```

Sample success responses

XML format

```
<CancelAutoSnapshotPolicyResponse>
  <RequestId>F3CD6886-D8D0-4FEE-B93E-1B73239673DE</RequestId>
</CancelAutoSnapshotPolicyResponse>
```

JSON format

```
{
  "RequestId": "F3CD6886-D8D0-4FEE-B93E-1B73239673DE"
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	ParameterInvalid	The specified RegionId parameter is invalid.	The error message returned because the specified regionId parameter is invalid.
403	ParameterInvalid	The specified DiskIds are invalid.	The error message returned because the specified diskIds parameter is invalid.
404	InvalidDiskId.NotFound	The specified DiskIds are not found.	The error message returned because the specified diskIds parameter does not exist.

3.6.5.13. CreateSnapshotGroup

Creates a snapshot-consistent group for the disks in an Elastic Compute Service (ECS) instance. A snapshot-consistent group can contain snapshots of one or more disks.

Description

When you call this operation, take note of the following items:

- The disk for which you want to create a snapshot must be in the **Running** (`Running`) or **Unattached** (`Available`) state.
 - If the disk is in the **Running** (`Running`) state, the instance to which the disk is attached must be in the **Running** (`Running`) or **Stopped** (`Stopped`) state.
 - If the disk is in the **Unattached** (`Available`) state, make sure that the disk has been attached to an ECS instance. Snapshots cannot be created for disks that have never been attached to ECS instances.
- The snapshot-consistent group feature can be used to create snapshots only for premium performance disks and standard performance disks.
- A single snapshot-consistent group can contain snapshots of up to 16 disks including system disks and data disks and cannot exceed 32 TiB in size.
- Snapshots that you created are retained until they are deleted. We recommend that you delete unnecessary snapshots on a regular basis to prevent the number of snapshots from exceeding the quota.
- Snapshot-consistent groups cannot be created for disks that have the multi-attach feature enabled. If disks that have the multi-attach feature enabled are attached to an instance, you must set the `ExcludeDiskId.N` parameter to exclude these disks.

For more information about the snapshot-consistent group feature, see *Creating a snapshot-consistent group* in *ECS User Guide*.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateSnapshotGroup	The operation that you want to perform. Set the value to CreateSnapshotGroup.
InstanceId	String	Yes	i-q1a01ac6ae3gcb11****	The ID of the instance.
RegionId	String	Yes	cn-qingdao-****	The region ID of the instance. You can call the DescribeRegions operation to query the most recent region list. For more information, see <i>DescribeRegions</i> in <i>ECS Developer Guide</i> .
ExcludeDiskId.N	RepeatList	No	d-q1a0113yyonhqbg****	The ID of disk N for which you do not need to create snapshots. After this parameter is specified, the created snapshot-consistent group does not contain snapshots of the disk. Valid values of N: 1 to 16. This parameter is empty by default, which indicates that snapshots are created for all disks in the instance.
InstantAccess	Boolean	No	false	Specifies whether to enable the instant access feature. Valid values: - true: enables the instant access feature. - false: disables the instant access feature. Default value: false.
InstantAccessRetentionDays	Integer	No	1	The number of days for which the instant access feature is available. Unit: days. Valid values: 1 to 65535. This parameter takes effect only when <code>InstantAccess</code> is set to true. The instant access feature is automatically disabled when the specified duration of instant access expires. This parameter is empty by default, which indicates that the expiration time of the instant access feature is determined by the time when snapshots are released.
Name	String	No	testName	The name of the snapshot-consistent group. The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). It must start with a letter or a digit and cannot start with <code>http://</code> or <code>https://</code> .
Description	String	No	This is description.	The description of the snapshot-consistent group. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
RequestId	String	ssg-j6ciyh3k52qp7ovm****	The ID of the snapshot-consistent group.
SnapshotGroupId	String	01ABBD93-1ABB-4D92-B496-1A3D20EC****	The ID of the request.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=CreateSnapshotGroup
&InstanceId=i-gla0lac6ae3gcb11****
&RegionId=cn-qingdao-****
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateSnapshotGroupResponse>
  <RequestId>ssg-j6ciyh3k52qp7ovm**** </RequestId>
  <SnapshotGroupId>01ABBD93-1ABB-4D92-B496-1A3D20EC****</SnapshotGroupId>
</CreateSnapshotGroupResponse>
```

JSON format

```
{
  "RequestId": "ssg-j6ciyh3k52qp7ovm****",
  "SnapshotGroupId": "01ABBD93-1ABB-4D92-B496-1A3D20EC****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidDiskId.NotFound	The specified DiskId does not exist.	The error message returned because the specified disk does not exist. Check whether the disk ID is correct.
400	InvalidSnapshotName.Malformed	The specified SnapshotName is wrongly formed.	The error message returned because the specified Name parameter is invalid.
404	InvalidDescription.Malformed	The specified description is wrongly formed.	The error message returned because the specified Description parameter is invalid. The description must be 2 to 256 characters in length and cannot start with http:// or https://.
400	IncorrectInstanceStatus	The current status of the resource does not support this operation.	The error message returned because the operation is not supported while the resource is in the current state.
403	IncorrectDiskStatus.CreatingSnapshot	A previous snapshot creation is in process.	The error message returned because another snapshot is being created for the disk. Wait until the snapshot is created and try again.
403	InstanceLockedForSecurity	The disk attached instance is locked due to security.	The error message returned because the instance to which the disk is attached is locked for security reasons.
403	IncorrectDiskStatus.NeverAttached	The specified disk has never been attached to any instance.	The error message returned because the removable disk has never been attached to instances and its data remains unchanged.
403	QuotaExceed.Snapshot	The snapshot quota exceeds.	The error message returned because the maximum number of snapshots has been reached. To create new snapshots, delete snapshots that are no longer needed.
403	IncorrectDiskStatus.NeverUsed	The specified disk has never been used after creating.	The error message returned because the specified disk has never been used and its data remains unchanged.
403	CreateSnapshot.Failed	The process of creating snapshot is failed.	The error message returned because the snapshot cannot be created.
403	DiskInArrears	The specified operation is denied as your disk has expired.	The error message returned because the disk has expired due to an overdue payment.
500	InternalError	The request processing has failed due to some unknown error.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
403	DiskId.ValueNotSupported	The specified parameter diskid is not supported.	The error message returned because the category of the specified Elastic Block Storage (EBS) device does not support this operation.

HTTP status code	Error code	Error message	Description
400	DiskCategory.OperationNotSupported	The operation is not supported to the specified disk due to its disk category	The error message returned because the specified disk category does not support this operation.
403	IncorrectDiskStatus	The current disk status does not support this operation.	The error message returned because the operation is not supported while the disk is in the current state. Make sure that the disk is usable and you have no overdue payments for it.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient. Add funds to your account and try again.
403	InvalidAccountStatus.SnapshotServiceUnavailable	Snapshot service has not been opened yet.	The error message returned because the operation is not supported while the snapshot service is not activated.
404	InvalidInstanceId.NotFound	The specified InstanceId does not exist.	The error message returned because the specified instance does not exist. Check whether the instance ID is correct.
403	IncorrectVolumeStatus	The current volume status does not support this operation.	The error message returned because the operation is not supported while the Shared Block Storage device is in the current state.
404	InvalidVolumeId.NotFound	The specified volume does not exist.	The error message returned because the specified Shared Block Storage device does not exist. Check whether the Shared Block Storage device ID is correct.
403	IdempotentParameterMismatch	The specified clientToken is used.	The error message returned because the specified client token is already in use.
403	IncorrectDiskType.NotSupport	The specified device type is not supported.	The error message returned because the specified disk type does not support the operation.
403	IncorrectDiskStatus.Transferring	The specified device is transferring, you can retry after the process is finished.	The error message returned because the specified disk is being migrated. Wait until the disk is migrated and try again.
403	InvalidParameter.KMSKeyId.CMKNotEnabled	The CMK needs to be enabled.	The error message returned because the customer master key (CMK) is not enabled when a Key Management Service (KMS) key ID is specified for the disk. You can call the DescribeKey operation of KMS to query information about the specified CMK.
403	InvalidParameter.KMSKeyId.KMSUnauthorized	ECS service have no right to access your KMS.	The error message returned because ECS is not authorized to access your KMS resources.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred. Try again later. If the error persists, submit a ticket.
400	Duplicate.TagKey	The Tag.N.Key contain duplicate key.	The error message returned because the specified tag key already exists. Tag keys must be unique.
400	InvalidTagKey.Malformed	The specified Tag.n.Key is not valid.	The error message returned because the specified tag key-related parameter is invalid.
400	InvalidTagValue.Malformed	The specified Tag.n.Value is not valid.	The error message returned because the specified tag value-related parameter is invalid.
403	IdempotentProcessing	The previous idempotent request(s) is still processing.	The error message returned because a previous idempotent request is being processed. Try again later.
400	InvalidRetentionDays.Malformed	The specified RetentionDays is not valid.	The error message returned because the specified RetentionDays parameter is invalid. Check whether the specified InstantAccessRetentionDays parameter is correct.
403	QuotaExceed.Tags	%s	The error message returned because the number of specified tags exceeds the upper limit. %s is a variable. An error message is dynamically returned based on call conditions.
403	InvalidSnapshotCategory.Malformed	The specified Category is not valid.	The error message returned because the specified snapshot category is invalid. Check whether the specified Category parameter is valid.
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified RegionId parameter does not exist.
400	InvalidParameter.Name	The specified Name is invalid.	The error message returned because the specified Name parameter is invalid.
403	NumberExceed.TooManyDisks	The number of all disks in the request exceeds.	The error message returned because the number of disks exceeds the limit.

3.6.5.14. DescribeSnapshotGroups

Queries the details of one or more snapshot-consistent groups.

Description

You can specify multiple request parameters such as `InstanceId`, `SnapshotGroupId.N`, and `Status.N` in the request. The specified parameters have logical AND relations. Only the specified parameters are included in the filter conditions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeSnapshotGroups	The operation that you want to perform. Set the value to <code>DescribeSnapshotGroups</code> .
RegionId	String	Yes	cn-qingdao-****	The region ID of the instance. You can call the <code>DescribeRegions</code> operation to query the most recent region list. For more information, see <i>DescribeRegions</i> in <i>ECS Developer Guide</i> .
InstanceId	String	No	i-q1a01ac6ae3gcb11****	The ID of the instance. You can call the <code>DescribeInstances</code> operation to query the instance ID.
SnapshotGroupId.N	RepeatList	No	ssg-j6ciyh3k52qp7ovm****	The ID of snapshot-consistent group N. Valid values of N: 1 to 10.
Status.N	RepeatList	No	accomplished	The state of snapshot-consistent group N. Valid values of N: 1 to 3. Valid values: - progressing: The snapshot-consistent group is being created. - accomplished: The snapshot-consistent group is created. - failed: The snapshot-consistent group fails to be created.
Name	String	No	testName	The name of the snapshot-consistent group.
AdditionalAttributes.N	RepeatList	No	META_OPTIONS	The value of attribute N. Valid values of N: 1 to 20. Valid values: - META_OPTIONS: instance metadata - DDH_CLUSTER: dedicated host cluster - NETWORK_PRIMARY_ENI_IP: secondary IP address of the primary elastic network interface (ENI)
NextToken	String	No	caeba0bbb2be03f84eb48b699f0a4883	The query token. Set the value to the <code>NextToken</code> value that is returned in the last call to the <code>DescribeSnapshotGroups</code> operation.
MaxResults	Integer	No	10	The maximum number of entries to return on each page. Maximum value: 100. Default value: 10.

Response parameters

Parameter	Type	Example	Description
NextToken	String	caeba0bbb2be03f84eb48b699f0a4883	The token used to start the next query. 🔔 Note If this parameter is empty, no more data is returned.
RequestId	String	3F9A4CC4-362F-469A-B9EF-B3204EF8AA3A	The ID of the request.
SnapshotGroups	Array of SnapshotGroup		Details of the snapshot-consistent groups.
SnapshotGroup			
CreationTime	String	2021-03-23T10:58:48Z	The time when the snapshot-consistent group was created.
Description	String	This is description.	The description of the snapshot-consistent group.

Parameter	Type	Example	Description
InstanceId	String	i-q1a01ac6ae3gcb11****	The ID of the instance to which the snapshot-consistent group belongs. This parameter has a value only when all disk snapshots in the snapshot-consistent group belong to the same instance.
Name	String	testName	The name of the snapshot-consistent group.
ProgressStatus	String	null	 Note This parameter is unavailable.
SnapshotGroupId	String	ssg-j6ciyh3k52qp7ovm****	The ID of the snapshot-consistent group.
Snapshots	Array of Snapshot		Details of the snapshots in the snapshot-consistent group.
Snapshot			
InstantAccess	Boolean	true	Indicates whether the instant access feature is enabled. Valid values: - true: The instant access feature is enabled. This feature can be enabled only for premium performance disks and standard performance disks. - false: The instant access feature is disabled. The snapshot is a normal snapshot for which the instant access feature is disabled.
InstantAccessRetentionDays	Integer	1	The retention period of the instant access feature. After the retention period ends, the snapshot is automatically released.
Progress	String	100%	The progress of the snapshot creation task. Unit: percent (%).
SnapshotId	String	s-j6cbzmrlbf09w72q****	The ID of the snapshot.
SourceDiskId	String	d-j6c3ogynmvp16wy7****	The ID of the source disk. This parameter is retained even after the source disk of the snapshot is released.
SourceDiskType	String	system	The type of the source disk. Valid values: - system: system disk - data: data disk
Status	String	accomplished	The state of the snapshot-consistent group. Valid values: - progressing: The snapshot-consistent group is being created. - accomplished: The snapshot-consistent group is created. - failed: The snapshot-consistent group fails to be created.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=DescribeSnapshotGroups
&RegionId=cn-qingdao-****
&SnapshotGroupId.1=ssg-j6ciyh3k52qp7ovm****
&<Common request parameters>
```

Sample success responses

XML **format**

```
<DescribeSnapshotGroupsResponse>
  <NextToken>caeba0bbb2be03f84eb48b699f0a4883</NextToken>
  <RequestId>3F9A4CC4-362F-469A-B9EF-B3204EF8AA3A </RequestId>
  <SnapshotGroups>
    <Status>accomplished</Status>
    <ProgressStatus>null</ProgressStatus>
    <Description>This is description. </Description>
    <InstanceId>i-qla01ac6ae3gcb1l****</InstanceId>
    <CreationTime>2021-03-23T10:58:48Z </CreationTime>
    <SnapshotGroupId>ssg-j6ciyh3k52qp7ovm**** </SnapshotGroupId>
    <Name>testName </Name>
    <Snapshots>
      <InstantAccess>true</InstantAccess>
      <SnapshotId>s-j6cbzmlbf09w72q**** </SnapshotId>
      <Progress>100% </Progress>
      <SourceDiskType>system</SourceDiskType>
      <InstantAccessRetentionDays>1</InstantAccessRetentionDays>
      <SourceDiskId>d-j6c3ogynmvp16wy7**** </SourceDiskId>
    </Snapshots>
  </SnapshotGroups>
</DescribeSnapshotGroupsResponse>
```

JSON format

```
{
  "NextToken": "caeba0bbb2be03f84eb48b699f0a4883",
  "RequestId": "3F9A4CC4-362F-469A-B9EF-B3204EF8AA3A",
  "SnapshotGroups": {
    "Status": "accomplished",
    "ProgressStatus": "null",
    "Description": "This is description.",
    "InstanceId": "i-qla01ac6ae3gcb1l****",
    "CreationTime": "2021-03-23T10:58:48Z",
    "SnapshotGroupId": "ssg-j6ciyh3k52qp7ovm****",
    "Name": "testName",
    "Snapshots": {
      "InstantAccess": true,
      "SnapshotId": "s-j6cbzmlbf09w72q****",
      "Progress": "100%",
      "SourceDiskType": "system",
      "InstantAccessRetentionDays": 1,
      "SourceDiskId": "d-j6c3ogynmvp16wy7****"
    }
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	InvalidStatus.ValueNotSupported	%s	The error message returned because the operation is not supported while the resource is in the current state.

3.6.5.15. DeleteSnapshotGroup

Deletes a snapshot-consistent group.

Description

If you have custom images that were created from a disk snapshot contained in a snapshot-consistent group, the disk snapshot is retained when the snapshot-consistent group is deleted. Before you can delete the disk snapshot, you must call the `DeleteImage` operation to delete the custom images. Then, you can call the `DeleteSnapshot` operation to delete the disk snapshot.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteSnapshotGroup	The operation that you want to perform. Set the value to DeleteSnapshotGroup.
RegionId	String	Yes	cn-qingdao-****	The region ID of the snapshot-consistent group. You can call the DescribeRegions operation to query the most recent region list.
SnapshotGroupId	String	Yes	ssg-j6ciyh3k52qp7ovm****	The ID of the snapshot-consistent group. You can call the DescribeSnapshotGroups operation to query the IDs of snapshot-consistent groups.

Response parameters

Parameter	Type	Example	Description
OperationProgressSet	Array of OperationProgress		Details about the delete operation.
OperationProgress			
ErrorCode	String	400	The error code. This parameter is empty when the operation was successful.
ErrorMsg	String	testErrorMsg	The error message. This parameter is empty when the operation was successful.
OperationStatus	String	Success	Indicates whether the operation was successful. If the operation was successful, a value of Success is returned. If the operation failed, an error code and an error message are returned.
RelatedItemSet	Array of RelatedItem		Details about the resources.
RelatedItem			
Name	String	SnapshotId	The name of the resource.
Value	String	s-j6c9lpuyxo2uxnx****	The ID of the resource.
RequestId	String	6EDE885A-FDC1-4FAE-BC44-6EACAEA6****	The ID of the request.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=DeleteSnapshotGroup
&RegionId=cn-qingdao-****
&SnapshotGroupId=ssg-j6ciyh3k52qp7ovm****
&<Common request parameters>
```

Sample success responses

XML format

```
<DeleteSnapshotGroupResponse>
  <RequestId>6EDE885A-FDC1-4FAE-BC44-6EACAEA6****</RequestId>
  <OperationProgressSet>
    <OperationStatus>Success</OperationStatus>
    <ErrorMsg>testErrorMsg</ErrorMsg>
    <ErrorCode>400 </ErrorCode>
    <RelatedItemSet>
      <Value></Value>
      <Name></Name>
    </RelatedItemSet>
  </OperationProgressSet>
</DeleteSnapshotGroupResponse>
```

JSON format

```
{
  "RequestId": "6EDE885A-FDC1-4FAE-BC44-6EACAEA6****",
  "OperationProgressSet": {
    "OperationStatus": "Success",
    "ErrorMsg": "testErrorMsg",
    "ErrorCode": 400,
    "RelatedItemSet": {
      "Value": "",
      "Name": ""
    }
  }
}
```

3.6.5.16. ModifySnapshotGroup

Modifies the name or description of a snapshot-consistent group.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifySnapshotGroup	The operation that you want to perform. Set the value to ModifySnapshotGroup.
RegionId	String	Yes	cn-qingdao-****	The region ID of the snapshot-consistent group. You can call the DescribeRegions operation to query the most recent region list. For more information, see <i>DescribeRegions</i> in <i>ECS Developer Guide</i> .
SnapshotGroupId	String	Yes	ssg-j6ciyh3k52qp7ovm****	The ID of the snapshot-consistent group. You can call the DescribeSnapshotGroups operation to query the IDs of one or more snapshot-consistent groups.
Name	String	No	testName02	The new name of the snapshot-consistent group. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). It must start with a letter and cannot start with <code>http://</code> or <code>https://</code> .
Description	String	No	This is new description	The new description of the snapshot-consistent group. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .

Response parameters

Parameter	Type	Example	Description
RequestId	String	A00B5E55-76B7-42C8-8A80-AF10E980****	The ID of the request.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=ModifySnapshotGroup
&RegionId=cn-qingdao-****
&SnapshotGroupId=ssg-j6ciyh3k52qp7ovm****
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifySnapshotGroupResponse>
  <RequestId>A00B5E55-76B7-42C8-8A80-AF10E980****</RequestId>
</ModifySnapshotGroupResponse>
```

JSON format

```
{
  "RequestId": "A00B5E55-76B7-42C8-8A80-AF10E980****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidParameter.Name	The specified Name is invalid.	The error message returned because the specified Name parameter is invalid.

3.6.6. Storage sets

3.6.6.1. CreateStorageSet

Creates a storage set in a zone within a specified region.

Description

A storage set can distribute disks or Shared Block Storage devices to different locations. You can specify the number of partitions in a storage set. A larger number of partitions indicate more discrete distribution of disks or Shared Block Storage devices.

For more information about storage sets, see *Create a storage set in Defined Storage User Guide*.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateStorageSet	The operation that you want to perform. Set the value to CreateStorageSet.
RegionId	String	Yes	cn-qingdao-****-d01	The region in which you want to create a storage set. You can call the DescribeRegions operation to query the most recent region list.
ZoneId	String	Yes	cn-qingdao-****-d01-a	The zone in which you want to create the storage set. You can call the DescribeZones operation to query the most recent zone list.
StorageSetName	String	No	storageSetTest	The name of the storage set. The name must be 2 to 128 characters in length. The name must start with a letter but cannot start with <code>http://</code> or <code>https://</code> . The name can contain letters, digits, colons (:), underscores (_), and hyphens (-).
MaxPartitionNumber	Integer	No	10	The maximum number of partitions in the storage set. Valid values: greater than or equal to 2. Default value: 2.
Description	String	No	storageSetTest	The description of the storage set. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The ClientToken value can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see <i>How to ensure idempotence in ECS Developer Guide</i> .

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3****	The ID of the request.
StorageSetId	String	ss-bp67acfmazb4ph***	The ID of the storage set.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=CreateStorageSet
&RegionId=cn-qingdao-****-d01
&ZoneId=cn-qingdao-****-d01-a
&StorageSetName=testStorageSetName
&MaxPartitionNumber=10
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateStorageSetResponse>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3****</RequestId>
  <StorageSetId>ss-bp67acfmazb4ph***</StorageSetId>
</CreateStorageSetResponse>
```

JSON format

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3****",
  "StorageSetId": "ss-bp67acfmazb4ph***"
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidClientToken.ValueNotSupported	clienttoken is not valid.	The error message returned because the specified ClientToken parameter is invalid.

HTTP status code	Error code	Error message	Description
403	InvalidUser.NotSupported	%s	The error message returned because your account does not support this operation.
403	InvalidRegionId.NotFound	%s	The error message returned because the specified RegionId parameter does not exist.
403	InvalidParameter	%s	The error message returned because a specified parameter is invalid.

3.6.6.2. DescribeStorageSets

Queries the information of one or more storage sets, such as the region, zone, creation time, and maximum number of partitions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeStorageSets	The operation that you want to perform. Set the value to DescribeStorageSets.
RegionId	String	Yes	cn-qingdao-****-d01	The region to which the storage set belongs. You can call the DescribeRegions operation to query the list of supported regions.
ClientToken	String	No	123e4567-e89b-12d3-a456-42665544****	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The value of ClientToken can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see <i>Create a storage set in Cloud Defined Storage (CDS) User Guide</i> .
StorageSetIds	String	No	["ss-bp1d6tsvzmfghy7y****", "ss-bp1jppxbaql9zet7****", ... "ss-bp1ib7bcz07l****"]	The list of storage set IDs to return. The value is a JSON array that consists of up to 100 storage set IDs. The storage set IDs must be separated by commas (,).
ZoneId	String	No	cn-qingdao-****-d01-a	The zone to which the queried disk or Shared Block Storage device belongs.
StorageSetName	String	No	storageSetTest	The name of the storage set. The name must be 2 to 128 characters in length. It must start with a letter but cannot start with <code>http://</code> or <code>https://</code> . It can contain letters, digits, colons (:), underscores (_), and hyphens (-).
PageNumber	Integer	No	1	The number of the page to return. Page start from page 1. Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. Maximum value: 100. Default value: 10.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3****	The ID of the request.
StorageSets	Array of StorageSet		Details about the storage sets. The value of this parameter is an array that consists of StorageSet data.
StorageSet			
CreationTime	String	2022-03-31T00:00:00Z	The time when the storage set was created.
Description	String	testDescription	The description of the storage set.

Parameter	Type	Example	Description
RegionId	String	cn-qingdao-****-d01	The ID of the region to which the storage set belongs.
StorageSetId	String	ss-bp67acfmazb4ph****	The ID of the storage set.
StorageSetName	String	testStorageSetName	The name of the storage set.
StorageSetPartitionNumber	Integer	3	The maximum number of partitions supported by the storage set.
ZoneId	String	cn-qingdao-****-d01-a	The zone to which the queried disk or Shared Block Storage device belongs.
TotalCount	Integer	20	The total number of storage sets.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=DescribeStorageSets
&RegionId=cn-qingdao-****-d01
&PageSize=1
&PageNumber=1
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeStorageSetsResponse>
  <TotalCount>20</TotalCount>
  <PageSize>10</PageSize>
  <RequestId>473469C7-AA6F-4DC5-B3DB-A3DC0DE3****</RequestId>
  <PageNumber>1</PageNumber>
  <StorageSets>
    <Description>testDescription</Description>
    <ZoneId>cn-qingdao-****-d01-a</ZoneId>
    <CreationTime>2022-03-31T00:00:00Z</CreationTime>
    <StorageSetPartitionNumber>3</StorageSetPartitionNumber>
    <RegionId>cn-qingdao-****-d01</RegionId>
    <StorageSetId>ss-bp67acfmazb4ph****</StorageSetId>
    <StorageSetName>testStorageSetName</StorageSetName>
  </StorageSets>
</DescribeStorageSetsResponse>
```

JSON format

```
{
  "TotalCount": 20,
  "PageSize": 10,
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3****",
  "PageNumber": 1,
  "StorageSets": {
    "Description": "testDescription",
    "ZoneId": "cn-qingdao-****-d01-a",
    "CreationTime": "2022-03-31T00:00:00Z",
    "StorageSetPartitionNumber": 3,
    "RegionId": "cn-qingdao-****-d01",
    "StorageSetId": "ss-bp67acfmazb4ph****",
    "StorageSetName": "testStorageSetName"
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidClientToken.ValueNotSupported	clienttoken is not valid.	The error message returned because the specified ClientToken parameter is invalid.
403	InvalidUser.NotSupported	%s	The error message returned because your account does not support this operation.
403	InvalidRegionId.NotFound	%s	The error message returned because the specified RegionId parameter does not exist.

HTTP status code	Error code	Error message	Description
403	InvalidParameter	%s	The error message returned because a specified parameter is invalid.

3.6.6.3. DescribeStorageSetDetails

Queries the detailed information of disks or Shared Block Storage devices in a storage set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeStorageSetDetails	The operation that you want to perform. Set the value to DescribeStorageSetDetails.
RegionId	String	Yes	cn-qingdao-****-d01	The region to which the storage set belongs. You can call the DescribeRegions operation to query the most recent region list.
StorageSetId	String	Yes	ss-bp67acfmazb4ph****	The ID of the storage set. You can call the DescribeStorageSets operation to query available resource set IDs.
ClientToken	String	No	123e4567-e89b-12d3-a456-42665544****	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that it is unique among different requests. The ClientToken value can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see the <i>Create a storage set</i> topic in <i>Cloud Defined Storage (CDS) User Guide</i> .
StorageSetPartitionNumber	Integer	No	10	The maximum number of partitions in the storage set.
DiskIds	String	No	["d-bp1d6tsvznfghy7y****", "d-bp1ippxbag9zet7****", ... "d-bp1ib7bcz07l****"]	The IDs of disks or Shared Block Storage devices. The value can be a JSON array that consists of up to 100 disk or Shared Block Storage device IDs. Separate the IDs with commas (,).
PageNumber	Integer	No	1	The number of the page to return. - Pages start from page 1. - Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. - Maximum value: 100. - Default value: 10.

Response parameters

Parameter	Type	Example	Description
Disks	Array of Disk		Details about disks or Shared Block Storage devices in the storage set. It is an array that consists of disk information.
Disk			
Category	String	cloud_ssd	The category of the disk or Shared Block Storage device.
CreationTime	String	2022-03-31T00:00:00Z	The time when the disk or Shared Block Storage device was created.
DiskId	String	d-bp67acfmazb4p****	The ID of the disk or Shared Block Storage device.
DiskName	String	testDiskName	The name of the disk or Shared Block Storage device.
RegionId	String	cn-qingdao-****-d01	The region to which the disk or Shared Block Storage device belongs.
StorageSetId	String	ss-bp67acfmazb4ph****	The ID of the storage set.
StorageSetPartitionNumber	Integer	10	The number of partitions in the storage set.
ZoneId	String	cn-qingdao-****-d01-a	The zone to which the disk or Shared Block Storage device belongs.

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	1	The number of entries returned per page.
RequestId	String	49765E79-0D5D-4451-B3AE-580A2083****	The ID of the request.
TotalCount	Integer	1	The total number of storage sets.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=DescribeStorageSetDetails
&RegionId=cn-qingdao-****-d01
&StorageSetId=ss-bp67acfmazb4ph****
&<Common request parameters>
```

Sample success responses

XML **format**

```
<DescribeStorageSetDetailsResponse>
  <TotalCount>1</TotalCount>
  <PageSize>1</PageSize>
  <RequestId>49765E79-0D5D-4451-B3AE-580A2083****</RequestId>
  <PageNumber>1</PageNumber>
  <Disks>
    <Category>cloud_ssd</Category>
    <ZoneId>cn-qingdao-****-d01-a</ZoneId>
    <CreationTime>2022-03-31T00:00:00Z</CreationTime>
    <StorageSetPartitionNumber>10</StorageSetPartitionNumber>
    <RegionId>cn-qingdao-****-d01</RegionId>
    <StorageSetId>ss-bp67acfmazb4ph****</StorageSetId>
    <DiskName>testDiskName</DiskName>
    <DiskId>d-bp67acfmazb4p****</DiskId>
  </Disks>
</DescribeStorageSetDetailsResponse>
```

JSON **format**

```
{
  "TotalCount": 1,
  "PageSize": 1,
  "RequestId": "49765E79-0D5D-4451-B3AE-580A2083****",
  "PageNumber": 1,
  "Disks": {
    "Category": "cloud_ssd",
    "ZoneId": "cn-qingdao-****-d01-a",
    "CreationTime": "2022-03-31T00:00:00Z",
    "StorageSetPartitionNumber": 10,
    "RegionId": "cn-qingdao-****-d01",
    "StorageSetId": "ss-bp67acfmazb4ph****",
    "DiskName": "testDiskName",
    "DiskId": "d-bp67acfmazb4p****"
  }
}
```

Error codes

HttpCode	Error code	Error message	Description
400	InvalidClientToken.ValueNotSupported	clienttoken is not valid.	The error message returned because the specified ClientToken parameter is invalid.
403	InvalidUser.NotSupported	%s	The error message returned because your account does not support this operation.
403	InvalidRegionId.NotFound	%s	The error message returned because the specified RegionId parameter does not exist.
403	InvalidParameter	%s	The error message returned because a specified parameter is invalid.

3.6.6.4. ModifyStorageSetAttribute

Modifies the name or description of a storage set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyStorageSetAttribute	The operation that you want to perform. Set the value to <code>ModifyStorageSetAttribute</code> .
RegionId	String	Yes	cn-qingdao-****-d01	The region to which the storage set belongs. You can call the <code>DescribeRegions</code> operation to query the list of supported regions.
StorageSetId	String	Yes	ss-bp67acfmazb4ph****	The ID of the storage set. You can call the <code>DescribeStorageSets</code> operation to query the ID of the storage set that you want to modify.
Description	String	No	testStorageSetDescription	The description of the storage set. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
StorageSetName	String	No	testStorageSetName	The name of the storage set. The name must be 2 to 128 characters in length. The name must start with a letter and cannot start with <code>http://</code> or <code>https://</code> . The name can contain letters, digits, colons (:), underscores (_), and hyphens (-).
ClientToken	String	No	123e4567-e89b-12d3-a456-42665544****	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The value of <code>ClientToken</code> can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see <i>How to ensure idempotence in ECS Developer Guide</i> .

Response parameters

Parameter	Type	Example	Description
RequestId	String	04F0F334-1335-436C-A1D7-6C044FE7****	The ID of the request.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=ModifyStorageSetAttribute
&RegionId=cn-qingdao-****-d01
&StorageSetId=ss-bp67acfmazb4ph****
&StorageSetName=testStorageSetName
&Description=testStorageSetDescription
&<Common request parameters>
```

Sample success responses

XML **format**

```
<ModifyStorageSetAttributeResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE7****</RequestId>
</ModifyStorageSetAttributeResponse>
```

JSON **format**

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE7****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidClientToken.ValueNotSupported	clienttoken is not valid.	The error message returned because the specified <code>ClientToken</code> parameter is invalid.
403	InvalidUser.NotSupported	%s	The error message returned because your account does not support this operation.
403	InvalidRegionId.NotFound	%s	The error message returned because the specified <code>RegionId</code> parameter does not exist.
403	InvalidParameter	%s	The error message returned because a specified parameter is invalid.

3.6.6.5. DeleteStorageSet

Deletes an empty storage set. The storage set that you want to delete must be empty and cannot contain disks or Shared Block Storage devices.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteStorageSet	The operation that you want to perform. Set the value to DeleteStorageSet.
RegionId	String	Yes	cn-qingdao-****-d01	The region to which the storage set belongs. You can call the DescribeRegions operation to query the list of supported regions.
StorageSetId	String	Yes	ss-bp67acfmazb4ph***	The ID of the storage set. You can call the DescribeStorageSets operation to query the ID of the storage set that you want to delete.
ClientToken	String	No	123e4567-e89b-12d3-a456-42665544****	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The value of ClientToken can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see <i>How to ensure idempotence</i> in <i>ECS Developer Guide</i> .

Response parameters

Parameter	Type	Example	Description
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3****	The ID of the request.

Examples

Sample requests

```
http(s)://ecs.example.com/?Action=DeleteStorageSet
&RegionId=cn-qingdao-****-d01
&StorageSetId=ss-bp67acfmazb4ph***
&<Common request parameters>
```

Sample success responses

XML format

```
<DeleteStorageSetResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE7****</RequestId>
</DeleteStorageSetResponse>
```

JSON format

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE7****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
400	InvalidClientToken.ValueNotSupported	clienttoken is not valid.	The error message returned because the specified ClientToken parameter is invalid.
403	InvalidUser.NotSupported	%s	The error message returned because your account does not support this operation.
403	InvalidRegionId.NotFound	%s	The error message returned because the specified RegionId parameter does not exist.
403	InvalidParameter	%s	The error message returned because a specified parameter is invalid.

3.6.7. Asynchronous replication

3.6.7.1. CreateDiskReplicaPair

Creates a replication pair.

Usage notes

The asynchronous replication feature can be used to protect data across zones in a region based on the data replication capability of EBS. This feature can be used to asynchronously replicate data from a disk in a zone of a region to a disk in another zone of the region for disaster recovery. You can use this feature to implement disaster recovery for critical business to protect data in your databases and improve business continuity.

Before you call this operation, take note of the following items:

- Make sure that the primary disk and secondary disk for disaster recovery are prepared. The secondary disk cannot be deployed in the same zone as the primary disk.
- After you call this operation to create a replication pair between the primary disk and the secondary disk, you must call the `StartDiskReplicaPair` operation to enable asynchronous replication for the replication pair. Then, data can be periodically replicated from the primary disk to the secondary disk across zones in the same region.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDiskReplicaPair	The operation that you want to perform. Set the value to <code>CreateDiskReplicaPair</code> .
DestinationDiskId	String	Yes	d-sa1f82p58p1tdw9g****	The ID of the secondary disk.
DestinationRegionId	String	Yes	cn-beijing	The ID of the region in which the secondary disk is deployed.
DestinationZoneId	String	Yes	cn-beijing-e	The ID of the zone in which the secondary disk is deployed.
DiskId	String	Yes	d-iq80sgp4d0xbk24q****	The ID of the primary disk.
RegionId	String	Yes	cn-beijing	The ID of the region in which the replication pair is deployed.
SourceZoneId	String	Yes	cn-beijing-f	The ID of the zone in which the primary disk is deployed.
PairName	String	No	TestReplicaPair	The name of the replication pair. The name must meet the following requirements: • The name must be 2 to 128 characters in length. • The name must start with a letter and cannot start with <code>http://</code> or <code>https://</code>. • The name can contain letters, digits, colons (:), underscores (_), periods (.), and hyphens (-).
Description	String	No	This is description.	The description of the replication pair. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence .
Bandwidth	Long	No	81920	The bandwidth that is used to asynchronously replicate data from the primary disk to the secondary disk. Unit: Kbit/s. Valid values: 81920 to 655360. Default value: 81920. Step size: 8192.
RPO	Long	No	120	The RPO of the replication pair. Unit: seconds. Valid values: 120 to 604800. Default value: 120. Step size: 60.

Response parameters

Parameter	Type	Example	Description
ReplicaPairId	String	pair-cn-dsa****	The ID of the replication pair.

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateDiskReplicaPair
&DestinationDiskId=d-salf82p58pltdw9g****
&DestinationRegionId=cn-beijing
&DestinationZoneId=cn-beijing-e
&DiskId=d-iq80sgp4d0xbk24q****
&RegionId=cn-beijing
&SourceZoneId=cn-beijing-f
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450 </RequestId>
<ReplicaPairId>pair-cn-dsa**** </ReplicaPairId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450",
  "ReplicaPairId": "pair-cn-dsa****"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.

HttpCode	Error code	Error message	Description
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.2. StartDiskReplicaPair

Enables the asynchronous replication feature for a replication pair to periodically replicate data from the primary disk to the secondary disk across regions.

Usage notes

- If you want to perform an automatic synchronization by setting the OneShot parameter to false, make sure that the replication pair is in the **created**, **initial_syncing**, **syncing**, **normal**, or **stopped** state and is not added to a replication pair-consistent group.
- If you want to perform manual synchronization by setting the OneShot parameter to true, make sure that the replication pair is in the **created**, **manual_syncing**, or **stopped** state and is not added to a replication pair-consistent group.
- After you enable the asynchronous replication feature for a replication pair, the replication pair enters the **initial_syncing** state. The system performs the first asynchronous replication to replicate all data from the primary disk to the secondary disk.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	StartDiskReplicaPair	The operation that you want to perform. Set the value to StartDiskReplicaPair.
RegionId	String	Yes	cn-beijing	The region ID of the primary or secondary disk in the replication pair. You can call the DescribeDiskReplicaPairs operation to query the region information of replication pairs.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.
OneShot	Boolean	No	true	Specifies whether to immediately synchronize data. Valid values: • true: immediately synchronizes data. • false: synchronizes data based on the RPO of the replication pair-consistent group. This is the default value.

Response parameters

Parameter	Type	Example	Description
RequestId	String	A37597A6-BB99-19B3-85EA-4C2B91F00123	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=StartDiskReplicaPair
&RegionId=cn-beijing
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>A37597A6-BB99-19B3-85EA-4C2B91F00123</RequestId>
```

JSON format

```
{
  "RequestId": "A37597A6-BB99-19B3-85EA-4C2B91F00123"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.

HttpCode	Error code	Error message	Description
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.3. DescribeDiskReplicaPairs

Queries the information of one or more replication pairs in a region.

Usage notes

- When you call this operation for a specific region, if the primary or secondary disk of a replication pair is located within the region, the information of the replication pair is contained in the response.
- We recommend that you use the MaxResults and NextToken parameters for a paged query. During a paged query, when you call the DescribeDiskReplicaPairs operation to retrieve the first page of results, set MaxResults to specify the maximum number of entries to return in the call. The return value of NextToken is a pagination token, which can be used in the next call to retrieve a new page of results. When you call the DescribeDiskReplicaPairs operation to retrieve a new page of results, set NextToken to the NextToken value returned in the previous call and set MaxResults to specify the maximum number of entries to return in this call.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDiskReplicaPairs	The operation that you want to perform. Set the value to DescribeDiskReplicaPairs.
RegionId	String	Yes	cn-beijing	The region ID of the primary disk in the replication pair.
MaxResults	Long	No	1	The maximum number of entries to return on each page. Valid values: 1 to 500. Default value: 10.
NextToken	String	No	AAAAAdWBF2****	The query token. Set the value to the NextToken value returned in the previous call to the DescribeDiskReplicaPairs operation. Leave this parameter empty the first time you call this operation.
PairIds	String	No	pair-cn-dsa****,pair-cn-asd****	The IDs of the replication pairs. You can specify the IDs of one or more replication pairs. Separate the IDs with commas (,). This parameter is empty by default, which indicates that all replication pairs in the specified region are queried.
Site	String	No	production	The type of the site from which the information of replication pairs is queried. - If the Site parameter is not specified, information such as the state of replication pairs at the primary site is queried and returned. - Otherwise, information such as the state of replication pairs at the site specified by the Site parameter is queried and returned. Valid values: - production: primary site - backup: secondary site
ReplicaGroupId	String	No	pg-myreplica****	The ID of the replication pair-consistent group. If you specify a replication pair-consistent group, the information of the replication pairs in the group is contained in the response. If you set the ReplicaGroupId parameter to a hyphen (-), the information of the replication pairs that do not belong to any replication pair-consistent group is contained in the response.

Response parameters

Parameter	Type	Example	Description
NextToken	String	AAAAAdWBF2****	The query token returned in this call.

Parameter	Type	Example	Description
ReplicaPairs	Array of ReplicaPair		The details of replication pairs.
Bandwidth	Long	10240	The bandwidth that is used to asynchronously replicate data from the primary disk to the secondary disk. Unit: Kbit/s.
CreateTime	Long	1637835114	The time when the replication pair was created. The value of this parameter is a timestamp. Unit: seconds.
Description	String	This is description.	The description of the replication pair.
DestinationDiskId	String	d-asdfjl2342kj2l3k4****	The ID of the secondary disk.
DestinationRegion	String	cn-shanghai	The ID of the region in which the secondary disk is deployed.
DestinationZoneId	String	cn-shanghai-e	The ID of the zone in which the secondary disk is deployed.
LastRecoverPoint	Long	1637835714	The time when data was last replicated from the primary disk to the secondary disk. The value of this parameter is a timestamp. Unit: seconds.
PairName	String	TestReplicaPair	The name of the replication pair.
PrimaryRegion	String	cn-beijing	The region ID of the initial primary disk in the replication pair.
PrimaryZone	String	cn-beijing-f	The zone ID of the initial primary disk in the replication pair.
RPO	Long	2	The RPO of the replication pair. Unit: seconds.
ReplicaGroupId	String	pg-myreplica****	The ID of the replication pair-consistent group to which the replication pair belongs.
ReplicaGroupName	String	myreplicagrouptest	The name of the replication pair-consistent group to which the replication pair belongs.
ReplicaPairId	String	pair-cn-dsa****	The ID of the replication pair.
Site	String	production	The type of the site from which the information of the replication pair and replication pair-consistent group is obtained.
SourceDiskId	String	d-bp131n0q38u3a4zi****	The ID of the primary disk.
SourceRegion	String	cn-beijing	The ID of the region in which the primary disk is deployed.
SourceZoneId	String	cn-beijing-f	The ID of the zone in which the primary disk is deployed.
StandbyRegion	String	shanghai	The region ID of the initial secondary disk in the replication pair.
StandbyZone	String	shanghai-e	The zone ID of the initial secondary disk in the replication pair.

Parameter	Type	Example	Description
Status	String	created	The states of the replication pair. Valid values: - invalid: The replication pair is invalid. This state indicates that the replication pair is abnormal. - creating: The replication pair is being created. - created: The replication pair is created. - create_failed: The replication pair cannot be created. - initial_syncing: Data is synchronized between the disks for the first time. After you create a replication pair and enable asynchronous replication for the replication pair, the replication pair is in this state the first time data is synchronized from the primary disk to the secondary disk. - manual_syncing: Data is being manually synchronized between the disks. When data is being manually synchronized for the first time, the replication pair is in this state. - syncing: Data is being synchronized between the disks. When data is being asynchronously replicated from the primary disk to the secondary disk not for the first time, the replication pair is in this state. - normal: The replication pair is working normally. When the system finishes replicating data from the primary disk to the secondary disk within the current replication cycle, the replication pair enters this state. - stopping: The replication pair is being stopped. - stopped: The replication pair is stopped. - stop_failed: The replication pair cannot be stopped. - failovering: A failover is being performed. - failovered: A failover is performed. - failover_failed: A failover cannot be performed. - reprotecting: A reverse replication is being performed. - reprotect_failed: A reverse replication cannot be performed. - deleting: The replication pair is being deleted. - delete_failed: The replication pair cannot be deleted. - deleted: The replication pair is deleted.
StatusMessage	String	DeviceRemoved	The message that describes the state of the replication pair. This parameter has a value only when the value of <code>Status</code> is invalid. Valid values: - DeviceRemoved: The primary or secondary disk has been deleted. - DeviceKeyChanged: The DeviceKey mapping of the primary or secondary disk has changed.
RequestId	String	AAA478A0-BEE6-1D42-BEB6-A9CFEAD64B29	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DescribeDiskReplicaPairs
&RegionId=cn-beijing
&<Common request parameters>
```

Sample success responses

XML format

```
<ReplicaPairs>
  <Site>production</Site>
  <Status>created</Status>
  <StandbyZone>shanghai-e</StandbyZone>
  <Description>This is description. </Description>
  <DestinationDiskId>d-asdfjl2342kj2l3k4**** </DestinationDiskId>
  <PrimaryRegion>cn-beijing </PrimaryRegion>
  <CreateTime>1637835114</CreateTime>
  <DestinationRegion>cn-shanghai </DestinationRegion>
  <LastRecoverPoint>1637835714</LastRecoverPoint>
  <ReplicaPairId>pair-cn-dsa****</ReplicaPairId>
  <ReplicaGroupId>pg-myreplica****</ReplicaGroupId>
  <SourceRegion>cn-beijing </SourceRegion>
  <SourceDiskId>d-bp13ln0q38u3a4zi**** </SourceDiskId>
  <StatusMessage>DeviceRemoved</StatusMessage>
  <RPO>2</RPO>
  <ReplicaGroupName>myreplicagrouptest</ReplicaGroupName>
  <Bandwidth>10240</Bandwidth>
  <PairName>TestReplicaPair</PairName>
  <DestinationZoneId>cn-shanghai-e </DestinationZoneId>
  <PrimaryZone>cn-beijing-f</PrimaryZone>
  <SourceZoneId>cn-beijing-f</SourceZoneId>
  <StandbyRegion>shanghai </StandbyRegion>
</ReplicaPairs>
<NextToken>AAAAAdDWBf2**** </NextToken>
<RequestId>AAA478A0-BEE6-1D42-BEB6-A9CFEAD64B29 </RequestId>
```

JSON **format**

```
{
  "ReplicaPairs": {
    "Site": "production",
    "Status": "created",
    "StandbyZone": "shanghai-e",
    "Description": "This is description.",
    "DestinationDiskId": "d-asdfjl2342kj2l3k4****",
    "PrimaryRegion": "cn-beijing",
    "CreateTime": 1637835114,
    "DestinationRegion": "cn-shanghai",
    "LastRecoverPoint": 1637835714,
    "ReplicaPairId": "pair-cn-dsa****",
    "ReplicaGroupId": "pg-myreplica****",
    "SourceRegion": "cn-beijing",
    "SourceDiskId": "d-bp13ln0q38u3a4zi****",
    "StatusMessage": "DeviceRemoved",
    "RPO": 2,
    "ReplicaGroupName": "myreplicagrouptest",
    "Bandwidth": 10240,
    "PairName": "TestReplicaPair",
    "DestinationZoneId": "cn-shanghai-e",
    "PrimaryZone": "cn-beijing-f",
    "SourceZoneId": "cn-beijing-f",
    "StandbyRegion": "shanghai"
  },
  "NextToken": "AAAAAdDWBf2****",
  "RequestId": "AAA478A0-BEE6-1D42-BEB6-A9CFEAD64B29"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.

HttpCode	Error code	Error message	Description
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.4. DescribeDiskReplicaPairProgress

Obtains the replication progress of a replication pair.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDiskReplicaPairProgress	The operation that you want to perform. Set the value to DescribeDiskReplicaPairProgress.
RegionId	String	Yes	cn-beijing	The ID of the region in which the primary disk is deployed.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.

Response parameters

Parameter	Type	Example	Description
Progress	Integer	100	The replication progress of the replication pair. Valid values: 0, 50, and 100.
RecoverPoint	Long	1637835114	The time when data was last replicated from the primary disk to the secondary disk. Unit: seconds.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DescribeDiskReplicaPairProgress
&RegionId=cn-beijing
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<Progress>100</Progress>
<RecoverPoint>1637835114</RecoverPoint>
```

JSON format

```
{
  "Progress": 100,
  "RecoverPoint": 1637835114
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.

HttpCode	Error code	Error message	Description

403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.5. ModifyDiskReplicaPair

Modifies the name, RPO, or description of a replication pair.

Usage notes

- The replication pair must be in the **created** or **stopped** state.
- After a replication pair is added to a replication pair-consistent group, you cannot directly modify the RPO of the replication pair.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyDiskReplicaPair	The operation that you want to perform. Set the value to ModifyDiskReplicaPair.
RegionId	String	Yes	cn-beijing	The region ID of the primary disk in the replication pair.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.
PairName	String	No	TestReplicaPair	The name of the replication pair.
Description	String	No	This is description.	The description of the replication pair.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.
Bandwidth	Long	No	10240	The bandwidth that is used to asynchronously replicate data from the primary disk to the secondary disk. Unit: Kbit/s. Valid values: 81920 to 655360. Default value: 81920. Step size: 8192.
RPO	Long	No	2	The RPO of the replication pair. Unit: seconds. Valid values: 120 to 604800. Default value: 120. Step size: 60.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ModifyDiskReplicaPair
&RegionId=cn-beijing
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.

HttpCode	Error code	Error message	Description
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.6. FailoverDiskReplicaPair

Enables the failover feature for a replication pair. When the primary disk in a replication pair fails, you can call this operation to enable the read and write permissions on the secondary disk.

Usage notes

- The replication pair must be in the **manual_syncing**, **syncing**, **normal**, **stopping**, **stop_failed**, **stopped**, **failovering**, **failover_failed**, or **failovered** state.
- After a failover is performed, the replication pair enters the **failovered** state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	FailoverDiskReplicaPair	The operation that you want to perform. Set the value to FailoverDiskReplicaPair.
RegionId	String	Yes	cn-hangzhou	The region ID of the secondary disk in the replication pair. You can call the DescribeDiskReplicaPairs operation to obtain the region information of secondary disks in replication pairs. Note The failover feature must be enabled in the region where the secondary disk is deployed.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=FailoverDiskReplicaPair
&RegionId=cn-hangzhou
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.7. ReprotectDiskReplicaPair

Enables the reverse replication feature for a replication pair. After you enable the reverse replication feature, the direction of data replication is inverted. Data is replicated from the original secondary disk to the original primary disk.

Usage notes

- The replication pair must be in the **failovered**, **reprotecting**, or **reprotect_failed** state and not be added to any replication pair-consistent group.

- The primary disk must be detached from its associated ECS instance and is in the Unattached state. You can call the DetachDisk operation to detach disks.
- After you enable the reverse replication feature, you must call the StartDiskReplicaPair operation again to enable the replication pair before data can be replicated from the original secondary disk to the original primary disk.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ReprotectDiskReplicaPair	The operation that you want to perform. Set the value to ReprotectDiskReplicaPair.
RegionId	String	Yes	cn-shanghai	The region ID of the secondary disk in the replication pair. You can call the DescribeDiskReplicaPairs operation to obtain the region information of secondary disks in replication pairs. Note The reverse replication feature must be enabled from the region in which the secondary disk is located.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate a client token. Make sure that unique client tokens are used for different requests. The client token can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample request

```
http(s)://[Endpoint]/?Action=ReprotectDiskReplicaPair
&RegionId=cn-shanghai
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.

HTTP status code	Error code	Error message	Description
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the ClientToken value is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.8. StopDiskReplicaPair

Stops a replication pair.

Usage notes

- The replication pair must be in the **initial_syncing**, **manual_syncing**, **syncing**, **normal**, **stopping**, **stopping_failed**, or **stopped** state and not be added to any replication pair-consistent group. After you stop a replication pair, the replication pair enters the **stopped** state. The secondary disk rolls back to the point in time when the last asynchronous replication was complete and drops all the data that is being replicated from the primary disk.
- If you stop a replication pair that is in the **initial_syncing** state and then perform a failover, an error is returned.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	StopDiskReplicaPair	The operation that you want to perform. Set the value to StopDiskReplicaPair.
RegionId	String	Yes	cn-shanghai	The region ID of the primary or secondary disk in the replication pair. You can call the DescribeDiskReplicaPairs operation to obtain the region information of replication pairs.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.

Parameter	Type	Required	Example	Description
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate a client token. Make sure that unique client tokens are used for different requests. The client token can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	A37597A6-BB99-19B3-85EA-4C2B91F00123	The ID of the request.

Examples

Sample request

```
http(s)://[Endpoint]/?Action=StopDiskReplicaPair
&RegionId=cn-shanghai
&ReplicaPairId=pair-cn-dsa****
&<<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>A37597A6-BB99-19B3-85EA-4C2B91F00123 </RequestId>
```

JSON format

```
{
  "RequestId": "A37597A6-BB99-19B3-85EA-4C2B91F00123"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the ClientToken value is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.

HTTP status code	Error code	Error message	Description
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.7.9. DeleteDiskReplicaPair

Deletes a replication pair.

Usage notes

- The replication pair must be in the **creating**, **created**, **create_failed**, **stopped**, **failovered**, **deleting**, **delete_failed**, or **invalid** state. This operation deletes only replication pairs. The primary and secondary disks in the deleted replication pairs are retained.
- To delete a replication pair, you must call this operation in the region in which the primary disk is located. After the replication pair is deleted, functionality limits are lifted from the primary and secondary disks. For example, you can attach the secondary disk, resize the disk, or read data from or write data to the disk.
- The request initiated by calling this operation will be successful even if you specify a replication pair ID that does not exist or is invalid.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteDiskReplicaPair	The operation that you want to perform. Set the value to DeleteDiskReplicaPair.
RegionId	String	Yes	cn-beijing	The region ID of the primary disk in the replication pair. You can call the DescribeDiskReplicaPairs operation to obtain the region information of replication pairs.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate a client token. Make sure that unique client tokens are used for different requests. The client token can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	A37597A6-BB99-19B3-85EA-4C2B91F00123	The ID of the request.

Examples

Sample request

```
http(s)://[Endpoint]/?Action=DeleteDiskReplicaPair
&RegionId=cn-beijing
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>A37597A6-BB99-19B3-85EA-4C2B91F00123 </RequestId>
```

JSON format

```
{
  "RequestId": "A37597A6-BB99-19B3-85EA-4C2B91F00123"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the ClientToken value is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.

HTTP status code	Error code	Error message	Description
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8. Replication pair-consistent groups

3.6.8.1. CreateDiskReplicaGroup

Creates a replication pair-consistent group.

Usage notes

A replication pair-consistent group is a combination of multiple replication pairs that replicate data in the same direction. A replication pair-consistent group allows you to start or stop multiple replication pairs, perform failovers for multiple replication pairs, or enable the reverse replication feature for multiple replication pairs at a time.

When you create a replication pair-consistent group, take note of the following items:

- Replication pair-consistent groups support disaster recovery across zones within the same region.
- A replication pair and a replication pair-consistent group replicate data in the same direction if they have the same primary region (production region), primary zone (production zone), secondary region (disaster recovery region), and secondary zone (disaster recovery zone). Replication pairs can be added to only a replication pair-consistent group that replicates data in the same direction as them.
- After replication pairs are added to a replication pair-consistent group, the recovery point objective (RPO) of the group takes effect on the pairs in place of their original RPOs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDiskReplicaGroup	The operation that you want to perform. Set the value to CreateDiskReplicaGroup.
DestinationRegionId	String	Yes	cn-shanghai	The ID of the region in which the secondary site is deployed.
DestinationZoneId	String	Yes	cn-shanghai-e	The ID of the zone in which the secondary site is deployed.
RegionId	String	Yes	cn-hangzhou	The ID of the region in which you want to create the replication pair-consistent group. The primary site is deployed in this region. The value of this parameter is also the identifier and a component of Endpoint, as shown in the following sample request.
SourceZoneId	String	Yes	cn-beijing-f	The ID of the zone in which the primary site is deployed.
GroupName	String	No	myreplicagrouptest	The name of the replication pair-consistent group. The name must meet the following requirements: - The name must be 2 to 128 characters in length. - The name must start with a letter and cannot start with <code>http://</code> or <code>https://</code>. - The name can contain letters, digits, colons (:), underscores (_), and hyphens (-). If you do not set this parameter, a name is automatically generated for the replication pair-consistent group.
Description	String	No	This is description.	The description of the replication pair-consistent group. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate a client token. Make sure that unique client tokens are used for different requests. The client token can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence.

Parameter	Type	Required	Example	Description
RPO	Long	No	2	The RPO of the replication pair-consistent group. Unit: seconds. Valid values:[120, 604800] Default value: 120. Step size: 60.

Response parameters

Parameter	Type	Example	Description
ReplicaGroupId	String	pg-myreplica****	The ID of the replication pair-consistent group.
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample request

```
http(s)://[Endpoint]/?Action=CreateDiskReplicaGroup
&DestinationRegionId=cn-shanghai
&DestinationZoneId=cn-shanghai-e
&RegionId=cn-hangzhou
&SourceZoneId=cn-beijing-f
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
<ReplicaGroupId>pg-myreplica****</ReplicaGroupId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450",
  "ReplicaGroupId": "pg-myreplica****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the ClientToken value is used in another request that is being processed. Try again later.

HTTP status code	Error code	Error message	Description
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.
403	OperationDenied.TooMuchEmptyGroup	Operation failed because of too many empty groups.	The error message returned because the maximum number of replication pair-consistent groups that are empty has been reached and thus the operation is not supported.
403	OperationDenied.GroupTotalDiskCapacityLimit	Operation failed because of total disk capacity in group over limit.	The error message returned because the total capacity of disks in the replication pair-consistent group exceeds the maximum value allowed.

3.6.8.2. AddDiskReplicaPair

Adds a replication pair to a replication pair-consistent group.

Usage notes

- A replication pair and a replication pair-consistent group replicate data in the same direction if they have the same primary region (production region), primary zone (production zone), secondary region (disaster recovery region), and secondary zone (disaster recovery zone). Replication pairs can be added to only a replication pair-consistent group that replicates data in the same direction as them.
- Up to 256 replication pairs can be added to a single replication pair-consistent group.
- Before you add a replication pair to a replication pair-consistent group, make sure that the pair and the group are in the **created** or **stopped** state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AddDiskReplicaPair	The operation that you want to perform. Set the value to AddDiskReplicaPair.
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group. The value of this parameter is also the identifier and a component of Endpoint, as shown in the following sample request.
ReplicaGroupId	String	Yes	pg-myreplica****	The ID of the replication pair-consistent group.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.

Parameter	Type	Required	Example	Description
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate a client token. Make sure that unique client tokens are used for different requests. The client token can contain only ASCII characters and cannot exceed 64 characters in length. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample request

```
http(s)://[Endpoint]/?Action=AddDiskReplicaPair
&RegionId=cn-beijing
&ReplicaGroupId=pg-myreplica****
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HTTP status code	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the ClientToken value is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.

HTTP status code	Error code	Error message	Description
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.3. StartDiskReplicaGroup

Enables the asynchronous replication feature for the replication pairs in a replication pair-consistent group. After you enable the asynchronous replication feature, the system first performs a full synchronization to synchronize all data from the primary disks to the secondary disks and then periodically synchronizes incremental data based on the RPO of the replication pair-consistent group.

Usage notes

- If you want to perform automatic synchronization by setting the OneShot parameter to false, make sure that the replication pair-consistent group is in the **created**, **syncing**, **normal**, or **stopped** state.
- If you want to perform manual synchronization by setting the OneShot parameter to true, make sure the replication pair-consistent group is in the **created**, **manual_syncing**, or **stopped** state.
- After you enable the asynchronous replication feature for a replication pair-consistent group, the group enters the **initial_syncing** state. The system performs the first asynchronous replication to replicate all data from the primary disks to the secondary disks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	StartDiskReplicaGroup	The operation that you want to perform. Set the value to StartDiskReplicaGroup.
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group. The value of this parameter is also the identifier and a component of Endpoint, as shown in the following sample request.
ReplicaGroupId	String	Yes	pg-myreplica****	The ID of the replication pair-consistent group.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.
OneShot	Boolean	No	true	Specifies whether to immediately synchronize data. Valid values: - true: immediately synchronizes data. - false: synchronizes data based on the RPO of the replication pair-consistent group. This is the default value.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=StartDiskReplicaGroup
&RegionId=cn-beijing
&ReplicaGroupId=pg-myreplica****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.

HttpCode	Error code	Error message	Description
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.4. DescribeDiskReplicaGroups

Queries the information of one or more replication pair-consistent groups in a region.

Usage notes

We recommend that you use the `MaxResults` and `NextToken` parameters for a paged query.

During a paged query, when you call the `DescribeDiskReplicaGroups` operation to retrieve the first page of results, set `MaxResults` to specify the maximum number of entries to return in the call. The return value of `NextToken` is a pagination token, which can be used in the next call to retrieve a new page of results. When you call the `DescribeDiskReplicaGroups` operation to retrieve a new page of results, set `NextToken` to the `NextToken` value returned in the previous call and set `MaxResults` to specify the maximum number of entries to return in this call.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDiskReplicaGroups	The operation that you want to perform. Set the value to <code>DescribeDiskReplicaGroups</code> .
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group. The value of this parameter is also the identifier and a component of Endpoint, as shown in the following sample request.
MaxResults	Long	No	10	The maximum number of entries to return on each page. Valid values: 1 to 500. Default value: 10.
NextToken	String	No	AAAAAdWBF2****	The query token. Set the value to the <code>NextToken</code> value returned in the previous call to the <code>DescribeDiskReplicaGroups</code> operation. Leave this parameter empty the first time you call this operation.
GroupIds	String	No	pg-cn-dsa****,pg-cn-asd****	The IDs of the replication pair-consistent groups. You can specify the IDs of one or more replication pair-consistent groups. Separate the IDs with commas (.). This parameter is empty by default, which indicates that all replication pair-consistent groups in the specified region are queried.
Site	String	No	production	The type of the site from which the information of replication pair-consistent groups is queried. This parameter is used for scenarios where data is replicated across zones by using replication pairs. - If the <code>Site</code> parameter is not specified, information such as the state of replication pair-consistent groups at the primary site is queried and returned. - Otherwise, information such as the state of replication pair-consistent groups at the site specified by the <code>Site</code> parameter is queried and returned. Valid values: - production: primary site - backup: secondary site

Response parameters

Parameter	Type	Example	Description
NextToken	String	AAAAAdWBF2****	The query token returned in this call.
ReplicaGroups	Array of ReplicaGroup		The details of replication pair-consistent groups.
Description	String	This is description.	The description of the replication pair-consistent group.
DestinationRegionId	String	cn-shanghai	The ID of the region in which the secondary site is deployed.
DestinationZoneId	String	cn-shanghai-e	The ID of the zone in which the secondary site is deployed.
GroupName	String	myreplicagrouptest	The name of the replication pair-consistent group.
LastRecoverPoint	Long	1637835114	The time when data was last replicated from the primary disks to the secondary disks in the replication pair-consistent group. The value of this parameter is a timestamp. Unit: seconds.
PairIds	List	pair-cn-dsa****,pair-cn-asd****	The IDs of replications pairs that belong to the replication pair-consistent group.
PairNumber	Long	2	The number of replications pairs that belong to the replication pair-consistent group.
RPO	Long	2	The RPO of the replication pair-consistent group. Unit: minutes.
ReplicaGroupId	String	pg-myreplica****	The ID of the replication pair-consistent group.
Site	String	production	The type of the site from which the information of the replication pair and replication pair-consistent group is obtained.
SourceRegionId	String	cn-beijing	The ID of the region in which the primary site is deployed.
SourceZoneId	String	cn-beijing-f	The ID of the zone in which the primary site is deployed.
Status	String	created	The state of the replication pair-consistent group. Valid values: - invalid: The replication pair-consistent group is invalid. This state indicates that specific replication pairs in the replication pair-consistent group are abnormal. - creating: The replication pair-consistent group is being created. - created: The replication pair-consistent group is created. - create_failed: The replication pair-consistent group cannot be created. - manual_syncing: Data is being manually synchronized between the disks in the replication pair-consistent group. When data is being manually synchronized for the first time, the replication pair-consistent group is in this state. - syncing: Data is being synchronized between the disks in the replication pair-consistent group. When data is being asynchronously replicated from the primary disks to the secondary disks not for the first time, the replication pair-consistent group is in this state. - normal: The replication pair-consistent group is working normally. When the system finishes replicating data from the primary disks to the secondary disks within the current replication cycle, the replication pair-consistent group enters this state. - stopping: The replication pair-consistent group is being stopped. - stopped: The replication pair-consistent group is stopped. - stop_failed: The replication pair-consistent group cannot be stopped. - failovering: A failover is being performed. - failovered: A failover is performed. - failover_failed: A failover cannot be performed. - reprotecting: A reverse replication is being performed. - reprotect_failed: A reverse replication cannot be performed. - deleting: The replication pair-consistent group is being deleted. - delete_failed: The replication pair-consistent group cannot be deleted. - deleted: The replication pair-consistent group is deleted.
RequestId	String	AAA478A0-BEE6-1D42-BEB6-A9CFEAD64B29	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DescribeDiskReplicaGroups
&RegionId=cn-beijing
&<Common request parameters>
```

Sample success responses

XML format

```
<ReplicaGroups>
  <Site>production</Site>
  <RPO>2</RPO>
  <Status>created</Status>
  <GroupName>myreplicagrouptest</GroupName>
  <SourceRegionId>cn-beijing</SourceRegionId>
  <PairNumber>2</PairNumber>
  <Description>This is description.</Description>
  <DestinationRegionId>cn-shanghai</DestinationRegionId>
  <DestinationZoneId>cn-shanghai-e</DestinationZoneId>
  <LastRecoverPoint>1637835114</LastRecoverPoint>
  <SourceZoneId>cn-beijing-f</SourceZoneId>
  <ReplicaGroupId>pg-myreplica****</ReplicaGroupId>
  <PairIds>pair-cn-dsa****,pair-cn-asd****</PairIds>
</ReplicaGroups>
<NextToken>AAAAAdWBF2**** </NextToken>
<RequestId>AAA478A0-BEE6-1D42-BEB6-A9CFEAD64B29 </RequestId>
```

JSON format

```
{
  "ReplicaGroups": {
    "Site": "production",
    "RPO": 2,
    "Status": "created",
    "GroupName": "myreplicagrouptest",
    "SourceRegionId": "cn-beijing",
    "PairNumber": 2,
    "Description": "This is description.",
    "DestinationRegionId": "cn-shanghai",
    "DestinationZoneId": "cn-shanghai-e",
    "LastRecoverPoint": 1637835114,
    "SourceZoneId": "cn-beijing-f",
    "ReplicaGroupId": "pg-myreplica****",
    "PairIds": "pair-cn-dsa****,pair-cn-asd****"
  },
  "NextToken": "AAAAAdWBF2****",
  "RequestId": "AAA478A0-BEE6-1D42-BEB6-A9CFEAD64B29"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.

HttpCode	Error code	Error message	Description
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.5. ModifyDiskReplicaGroup

Modifies the name, description, or RPO of a replication pair-consistent group.

Usage notes

The replication pair-consistent group must be in the **created** or **stopped** state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyDiskReplicaGroup	The operation that you want to perform. Set the value to ModifyDiskReplicaGroup.
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group.
ReplicaGroupId	String	Yes	pg-myreplica****	The ID of the replication pair-consistent group.
GroupName	String	No	myreplicagrouptest	The name of the replication pair-consistent group.
Description	String	No	This is description.	The description of the replication pair-consistent group.
RPO	Long	No	120	The RPO of the replication pair-consistent group. Unit: seconds. Valid values: 120 to 604800. Default value: 120. Step size: 60.

Parameter	Type	Required	Example	Description
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ModifyDiskReplicaGroup
&RegionId=cn-beijing
&ReplicaGroupId=pg-myreplica****
&<Common request parameters>
```

Sample success responses

XML **format**

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON **format**

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.

HttpCode	Error code	Error message	Description
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.6. FailoverDiskReplicaGroup

Enables the failover feature for the replication pairs in a replication pair-consistent group. When the primary disks of specific replication pairs in a replication pair-consistent group fail, you can call this operation to enable the read and write permissions on the secondary disks.

Usage notes

- The replication pair-consistent group must be in the **manual_syncing**, **syncing**, **normal**, **stopping**, **stop_failed**, **stopped**, **failovering**, **failover_failed**, or **failovered** state.
- After a failover is performed, the replication pair-consistent group enters the **failovered** state.
- Before you perform a failover, make sure that the first full data synchronization is complete between the primary site and the secondary site.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	FailoverDiskReplicaGroup	The operation that you want to perform. Set the value to <code>FailoverDiskReplicaGroup</code> .
RegionId	String	Yes	cn-shanghai	The region ID of the secondary site of the replication pair-consistent group. Note The failover feature must be enabled in the region where the secondary site is deployed.
ReplicaGroupId	String	Yes	group-myreplica****	The ID of the replication pair-consistent group.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence .

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-ABD6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=FailoverDiskReplicaGroup
&RegionId=cn-shanghai
&ReplicaGroupId=group-myreplica****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.

HttpCode	Error code	Error message	Description
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.7. ReprotectDiskReplicaGroup

Enables the reverse replication feature for the replication pairs in a replication pair-consistent group. After you enable the reverse replication feature, the direction of data replication is inverted. Data is replicated from the original secondary disks to the original primary disks.

Usage notes

- The replication pair-consistent group for which you want to enable reverse replication must be in the **failovered** state. You can call the `FailoverDiskReplicaPair` operation to enable the failover feature.
- Before a reverse replication is performed, the primary disks must be detached from the associated ECS instances. You can call the `DetachDisk` operation to detach disks.
- After you enable the reverse replication feature, you must call the `StartDiskReplicaPairGroup` operation again to activate the asynchronous replication feature before data can be replicated from the original secondary disks to the original primary disks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ReprotectDiskReplicaGroup	The operation that you want to perform. Set the value to <code>ReprotectDiskReplicaGroup</code> .
RegionId	String	Yes	cn-shanghai	The region ID of the secondary site of the replication pair-consistent group. The value of this parameter is also the identifier and a component of Endpoint, as shown in the following sample request.
ReplicaGroupId	String	Yes	pg-myreplica****	The ID of the replication pair-consistent group.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.
SourceRegionId	String	No	cn-beijing	The ID of the region in which the primary site is deployed.
SourceZoneId	String	No	cn-beijing-f	The ID of the zone in which the primary site is deployed.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ReprotectDiskReplicaGroup
&RegionId=cn-shanghai
&ReplicaGroupId=group-myreplica****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.8. RemoveDiskReplicaPair

Removes a replication pair from a replication pair-consistent group. After a replication pair is removed from a replication pair-consistent group, the pair is disassociated from the group but is not deleted.

Usage notes

The replication pair-consistent group from which you want to remove a replication pair must be in the **created**, **stopped**, or **invalid** state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveDiskReplicaPair	The operation that you want to perform. Set the value to RemoveDiskReplicaPair.
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group.
ReplicaGroupId	String	Yes	pg-cn-dsa****	The ID of the replication pair-consistent group.
ReplicaPairId	String	Yes	pair-cn-dsa****	The ID of the replication pair.
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-A8D6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RemoveDiskReplicaPair
&RegionId=cn-beijing
&ReplicaGroupId=pg-cn-dsa****
&ReplicaPairId=pair-cn-dsa****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.

HttpCode	Error code	Error message	Description
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.9. StopDiskReplicaGroup

Disables the asynchronous replication feature for the replication pairs in a replication pair-consistent group.

Usage notes

- The replication pair-consistent group must be in the **manual_syncing**, **syncing**, **normal**, **stopping**, **stop_failed**, or **stopped** state.
- After the asynchronous replication feature is disabled for a replication pair-consistent group, the group enters the **stopped** state. If the asynchronous replication feature cannot be disabled, the state of the group remains unchanged or changes to **stop_failed**. In this case, try again later.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	StopDiskReplicaGroup	The operation that you want to perform. Set the value to StopDiskReplicaGroup.
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group.
ReplicaGroupId	String	Yes	pg-myreplica****	The ID of the replication pair-consistent group.

Parameter	Type	Required	Example	Description
ClientToken	String	No	123e4567-e89b-12d3-a456-426655440000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	A37597A6-BB99-19B3-85EA-4C2B91F00123	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=StopDiskReplicaGroup
&RegionId=cn-beijing
&ReplicaGroupId=pg-myreplica****
&<Common request parameters>
```

Sample success responses

XML **format**

```
<RequestId>A37597A6-BB99-19B3-85EA-4C2B91F00123</RequestId>
```

JSON **format**

```
{
  "RequestId": "A37597A6-BB99-19B3-85EA-4C2B91F00123"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.

HttpCode	Error code	Error message	Description
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.8.10. DeleteDiskReplicaGroup

Deletes a replication pair-consistent group.

Usage notes

- Before you can delete a replication pair-consistent group, make sure that no replication pairs are present in the group.
- The replication pair-consistent group must be in the **created**, **create_failed**, **stopped**, **failovered**, **deleting**, **delete_failed**, or **invalid** state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteDiskReplicaGroup	The operation that you want to perform. Set the value to DeleteDiskReplicaGroup.
RegionId	String	Yes	cn-beijing	The region ID of the replication pair-consistent group. The value of this parameter is also the identifier and a component of Endpoint, as shown in the following sample request.
ReplicaGroupId	String	Yes	pg-myreplica****	The ID of the replication pair-consistent group.
ClientToken	String	No	123e4567-e89b-11d3-a456-426654400000	The client token that is used to ensure the idempotence of the request. You can use the client to generate the value, but you must make sure that the value is unique among different requests. The token is a string of up to 64 ASCII characters. For more information, see How to ensure idempotence.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C123F94F-4E38-19AE-942A-ABD6F44F0450	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteDiskReplicaGroup
&RegionId=cn-beijing
&ReplicaGroupId=pg-myreplica****
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>C123F94F-4E38-19AE-942A-A8D6F44F0450</RequestId>
```

JSON format

```
{
  "RequestId": "C123F94F-4E38-19AE-942A-A8D6F44F0450"
}
```

Error codes

HttpCode	Error code	Error message	Description
404	InvalidRegionId.NotFound	The specified RegionId does not exist.	The error message returned because the specified region ID does not exist.
404	InvalidZoneId.NotFound	The specified ZoneId does not exist.	The error message returned because the specified zone ID does not exist.
404	NoSuchResource	The specified resource does not exist.	The error message returned because the specified resource does not exist.
504	RequestTimeout	The request is timeout, please try again later.	The error message returned because the request timed out. Try again later.
500	InternalServerError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.
403	InvalidPayMethod	The specified pay method is not valid.	The error message returned because the billing method is not supported. Check whether the chargeType parameter value is valid.
403	Forbidden	User is not authorized to operate.	The error message returned because you are not authorized to manage the resource. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	Forbidden.Action	User is not authorized to operate this action.	The error message returned because you are not authorized to perform this operation. Check the permissions of your account or contact the administrator of the Apsara Stack tenant account.
403	NoPermission.SLR	The RAM user does not have the permission to create service linked role: AliyunServiceLinkedRoleForEBS.	The error message returned because you are not authorized to create service-linked roles.
403	LastTokenProcessing	The last token request is processing.	The error message returned because the token is used in another request that is being processed. Try again later.
403	InvalidAccountStatus.NotEnoughBalance	Your account does not have enough balance.	The error message returned because your account balance is insufficient.
403	InvalidDiskCategory.NotSupported	The specified disk category is not supported.	The error message returned because the specified disk type is not supported by this operation. Check the specified value and view the related topics about API operations for reference.
403	OperationDenied	The operation is not allowed.	The error message returned because the operation is not supported.
403	OperationDenied.QuotaExceed	The operation is not allowed due to quota exceed.	The error message returned because the quota for performing this operation has been reached.
403	NotOnSale.Region	The specified region is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the region.
403	NotOnSale.Zone	The specified zone is not on sale.	The error message returned because requested resources are unavailable for purchase or out of stock in the zone.
403	OperationDenied.PairInAnotherGroup	The operation is not allowed because pair is already in another group.	The error message returned because the replication pair has been added to another replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.PairNotInGroup	The operation is not allowed because pair is not in any group.	The error message returned because the specified replication pair has not been added to any replication pair-consistent group and thus the operation is not supported.
403	OperationDenied.NoPairInGroup	The operation is not allowed because no pair in the group.	The error message returned because no replication pairs belong to the specified replication pair-consistent group and thus the operation is not supported.

HttpCode	Error code	Error message	Description
403	OperationDenied.PairsExceedInGroup	The operation is not allowed due to too much pair in the group.	The error message returned because the maximum number of replication pairs in the replication pair-consistent group has been reached and thus the operation is not supported.

3.6.9. Appendix

3.6.9.1. How to ensure idempotence

If a request timeout or an internal server error occurs when you call an EBS API operation, the client may resend the request. In this case, you can set the `ClientToken` parameter in the request to prevent the client from repeatedly sending the request.

What is idempotence

Idempotence is a property of some operations and resources in mathematics and computer science. Idempotence allows you to obtain the result of the initial request after a request is resent multiple times. Idempotence plays an important role in the distributed system design.

How to ensure idempotence

In most cases, the client resends a request only when error 500 (`InternalServerError`) or 503 (`ServiceUnavailable`) is returned or when no response is received. When you use EBS API, the client generates a parameter value that consists of up to 64 ASCII characters, and you can assign the value to `ClientToken` to ensure the idempotence of the resent requests.

ClientToken details

The following section describes the details of `ClientToken`:

- The `ClientToken` value is a unique and case-sensitive string that is generated by a client and can contain up to 64 ASCII characters in length. Example: `ClientToken=123e4567-e89b-12d3-a456-426655440000`.
- If you provide a used value for the `ClientToken` parameter and use different values for other request parameters, the cloud service returns the error code `IdempotencyParameterMismatch`.

Note You must use different values for `SignatureNonce`, `Timestamp`, and `Signature` in each retry. This is because ECS uses `SignatureNonce` to prevent replay attacks and uses `Timestamp` to identify the timestamp of each request. Therefore, you must specify different `SignatureNonce` and `Timestamp` values in a resent request. Consequently, this changes the value of `Signature`.

- After you specify the `ClientToken` parameter in a request, the following results may be returned:
 - If the HTTP status code 200 is returned, the client will receive the same result as the last request after the request is resent. However, your server status will not be affected.
 - If an HTTP status code 4xx is returned without the `try it later` error message, a retry will fail. You must troubleshoot issues based on the returned error message before you resend a request.

Sample idempotent requests

In this example, the `CreateDisk` operation is called to create a cloud disk. If you send multiple requests that use the same `ClientToken` value whereas other request parameters remain unchanged, the server returns identical request results with identical `DiskId` values.

```
https://ecs.example.com/?Action=CreateDisk
&RegionId=cn-qingdao-env17-d01
&ZoneId=cn-qingdao-env17-amtest11001-a
&SnapshotId=s-bp67acfmazb4p****
&DiskName=test-DiskName
&Size=2000
&DiskCategory=cloud_ssd
&Description=This is a description example.
&Encrypted=false
&ClientToken=123e4567-e89b-12d3-a456-426655440000
&Tag.1.Key=TestKey
&Tag.1.Value=TestValue
&<Common request parameters>
```

If you do not use `ClientToken`, multiple cloud disks may be created when the client retries operations after errors occur.

List of operations

The following section lists some API operations that include `ClientToken` for your reference. For information about whether an API operation includes `ClientToken`, see the topic that describes the API operation.

- [CreateDisk](#)
- [ResizeDisk](#)
- [ReplaceSystemDisk](#)
- [CreateSnapshot](#)
- [CreateDiskReplicaPair](#)
- [StartDiskReplicaPair](#)
- [ModifyDiskReplicaPair](#)
- [FailoverDiskReplicaPair](#)
- [ReprotectDiskReplicaPair](#)
- [StopDiskReplicaPair](#)

- [DeleteDiskReplicaPair](#)
- [CreateDiskReplicaGroup](#)
- [AddDiskReplicaPair](#)
- [StartDiskReplicaGroup](#)
- [ModifyDiskReplicaGroup](#)
- [FailoverDiskReplicaGroup](#)
- [ReprotectDiskReplicaGroup](#)
- [RemoveDiskReplicaPair](#)
- [StopDiskReplicaGroup](#)
- [DeleteDiskReplicaGroup](#)

3.6.9.2. API operations and states related to a replication pair

This topic describes the API operations and states related to a replication pair.

API operations

API operation	State		
	Before the operation	If the operation succeeds	If the operation fails
CreateDiskReplicaPair	The replication pair does not exist.	created	The operation returns an error, or the replication pair enters the <code>create_failed</code> state.
StartDiskReplicaPair (Automatic synchronization)	created, <code>initial_syncing</code> , or stopped	The replication pair is in the <code>initial_syncing</code> state during the first synchronization. Otherwise, the replication pair is in the <code>syncing</code> state. After the synchronization of the current cycle is complete, the replication pair enters the <code>normal</code> state.	The state remains unchanged.
	<code>syncing</code> or <code>normal</code>	The state remains unchanged.	The state remains unchanged.
StartDiskReplicaPair (Manual synchronization)	created, <code>manual_syncing</code> , or stopped	<code>manual_syncing</code>	The state remains unchanged.
StopDiskReplicaPair	<code>syncing</code> , <code>init_syncing</code> , <code>manual_syncing</code> , <code>normal</code> , <code>stopping</code> , <code>stop_failed</code> , or stopped	stopped	The state remains unchanged or changes to <code>stop_failed</code> .
FailoverDiskReplicaPair	<code>manual_syncing</code> , <code>syncing</code> , <code>normal</code> , <code>stopping</code> , <code>stop_failed</code> , stopped, <code>failovering</code> , <code>failover_failed</code> , or <code>failovered</code>	failovered	<code>failover_failed</code>
ReprotectDiskReplicaPair	<code>failovered</code> , <code>reprotect_failed</code> , or <code>reprotecting</code>	stopped	<code>reprotect_failed</code>
DeleteDiskReplicaPair	created, stopped, invalid, deleted, <code>delete_failed</code> , or deleting	The replication pair does not exist.	<code>delete_failed</code>
ModifyDiskReplicaPair	created or stopped	The state remains unchanged.	The state remains unchanged.

States

State	Name	Description
creating	Creating	The replication pair is being created.
created	Created	The replication pair is created.
<code>create_failed</code>	Creation Failed	The replication pair cannot be created. We recommend that you delete the replication pair and call the CreateDiskReplicaPair operation again.
normal	Normal	The synchronization of the current cycle is complete.
<code>initial_syncing</code>	Initial Syncing	Data is being synchronized for the first time.
<code>manual_syncing</code>	One-time Syncing	After data is manually synchronized from the primary disk to the secondary disk, the replication pair returns to the stopped state. If data is manually synchronized from the primary disk to the secondary disk for the first time, the replication pair is in the <code>manual_syncing</code> state during the synchronization.
<code>syncing</code>	Syncing	Data is being synchronized.
invalid	Invalid	The replication pair becomes abnormal due to a user operation. We recommend that you troubleshoot related issues based on the status message.
stopping	Stopping	The replication pair is being stopped.
stopped	Stopped	The replication pair is stopped.

State	Name	Description
stop_failed	Stop Failed	The replication pair cannot be stopped. We recommend that you call the StopDiskReplicaPair operation again.
deleting	Deleting	The replication pair is being deleted.
delete_failed	Deletion Failed	The replication pair cannot be deleted. We recommend that you call the DeleteDiskReplicaPair operation again.
deleted	Deleted	The replication pair is deleted. This state is invisible.
failovering	In Failover	The failover is being performed.
failover_failed	Failover Failed	The failover cannot be performed. We recommend that you call the FailoverDiskReplicaPair operation again.
failovered	Failed Over	The failover is complete.
reprotecting	In Reverse Replication	The reverse replication is being performed.
reprotect_failed	Reverse Replication Failed	The reverse replication cannot be performed. We recommend that you call the ReprotectDiskReplicaPair operation again.

3.6.9.3. API operations and states related to a replication pair-consistent group

This topic describes the API operations and states related to a replication pair-consistent group.

API operations

API operation	State		
	Before the operation	If the operation succeeds	If the operation fails
CreateDiskReplicaGroup	The replication pair-consistent group does not exist.	created	The operation returns an error, or the replication pair enters the <code>create_failed</code> state.
AddDiskReplicaPair	created or stopped	The state remains unchanged.	The state remains unchanged.
StartDiskReplicaGroup (Automatic synchronization)	created or stopped	syncing After the synchronization of the current cycle is complete, the replication pair-consistent group enters the normal state.	The state remains unchanged.
	syncing or normal	The state remains unchanged.	The state remains unchanged.
StartDiskReplicaGroup (Manual synchronization)	created, manual_syncing, or stopped	manual_syncing	The state remains unchanged.
StopDiskReplicaGroup	syncing, manual_syncing, normal, stopping, stop_failed, or stopped	stopped	The state remains unchanged or changes to <code>stop_failed</code> .
FailoverDiskReplicaGroup	A state other than invalid or deleted	failovered	failover_failed
ReprotectDiskReplicaGroup	failovered, reprotect_failed, or reprotecting	stopped	reprotect_failed
RemoveDiskReplicaPair	created, stopped, or invalid	The state remains unchanged.	The state remains unchanged.
DeleteDiskReplicaGroup	created, stopped, invalid, deleted, delete_failed, or deleting	The replication pair-consistent group does not exist.	delete_failed
ModifyDiskReplicaGroup	created or stopped	The state remains unchanged.	The state remains unchanged.

States

State	Name	Description
creating	Creating	The replication pair-consistent group is being created.
created	Created	The replication pair-consistent group is created.
create_failed	Creation Failed	The replication pair-consistent group cannot be created. We recommend that you delete the replication pair-consistent group and call the CreateDiskReplicaGroup operation again.
normal	Normal	The synchronization of the current cycle is complete.

State	Name	Description
manual_syncing	One-time Syncing	After data is manually synchronized from the primary disks to the secondary disks, the replication pair-consistent group returns to the stopped state. If data is manually synchronized from the primary disks to the secondary disks for the first time, the replication pair-consistent group is in the manual_syncing state during the synchronization.
syncing	Syncing	Data is being synchronized.
invalid	Invalid	The replication pair-consistent group becomes abnormal due to a user operation. We recommend that you troubleshoot related issues based on the status message.
stopping	Stopping	The replication pair-consistent group is being stopped.
stopped	Stopped	The replication pair-consistent group is stopped.
stop_failed	Stop Failed	The replication pair-consistent group cannot be stopped. We recommend that you call the StopDiskReplicaGroup operation again.
deleting	Deleting	The replication pair-consistent group is being deleted.
delete_failed	Deletion Failed	The replication pair-consistent group cannot be deleted. We recommend that you call the DeleteDiskReplicaGroup operation again. Before you can delete a replication pair-consistent group, make sure that no replication pairs are present in the group.
deleted	Deleted	The replication pair-consistent group is deleted. This state is not visible.
failovering	In Failover	The failover is being performed.
failover_failed	Failover Failed	The failover cannot be performed. We recommend that you call the FailoverDiskReplicaGroup operation again.
failovered	Failed Over	The failover is complete.
reprotecting	In Reverse Replication	The reverse replication is being performed.
reprotect_failed	Reverse Replication Failed	The reverse replication cannot be performed. We recommend that you call the ReprotectDiskReplicaGroup operation again.

4. Log Service

4.1. Preparations

4.1.1. Log on to the API Tool console

You can log on to the API Tool console from the Apsara Uni-manager Management Console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment personnel.
- A browser is available. We recommend that you use Google Chrome.

Procedure

1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press Enter.
2. Enter your username and password.

Obtain the username and password that are used to log on to the Apsara Uni-manager Management Console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your account. For security purposes, your password must meet the minimum complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA verification code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Others > API Tool**.

4.1.2. Obtain API information

To call API operations by using an SDK, you must first obtain information about the service name, API operation name, and API version.

Context

The information on the API Directory page of the Apsara Uni-manager Management Console is for reference only. For more information about how to call API operations, see the developer guides of cloud services. If the information on the API Directory page is different from that in the developer guide of a cloud service, the information in the developer guide prevails.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Products** and choose **Others > API Tool**.
3. In the left-side navigation pane, click **API Directory**.
4. On the **API Directory** page, select the required cloud service from the Product Name drop-down list, enter a keyword of an operation name in the search box, and then click Search.
5. In the search result, you can view the **cloud service name**, **API operation name**, and **API version**.

4.1.3. Obtain an AccessKey pair

AccessKey pairs are classified into AccessKey pairs assigned by Resource Access Management (RAM) and AccessKey pairs assigned by Security Token Service (STS). You can use either type of AccessKey pair for authentication when you call operations. This topic describes how to obtain an AccessKey pair that is assigned by RAM.

Context

An AccessKey pair is used to verify the identity of request senders. An AccessKey pair consists of an AccessKey ID and an AccessKey secret that are used to implement symmetric encryption. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string.

You can use RAM or STS to assign an AccessKey pair for authentication.

- Use RAM to assign an AccessKey ID and an AccessKey secret to a third party.
- Use STS to assign an AccessKey ID, an AccessKey secret, and a security token to a third party.

The Apsara Uni-manager Management Console provides AccessKey pairs for personal accounts or organizations. Only the administrators of level-1 organizations and operations administrators can obtain the AccessKey pairs of organizations. We recommend that you use the AccessKey pairs of personal accounts to call the operations provided by the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, you must add the following parameters in request headers for access control. If you do not add the parameters, an error may occur. The error indicates that permissions are insufficient.

 **Warning** The AccessKey pair of a personal account is managed by the permission system of the Apsara Uni-manager Management Console. The AccessKey pair of an organization has more permissions than the AccessKey pair of a personal account. If you want to obtain the AccessKey pair of an organization, make sure that you are authorized by an administrator. This helps ensure security.

Parameter	Description
x-acs-regionid	The ID of the region. Example: cn-hangzhou-*
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain organization IDs, see the <i>GetOrganizationList</i> section in <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain resource set IDs, see the <i>ListResourceGroup</i> section in <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-instanceid	The ID of the instance on which you want to perform operations. For example, you can obtain the ID of an Elastic Compute Service (ECS) instance from the ECS instance list.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following operations:

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **AccessKey Pair** section, view the AccessKey pair of your account.

 **Note** An AccessKey pair consists of an AccessKey ID and an AccessKey secret. These credentials provide full permissions on cloud resources within your account. You must keep the AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following operations:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization list, click the name of the level-1 organization whose AccessKey pair you want to obtain.
5. Click **Management Accesskey**.
6. In the dialog box that appears, view the AccessKey pair of the organization.

4.1.4. Obtain the values of common headers

You must provide multiple headers when you call Apsara Stack operations. This topic describes the headers and how to obtain the values of the headers. This topic also describes the return values of some headers after you call Apsara Stack operations.

Headers

Name	Description
x-acs-regionid	The ID of the region. For more information about how to obtain region IDs, see Obtain a region ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain organization IDs, see Obtain an organization ID .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain resource set IDs, see Obtain a resource set ID .
x-acs-instanceid	The ID of the instance on which you want to perform operations. For more information about how to obtain instance IDs, see Obtain an instance ID .

Obtain a region ID

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Regions**.
4. Click the name of the required organization.
5. View the region ID in the **Regions** section.

Obtain an organization ID

You must call an operation to obtain an organization ID. For more information about how to obtain organization IDs, see the *GetOrganizationList* section in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a resource set ID

You must call an operation to obtain a resource set ID. For more information about how to obtain resource set IDs, see the *ListResourceGroup* section in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a project name

To obtain a project name, perform the following operations in the Apsara Uni-manager Management Console:

1. Log on to the Log Service console.
For more information, see the *Log on to the Log Service console* section in *Log Service User Guide*.
2. In the Projects section, obtain the **project name**.

Response parameters

The following response parameters are returned by default when you call operations to query the instances of cloud services.

Name	Description
ResourceGroupName	The name of the resource set.
ResourceGroup	The ID of the resource set.
DepartmentName	The name of the organization.
Department	The ID of the organization.

4.1.5. Obtain the endpoint of Log Service

This topic describes how to obtain the endpoint of Log Service.

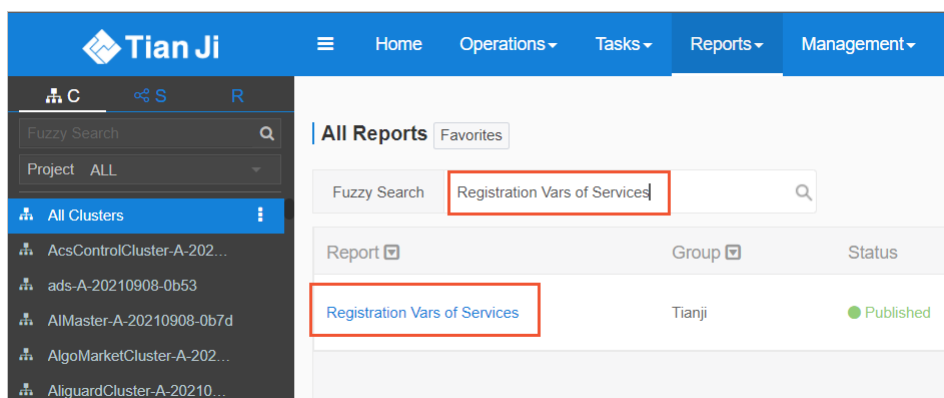
Background information

When you call Apsara Stack operations, endpoints are used. You can obtain an endpoint from onsite Alibaba Cloud O&M engineers. You can alternatively perform the following operations to obtain an endpoint.

Procedure

Log on to the Apsara Infrastructure Management console.

1. Log on to the Apsara Uni-manager Operations Console.
2. In the top navigation bar, click **Products**. Then, choose **Platforms > Apsara Infrastructure Management Framework**.
3. In the left-side navigation pane, click **Reports**.
4. On the **All Reports** page, search for `Registration Vars of Services`. In the report list that appears, click the name of the report.



5. On the **Registration Vars of Services** page, click the icon next to **Service** and search for `sls`.
6. Find `sls-backend-server`. Right-click the value in the **Service Registration** column and select **Show More**.

On the **Details** page, view the value of `sls_data.endpoint`. The value indicates the `endpoint` of Log Service.

4.1.6. Obtain a Log Service SDK

When you call Apsara Stack operations by using the point of presence (POP) gateway, you must use the SDK of your cloud service. This topic describes how to obtain Alibaba Cloud SDKs.

An SDK includes code and examples. You can create cloud applications by using an SDK in the programming language that you require.

The SDKs for Java, Python, and Go are provided. You can obtain the SDKs from [Alibaba Cloud SDKs](#) based on your business requirements.

Note

We recommend that you use an SDK for Java, Python, or Go. If you use SDKs in other programming languages, the SDKs may be incompatible with your system.

We recommend that you use the SDK core library that is provided on the Alibaba Cloud official website. The SDK core library is shared by all cloud services and is compatible with Apsara Stack.

The following table provides GitHub links to the source code of Log Service SDKs for common programming languages.

SDK language	Source code on GitHub
SDK for Java	Log Service SDK for Java
SDK for Python	Log Service SDK for Python
SDK for Go	Log Service SDK for Go
SDK for PHP	Log Service SDK for PHP

4.2. SDK Reference

4.2.1. Introduction

4.2.1.1. Overview

Log Service provide software development kits (SDKs) for multiple languages, such as Java, .NET, Python, PHP, and C. You can select an SDK based on your business requirements.

Log Service SDKs are implemented based on Log Service APIs. Log Service SDKs provide the same capabilities as Log Service APIs. For more information about Log Service APIs, see [API Reference](#).

Before you can use a Log Service SDK, you must have an active Alibaba Cloud AccessKey pair (AccessKey ID and AccessKey secret).

In addition, you must specify the endpoint of Log Service in the SDK. For more information, see [Basic configurations](#).

Log Service SDKs can be regarded as Log Service APIs that are encapsulated in different languages to provide the following features:

- Uniform encapsulation. Log Service APIs are uniformly encapsulated. You do not need to create API requests and parse API responses. For more information, see [API specifications](#). The APIs in different languages are similar. This simplifies the switchover between different languages.
- Automatic digital signatures. You do not need to specify the digital signature logic of Log Service APIs. This simplifies the use of Log Service APIs.
- Protocol Buffer-formatted encapsulation. The logs collected by Log Service are encapsulated in the Protocol Buffer format. You do not need to specify the format.
- Automatic compression. Logs are compressed by using the method defined in Log Service APIs. You do not need to manually compress logs. SDKs in some languages allow you to specify whether to write logs in the compression mode. The compression mode is used by default.
- Centralized error handling methods. You can use SDKs to handle errors based on the relevant language. For more information, see [Error handling](#).
- Synchronous requests. SDKs in all languages support only synchronous requests.

The following table describes the download address and user guide of Log Service SDKs in different languages.

SDK language	Documentation	Source code
Java	SDK for Java and Log Service SDK for Java 0.6.0 API	GitHub
.NET	SDK for .NET and Aliyun.Api.SLS Namespace	GitHub
PHP	SDK for PHP and SLS_PHP_SDK	GitHub
Node.js	SDK for Node.js	GitHub
Python	SDK for Python and Python documentation	GitHub
C	SDK for C	GitHub
GO	SDK for Go	GitHub
iOS	SDK for iOS	GitHub
Android	SDK for Android	GitHub
C++	SDK for C++	GitHub

4.2.1.2. Basic configurations

Before you can use SDKs to interact with Log Service, you must configure the basic settings. This topic describes the basic SDK configurations.

Basic configurations:

- Endpoint: You must specify the endpoint that is used by the client to access Log Service.
- AccessKey pair (AccessKey ID and AccessKey secret): You must specify the AccessKey pair that is used by the client to access Log Service.

The following sections describe how to use endpoints and AccessKey pairs.

Endpoint

Before you use an SDK, you must determine the access method and endpoint.

1. When you select an endpoint for a client instance, make sure that the region of your project is the same as the region of your endpoint. Otherwise, you cannot use the SDK to access your project.
2. All API endpoints support only the HTTP protocol.

AccessKey pairs

All interaction requests that are sent to Log Service must be authenticated. An AccessKey pair that consists of an AccessKey ID and AccessKey secret is the critical factor for the authentication. When you create a client instance, you must specify an AccessKey ID and AccessKey secret. You must obtain an AccessKey pair in the Apsara Uni-manager Management Console.

Note

- An Apsara Stack tenant account can have multiple AccessKey pairs. You must specify the AccessKey ID and AccessKey secret of the same AccessKey pair when you create a client instance. Otherwise, authentication fails.
- The specified AccessKey must be in the Enabled state. Otherwise, the requests are denied by Log Service. To view the status of the AccessKey pair, you can log on to the Apsara Uni-manager Management Console.

Examples

Assume that you want to access a project and you have an AccessKey pair that is in the Enabled state. The following AccessKey pair is used:

```
AccessKeyId = "<yourAccessKeyId>"
AccessKeySecret = "<yourAccessKeySecret>"
```

You can use the following snippets to create a client instance based on the language:

• Java

```
String endpoint = "regionid.example.com"; // Specify the endpoint based on your region and access method.
String accessKeyId = "<youraccessKeyId>"; // Specify your AccessKey ID.
String accessKeySecret = "<youraccessKeySecret>"; // Specify your AccessKey secret.
Client client = new Client(endpoint, accessKeyId, accessKeySecret);
//use client to operate log service project.....
```

• .NET(C#)

```
String endpoint = "regionid.example.com"; // Specify the endpoint based on your region and access method.
String accessKeyId = "<youraccessKeyId>"; // Specify your AccessKey ID.
String accessKeySecret = "<youraccessKeySecret>"; // Specify your AccessKey secret.
SLSCClient client = new SLSCClient(endpoint, accessKeyId, accessKeySecret);
//use client to operate sls project.....
```

• PHP

```
$endpoint = 'regionid.example.com'; // The endpoint of the China (Hangzhou) region is used as an example. Replace it with the actual endpoint based
on your region and access method.
$accessKeyId = '<youraccessKeyId>'; // Specify your AccessKey ID.
$accessKey = '<youraccessKey>'; // Specify your AccessKey secret.
$client = new Aliyun_Sls_Client($endpoint, $accessKeyId, $accessKey);
//use client to operate sls project.....
```

• Python

```
# // Specify the endpoint based on your region and access method.
endpoint = 'regionid.example.com'
# Specify your AccessKey ID.
accessKeyId = '<youraccessKeyId>'
# Specify your AccessKey secret.
accessKey = '<youraccessKey>'
client = LogClient(endpoint, accessKeyId, accessKey)
#use client to operate log project.....
```

4.2.1.3. Error handling

This topic describes the basic error handling.

SDK errors are classified into the following types:

- Errors returned by Log Service. This type of errors are handled by SDKs. For more information about this type of errors, see the common error codes of the Log Service API operations.
- Network errors that occur when you use SDKs to send requests to the server. This type of errors include network disconnection and server response timeout.
- Errors that are generated by SDKs and are related to platforms or programming languages, for example, memory overflows.

All SDKs that are provided by Alibaba Cloud handle errors by throwing exceptions based on the following rules:

- If the first or second types of errors occur, the SDKs encapsulate the errors as the LogException class and then send the class to users.
- If the third type of errors occur, the SDKs do not handle the errors but send the errors as the platform-specific and language-specific NativeException class to users.

LogException

The LogException class is defined by SDKs to handle the logical errors of Log Service. This class inherits the base class for all exceptions from each language, and provides the following exception information:

- Error code: indicates the error type. For an error returned by Log Service when you use an SDK, the error code is the same as the error code that is returned by the

relevant API operation. The error code for network request errors of SDKs is `RequestError`. For more information, see [API Reference](#).

- **Error message:** indicates the message that is returned together with an error. For an error returned by Log Service when you use an SDK, the error message is the same as the error code that is returned by the relevant API operation. The error message of network errors is "request is failed". For more information, see [API Reference](#).
- **Request ID:** indicates the ID of the request for which the current error is returned. The request ID is generated by Log Service. The request ID is returned only when Log Service returns an error message. If Log Service does not return an error message, an empty string is returned. If you receive a request error, you can provide the relevant request ID to the Log Service team for troubleshooting.

Request failures and retries

If you use an SDK to access Log Service, the access request may fail due to temporary network disconnection, transmission delay, or slow server response. These errors are thrown as exceptions. Log Service does not implement retry. When you use the SDK, you must customize the handling logic, for example, resend the request or report an error.

Examples

For example, you want to access a project named **big-game** and resend the access request for a specified number of times if network exceptions occur. You can use the following snippets based on the language:

- Java

```
// Other code...
String accessId = "<youraccessId>"; // TODO: Specify the AccessKey ID of your Apsara Stack tenant account.
String accessKey = "<youraccessKey>"; // TODO: Specify the AccessKey secret of your Apsara Stack tenant account.
String project = "big-game";
String endpoint = "regionid.example.com";
int max_retries = 3;
/*
 * Create a client.
 */
Client client = new Client(accessId, accessKey, endpoint);
ListLogStoresRequest lsRequest = new ListLogStoresRequest(project);
for (int i = 0; i < max_retries; i++)
{
    try
    {
        ListLogStoresResponse res = client.ListLogStores(lsRequest);
        //TODO: Process the returned data...
        break;
    }
    catch(LogException ex)
    {
        if (e.GetErrorCode() == "RequestError")
        {
            if (i == max_retries - 1)
            {
                System.out.println("request is still failed after all retries.");
                break;
            }
            else
            {
                System.out.println("request error happens, retry it!");
            }
        }
        else
        {
            System.out.println("error code :" + e.GetErrorCode());
            System.out.println("error message :" + e.GetErrorMessage());
            System.out.println("error requestId :" + e.GetRequestId());
            break;
        }
    }
    catch(...)
    {
        System.out.println("unrecoverable exception when listing logstores.");
        break;
    }
}
// Other code...
```

- .NET(C#)

```
// Other code...
String accessId = "<youraccessId>"; // TODO: Specify the AccessKey ID of your Apsara Stack tenant account.
String accessKey = "<youraccessKey>"; // TODO: Specify the AccessKey secret of your Apsara Stack tenant account.
String project = "big-game";
String endpoint = "regionid.example.com";
int max_retries = 3;
// Create a client.
SLSClient client = new SLSClient(endpoint, accessId, accessKey);
ListLogstoresRequest request = new ListLogstoresRequest();
request.Project = project;
for (int i = 0; i < max_retries; i++)
{
    try
    {
        ListLogstoresResponse response = client.ListLogstores(request);
        //TODO: Process the returned data...
        break;
    }
    catch(LogException ex)
    {
        if (e.errorCode == "SLSRequestError")
        {
            if ( i == max_retries - 1)
            {
                Console.WriteLine("request is still failed after all retries.");
                break;
            }
            else
            {
                Console.WriteLine("request error happens, retry it!");
            }
        }
        else
        {
            Console.WriteLine("error code :" + e.errorCode;
            Console.WriteLine("error message :" + e.Message;
            Console.WriteLine("error requestId :" + e.RequestId;
            break;
        }
    }
}
catch(...)
{
    Console.WriteLine("unrecoverable exception when listing logstores.");
    break;
}
}
// Other code...
```

- PHP

```

<?php
// Other code...
$endpoint = 'regionid.example.com';
$accessId = '<youraccessId>'; // TODO: Specify the AccessKey ID of your Apsara Stack tenant account.
$accessKey = '<youraccessKey>'; // TODO: Specify the AccessKey secret of your Apsara Stack tenant account.
$maxRetries = 3;
// Create a Log Service client.
$client = new Aliyun_Sls_Client($endpoint, $accessId, $accessKey);
$project = 'big-game';
$request = new Aliyun_Sls_Models_ListLogstoresRequest($project);
for($i = 0; $i < $maxRetries; ++$i)
{
    try
    {
        $response = $client->ListLogstores($request);
        //TODO: Process the returned data...
        break;
    }
    catch (Aliyun_Sls_Exception $e)
    {
        if ($e->getErrorCode()=='RequestError')
        {
            if ($i+1 == $maxRetries)
            {
                echo "error code :" . $e->getErrorCode() . PHP_EOL;
                echo "error message :" . $e->getErrorMessage() . PHP_EOL;
                break;
            }
            echo 'request error happens, retry it!' . PHP_EOL;
        }
        else
        {
            echo "error code :" . $e->getErrorCode() . PHP_EOL;
            echo "error message :" . $e->getErrorMessage() . PHP_EOL;
            echo "error requestId :" . $e->getRequestId() . PHP_EOL;
            break;
        }
    }
    catch (Exception $ex)
    {
        echo 'unrecoverable exception when listing logstores.' . PHP_EOL;
        var_dump($ex);
        break;
    }
}
// Other code...

```

- Python

```

// Other code...
endpoint = 'regionid.example.com'
accessId = '<youraccessId>' # TODO: Specify the AccessKey ID of your Apsara Stack tenant account.
accessKey = '<youraccessKey>' # TODO: Specify the AccessKey secret of your Apsara Stack tenant account.
maxRetries = 3
# Create a client.
client = Client(endpoint, accessId, accessKey)
project = 'big-game'
lsRequest = ListLogstoresRequest(project)
for i in xrange(maxRetries):
    try:
        res = client.ListLogstores(lsRequest)
        # TODO: Process the returned data...
        break
    except LogException as e:
        if e.getErrorCode() == "RequestError":
            if i+1 == maxRetries:
                print "error code :" + e.getErrorCode()
                print "error message :" + e.getErrorMessage()
                break
            else:
                print "request error happens, retry it!"
        else:
            print "error code :" + e.getErrorCode()
            print "error message :" + e.getErrorMessage()
            print "error requestId :" + e.getRequestId()
            break
    except Exception as e:
        print 'unrecoverable exception when listing logstores.'
        break
// Other code...

```

4.2.1.4. API specifications

SDKs in different languages are implemented in different methods. However, all relevant API operations comply with the request-response principle. These API operations are called by using the following procedure:

1. Create a request instance by using request parameters.
2. Call the corresponding API operation in the SDK and pass the API information to the request instance.
3. Encapsulate the results returned by the SDK interface into a response instance and then return the response instance to the user.

The following snippets show how to obtain the names of all Logstores in a project based on the preceding steps.

Java

```
// Other code...
String accessId = "<youraccessId>";    // Specify your AccessKey ID.
String accessKey = "<youraccessKey>";    // Specify your AccessKey secret.
String project = "your_project";        // Specify your project name.
String endpoint = "region_endpoint";    // The China (Hangzhou) region is used as an example. Specify the endpoint based on your region and access method
.
// Create a client instance.
Client client = new Client(endpoint, accessId, accessKey);
// Specify the project request parameter to initialize the ListLogstores request class.
ListLogStoresRequest lsRequest = new ListLogStoresRequest(project);
// Use the request instance to call the ListLogstores API operation. The returned parameter is the corresponding response instance.
ListLogStoresResponse res = client.ListLogStores(lsRequest);
// Access the response instance to retrieve the request results.
ArrayList<String> names = res.GetLogStores();
// Other code...
```

.NET(C#)

```
// Other code...
String accessId = "<youraccessId>";    // Specify your AccessKey ID.
String accessKey = "<youraccessKey>";    // Specify your AccessKey secret.
String project = "your_project";        // Specify your project name.
String endpoint = "region_endpoint";    // The China (Hangzhou) region is used as an example. Specify the endpoint based on your region and access method
.
// Create a client instance.
SLSClient client = new SLSClient(endpoint, accessId, accessKey);
// Specify the project request parameter to initialize the ListLogstores request class.
ListLogStoresRequest lsRequest = new ListLogStoresRequest();
lsRequest.Project = project;
// Use the request instance to call the ListLogstores API operation. The returned parameter is the corresponding response instance.
ListLogStoresResponse res = client.ListLogStores(lsRequest);
// Access the response instance to retrieve the request results.
List<String> names = res.Logstores;
// Other code...
```

PHP

```
// Other code...
$accessId = "<youraccessId>";    // Specify your AccessKey ID.
$accessKey = "<youraccessKey>";    // Specify your AccessKey secret.
$project = "your_project";        // Specify your project name.
$endpoint = "region_endpoint";    // The China (Hangzhou) region is used as an example. Specify the endpoint based on your region and access method.
//Create a Log Service client instance.
$client = new Aliyun_Sls_Client($endpoint, $accessId, $accessKey);
// Specify the project request parameter to initialize the ListLogstores request class.
$request = new Aliyun_Sls_Models_ListLogstoresRequest($project);
// Use the request instance to call the ListLogstores API operation. The returned parameter is the corresponding response instance.
$response = $client->listLogstores($request);
// Access the response instance to retrieve the request results.
$names = $response->getLogstores();
// Other code...
```

Python

```
// Other code...
accessId = '<youraccessId>';    // Specify your AccessKey ID.
accessKey = '<youraccessKey>';    // Specify your AccessKey secret.
project = 'your_project';        // Specify your project name.
endpoint = 'region_endpoint';    // The China (Hangzhou) region is used as an example. Specify the endpoint based on your region and access method.
# Create a client instance.
client = LogClient(endpoint, accessId, accessKey)
# Specify the project request parameter to initialize the ListLogstores request class.
lsRequest = ListLogstoresRequest(project)
# Use the request instance to call the ListLogstores API operation. The returned parameter is the corresponding response instance.
res = client.list_logstores(lsRequest)
# Access the response instance to retrieve the request results.
names = res.get_logstores();
// Other code...
```

The Log Service SDKs implement multiple API operations such as ListLogStores, and define the corresponding request and response classes. The Log Service SDKs of different languages also provide secondary APIs encapsulated with these basic request-response APIs. You do not need to create request instances or parse the content of the response. For more information about the secondary interfaces, see the API reference of each SDK.

4.2.2. Java SDK

Log Service SDK for Java allows Java developers to perform operations in Log Service by using Java programs. You can use Maven dependencies to add the SDK. You can also download the SDK package to an on-premises server.

Download URL

Log Service SDK for Java supports J2SE V6.0 and later. Click [here](#) to download the latest SDK.

Step 1: Create an Apsara Stack tenant account

To access Log Service, you must create an account in the Apsara Uni-manager Management Console.

Step 2: Obtain an Apsara Stack AccessKey pair

To use Log Service SDK for Java, you must obtain an AccessKey pair.

Log on to the Apsara Uni-manager Management Console. Select an AccessKey pair for the SDK.

The AccessKey pair will be used in the following steps and must be kept confidential. For information about how to use an AccessKey pair in Log Service SDK for Java, see [Basic configurations](#).

Step 3: Create a project and a Logstore in the Log Service console

Use Log Service SDK for Java to create a project and a Logstore.

Note

- The account that you use to create the project and the Logstore must be the same as the account that you used to obtain the AccessKey pair.
- The project name must be unique in Log Service. The Logstore name must be unique in the project.
- After you create the project, you cannot modify its region or migrate the project across regions.

Step 4: Install a Java development environment

Log Service SDK for Java supports Java development environment J2SE V6.0 and later. You can visit the [Java official website](#) to download and install the Java development environment.

Step 5: Install Log Service SDK for Java

After you install the Java development environment, you must install Log Service SDK for Java. To install Log Service SDK for Java, use one of the following methods:

Download the latest SDK for Java package from [Apache Maven](#). Add the following configurations to your Maven project:

```
<dependency>
  <groupId>com.google.protobuf</groupId>
  <artifactId>protobuf-java</artifactId>
  <version>2.5.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>aliyun-log</artifactId>
  <version>0.6.7</version>
  <exclusions>
    <exclusion>
      <groupId>com.google.protobuf</groupId>
      <artifactId>protobuf-java</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

Download Log Service SDK for Java package, and then reference the local package in your Java project.

1. Download Log Service SDK for Java package. Log Service SDK for Java is periodically updated. You can use Maven to obtain the latest version.
2. Decompress the downloaded package to a specified directory. Log Service SDK for Java does not require installation.
3. Add all JAR packages (including third-party dependent packages) in the SDK package to your Java project. For more information, see the relevant documents about integrated development environment (IDE).

Step 6: Start a new Java project

The following sample code shows how to use Log Service SDK for Java. To interact with Log Service and obtain the relevant output, run the following sample code in a text editor or Java IDE:

```
package sdksample;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import java.util.Date;
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.*;
import com.aliyun.openservices.log.exception.*;
import com.aliyun.openservices.log.request.*;
import com.aliyun.openservices.log.response.*;
import com.aliyun.openservices.log.common.LogGroupData;
import com.aliyun.openservices.log.common.LogItem;
import com.aliyun.openservices.log.common.Logs.Log;
import com.aliyun.openservices.log.common.Logs.Log.Content;
import com.aliyun.openservices.log.common.Logs.LogGroup;
import com.aliyun.openservices.log.common.Consts.CursorMode;
public class sdksample {
    public static void main(String args[]) throws LogException, InterruptedException {
        // ...
    }
}
```

```

String endpoint = "<log_service_endpoint>"; // Select the endpoint that matches the region or the project created in the preceding steps.
// Endpoint
String accessKeyId = "<youraccessKeyId>"; // Use your AccessKey ID.
String accessKeySecret = "<youraccessKeySecret>"; // Use your AccessKey secret.
// AccessKeySecret
String project = "<project_name>"; // The name of the project created in the preceding steps.
String logstore = "<logstore_name>"; // The name of the Logstore created in the preceding steps.
// Create a client instance.
Client client = new Client(endpoint, accessKeyId, accessKeySecret);
// List all Logstores in the current project.
int offset = 0;
int size = 100;
String logStoreSubName = "";
ListLogStoresRequest req1 = new ListLogStoresRequest(project, offset, size, logStoreSubName);
ArrayList<String> logStores = client.ListLogStores(req1).GetLogStores();
System.out.println("ListLogs:" + logStores.toString() + "\n");
// Write logs.
String topic = "";
String source = "";
// Send 10 packets. Each packet contains 10 log entries.
for (int i = 0; i < 10; i++) {
    Vector<LogItem> logGroup = new Vector<LogItem>();
    for (int j = 0; j < 10; j++) {
        LogItem logItem = new LogItem((int) (new Date().getTime() / 1000));
        logItem.PushBack("index"+String.valueOf(j), String.valueOf(i * 10 + j));
        logGroup.add(logItem);
    }
    PutLogsRequest req2 = new PutLogsRequest(project, logstore, topic, source, logGroup);
    client.PutLogs(req2);
    /*
     * You can specify a shard to which data is sent.
     * After you set a hash key, data is written to the shard whose MD5 value range includes the hash key. For more information, see the API
operation PutLogsResponse.
     * PutLogs( String project, String logStore, String topic,
     * List<LogItem> logItems, String source, String shardHash //
     * Write data to the shard based on the hash key, which can be MD5(ip) or MD5(id).
     */
}
// Read the data that was written to shard 0 in the last minute.
int shard_id = 0;
long curTimeInSec = System.currentTimeMillis() / 1000;
GetCursorResponse cursorRes = client.GetCursor(project, logstore, shard_id, curTimeInSec - 60);
String beginCursor = cursorRes.GetCursor();
cursorRes = client.GetCursor(project, logstore, shard_id, CursorMode.END);
String endCursor = cursorRes.GetCursor();
String curCursor = beginCursor;
while (curCursor.equals(endCursor) == false) {
    int loggroup_count = 2; // Read two log groups at a time.
    BatchGetLogResponse logDataRes = client.BatchGetLog(project, logstore, shard_id, loggroup_count, curCursor,
        endCursor);
    // Read the log group list.
    List<LogGroupData> logGroups = logDataRes.GetLogGroups();
    for (LogGroupData logGroup: logGroups) {
        FastLogGroup flg = logGroup.GetFastLogGroup();
        System.out.println(String.format("\tcategory\t:\t%s\n\tsource\t:\t%s\n\ttopic\t:\t%s\n\tmachineUUID\t:\t%s",
            flg.getCategory(), flg.getSource(), flg.getTopic(), flg.getMachineUUID());
        System.out.println("Tags");
        for (int tagIdx = 0; tagIdx < flg.getLogTagsCount(); ++tagIdx) {
            FastLogTag logtag = flg.getLogTags(tagIdx);
            System.out.println(String.format("\t%s\t:\t%s", logtag.getKey(), logtag.getValue()));
        }
        for (int lIdx = 0; lIdx < flg.getLogsCount(); ++lIdx) {
            FastLog log = flg.getLogs(lIdx);
            System.out.println("-----\nLog: " + lIdx + ", time: " + log.getTime() + ", GetContentCount: " + log.getContentsCount());
            for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx) {
                FastLogContent content = log.getContents(cIdx);
                System.out.println(content.getKey() + "\t:\t" + content.getValue());
            }
        }
    }
    String next_cursor = logDataRes.GetNextCursor();
    System.out.println("The Next cursor:" + next_cursor);
    curCursor = next_cursor;
}
// Note: You can call the following API operation only after you enable the indexing feature.
// Before you query logs, wait for 1 minute.
try {
    Thread.sleep(60 * 1000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
// Query the distribution of logs.
String query = "<Specify the query keyword. To query all contents, use an empty string.>";
int from = (int) (new Date().getTime() / 1000 - 300);
int to = (int) (new Date().getTime() / 1000);
GetHistogramsResponse res3 = null;
while (true) {

```

```

        GetHistogramsRequest req3 = new GetHistogramsRequest(project, logstore, topic, query, from, to);
        res3 = client.GetHistograms(req3);
        if (res3 != null && res3.IsCompleted()) // IsCompleted()
            // If "true" is returned, the query result is accurate.
            // If "false" is returned, the query is repeated.
        {
            break;
        }
        Thread.sleep(200);
    }
    System.out.println("Total count of logs is " + res3.GetTotalCount());
    for (Histogram ht : res3.GetHistograms()) {
        System.out.printf("from %d, to %d, count %d.\n", ht.GetFrom(), ht.GetTo(), ht.GetCount());
    }
    // Query log data.
    long total_log_lines = res3.GetTotalCount();
    int log_offset = 0;
    int log_line = 10;
    while (log_offset <= total_log_lines) {
        GetLogsResponse res4 = null;
        // For each log offset, 10 lines are read at a time. If a read operation fails, a maximum of three retries are allowed.
        for (int retry_time = 0; retry_time < 3; retry_time++) {
            GetLogsRequest req4 = new GetLogsRequest(project, logstore, from, to, topic, query, log_offset,
                log_line, false);
            res4 = client.GetLogs(req4);
            if (res4 != null && res4.IsCompleted()) {
                break;
            }
            Thread.sleep(200);
        }
        System.out.println("Read log count:" + String.valueOf(res4.GetCount()));
        log_offset += log_line;
    }
    // Enable the analytics feature. You can use SQL statements only after you enable the analytics feature. You can log on to the Log Service console or use the SDK to enable the analytics feature.
    IndexKeys indexKeys = new IndexKeys();
    ArrayList<String> tokens = new ArrayList<String>();
    tokens.add(",");
    tokens.add(".");
    tokens.add("#");
    IndexKey keyContent = new IndexKey(tokens, false, "text");
    indexKeys.AddKey("index0", keyContent);
    keyContent = new IndexKey(new ArrayList<String>(), false, "long");
    indexKeys.AddKey("index1", keyContent);
    keyContent = new IndexKey(new ArrayList<String>(), false, "double");
    indexKeys.AddKey("index2", keyContent);
    IndexLine indexLine = new IndexLine(new ArrayList<String>(), false);
    Index index = new Index(7, indexKeys, indexLine);
    CreateIndexRequest createIndexRequest = new CreateIndexRequest(project, logstore, index);
    client.CreateIndex(createIndexRequest);
    // Use the analytics feature.
    GetLogsRequest req4 = new GetLogsRequest(project, logstore, from, to, "", "index0:value | select avg(index1) as v1, sum(index2) as v2, index0 group by index0");
    GetLogsResponse res4 = client.GetLogs(req4);
    if (res4 != null && res4.IsCompleted()) {
        for (QueriedLog log : res4.GetLogs()) {
            LogItem item = log.GetLogItem();
            for (LogContent content : item.GetLogContents()) {
                System.out.print(content.GetKey() + ":" + content.GetValue());
            }
            System.out.println();
        }
    }
}
}
}

```

Usage notes

- To improve the I/O efficiency of your system, we recommend that you do not use SDKs to write data to Log Service. For information about the recommended data writing methods, see the "Producer Library" section in *User Guide. Producer Library*.
- To call API operations for data consumption in Log Service, we recommend that you do not use SDKs. Log Service provides an advanced consumer library. The consumer library consumes data in a manner that is transparent to the user, and provides advanced features such as load balancing and sequential consumption. For more information, see Consumer Library in *User Guide. Consumer Library*.

4.2.3. .NET SDK

Log Service provides developers with Log Service SDK for .NET to simplify the use of Log Service.

Download URL

Log Service SDK for .NET allows developers on Windows to use Log Service on the .NET platform. Log Service SDK for .NET supports .NET Framework 3.5, .NET Framework 4.0, and .NET Framework 4.5. Even though different SDK files are used for different .NET Framework versions, the APIs and features of these versions are the same.

The download URL of Log Service SDK for .NET is [GitHub](#).

Step 1: Create an Apsara Stack tenant account

To access Log Service, you must create an account in the Apsara Uni-manager Management Console.

Step 2: Obtain an AccessKey pair

To use Log Service SDK for .NET, you must obtain an AccessKey pair.

Log on to the Apsara Uni-manager Management Console and select an AccessKey pair for the SDK.

The AccessKey pair will be used in the following steps and must be kept confidential. For information about how to use an AccessKey pair in Log Service SDK for .NET, see [Basic configurations](#).

Step 3: Create a project and a Logstore in the Log Service console

Use Log Service SDK for .NET to create a project and a Logstore.

Note

- The account that you use to create the project and the Logstore must be the same as the account that you used to obtain the AccessKey pair.
- The project name must be unique in Log Service. The Logstore name must be unique in the project.
- After you create the project, you cannot modify its region or migrate the project across regions.

Step 4: Install a .NET development environment

Log Service SDK for .NET supports the .NET Framework 3.5, .NET Framework 4.0, and .NET Framework 4.5 development environments. We recommend that you install one of the following .NET development environments:

- Microsoft .NET Framework 3.5, Microsoft .NET Framework 4.0, or Microsoft .NET Framework 4.5 (based on your environment)
- Visual Studio 2010 or later

Step 5: Download and install Log Service SDK for .NET

After you build a .NET development environment, you must install Log Service SDK for .NET. To install Log Service SDK for .NET, perform the following steps:

1. Download Log Service SDK for .NET.
 - Download Log Service SDK for .NET from [GitHub](#).
 - Click [here](#) to download the latest Log Service SDK for .NET package.
2. Decompress the downloaded package to a specified directory. Log Service SDK for .NET does not require installation. The following step describes how to use the SDK in your Visual Studio project.

Step 6: Create a Log Service .NET project

After you install the .NET development environment and Log Service SDK for .NET, you can create a Log Service .NET project. For more information, see the LOG SDK Sample project in the SLS SDK 4.0 solution at [GitHub](#).

4.2.4. .NET Core SDK

Context

Log Service SDK for .NET Core allows developers to use Log Service by using the .NET Core framework.

The download URL of Log Service SDK for .NET Core is [GitHub](#).

Procedure

1. Create an Apsara Stack tenant account.

To access Log Service, you must create an account in the Apsara Uni-manager Management Console.
2. Obtain an Apsara Stack AccessKey pair.

To use Log Service SDK for .NET Core, you must obtain an Apsara Stack AccessKey pair.

Log on to the Apsara Uni-manager Management Console and select an AccessKey pair for the SDK.

The AccessKey pair will be used in the following steps and must be kept confidential. For information about how to use the AccessKey pair in Log Service SDK for .NET Core, see [Basic configurations](#).
3. Create a project and a Logstore.

Before you can use the SDK, you must create a project and a Logstore in the Log Service console.

 - i. Log on to the Log Service console.
 - ii. In the Projects section, click **Create Project**.
 - iii. Enter a **Project Name**, select a **Region**, and then click **OK**.
 - iv. In the **Projects** section, click the name of the project. On the page that appears, click the plus sign (+) in the left-side navigation pane.

You can also click **Create** to create a Logstore as prompted after you create a project.
 - v. In the dialog box that appears, set the required parameters, and then click **OK**.

You must specify the Logstore name, data retention period, and number of shards. Set the number of shards based on your business requirements. In this example, four shards are created.

Note

- The account that you use to create the project and the Logstore must be the same as the account that you used to obtain the AccessKey pair.
- The project name must be unique in Log Service. The Logstore name must be unique in the project.
- After you create the project, you cannot modify its region or migrate the project across regions.

4. Install the .NET Core development environment.

Log Service SDK for .NET Core supports the following versions:

- .NET Core 2.0
- .NET Framework (with .NET Core 1.x SDK) 4.6.2
- .NET Framework (with .NET Core 2.0 SDK) 4.6.1

For more information about all supported versions, visit [GitHub](#).

5. Download and install Log Service SDK for .NET Core.

After you build the .NET Core development environment, you must install Log Service SDK for .NET Core.

i. Download Log Service SDK for .NET Core.

[Click here](#) to download Log Service SDK for .NET Core from GitHub.

ii. Decompress the downloaded package to a specified directory.

Log Service SDK for .NET Core does not require installation. You can use Log Service SDK for .NET Core for your Visual Studio project.

6. Create a Log Service .NET Core project.

After you install the .NET Core development environment and Log Service SDK for .NET Core, you can create a .NET Core project. For more information, visit [GitHub Wiki](#).

4.2.5. PHP SDK

Log Service provides developers with Log Service SDK for PHP to simplify the use of Log Service.

Download URL

You can download Log Service SDK for PHP from [GitHub](#):

Step 1: Create an Apsara Stack tenant account

To access Log Service, you must create an account in the Apsara Uni-manager Management Console.

Step 2: Obtain an Apsara Stack AccessKey pair

To use Log Service SDK for PHP, you must obtain an AccessKey pair.

Log on to the Apsara Uni-manager Management Console. Select an AccessKey pair for the SDK. The AccessKey pair consists of an AccessKey ID and an AccessKey secret.

The AccessKey pair will be used in the following steps and must be kept confidential. For information about how to use an AccessKey pair in Log Service SDK for PHP, see [Basic configurations](#).

Step 3: Create a project and a Logstore in the Log Service console

Use Log Service SDK for PHP to create a project and a Logstore.

Note

- The account that you use to create the project and the Logstore must be the same as the account that you used to obtain the AccessKey pair.
- The project name must be unique in Log Service. The Logstore name must be unique in the project.
- After you create the project, you cannot modify its region or migrate the project across regions.

Step 4: Install a PHP development environment

Log Service SDK for PHP supports PHP V5.2.1 and later. You can install a supported PHP on an on-premises server and build a PHP development environment.

Step 5: Download and install Log Service SDK for PHP

After you build a PHP development environment, you must install Log Service SDK for PHP. To install Log Service SDK for PHP, perform the following steps:

1. Download the latest SDK for PHP package from [GitHub](#).
2. Decompress the downloaded package to a specified directory. Log Service SDK for PHP does not require installation. Log Service SDK for PHP includes the SDK code, third-party dependent packages, and an autoloader to simplify the use of the SDK. The following step describes how to use the SDK in your PHP project.

Step 6: Create a PHP project

To interact with Log Service and obtain the output, you can run the following sample code in a text editor or the PHP integrated development environment (IDE):

```
<? php
/* Use the autoloader to load all required PHP modules. Specify the path of the file that contains the autoloader. */
require_once realpath(dirname(__FILE__) . '/../Log_Autoload.php');
$endpoint = 'regionid.example.com'; // The endpoint of the China (Hangzhou) region is used as an example. Replace it with the actual endpoint based on your region and access method.
$accessKeyId = '<youraccessKeyId>'; // Specify your AccessKey ID.
$accessKey = '<youraccessKey>'; // Specify your AccessKey secret.
$project = 'your_project'; // The name of the project created in the preceding steps.
$logstore = 'your_logstore'; // The name of the Logstore created in the preceding steps.
$client = new Aliyun_Log_Client($endpoint, $accessKeyId, $accessKey);
# List all Logstores in the current project.
$req1 = new Aliyun_Log_Models_ListLogstoresRequest($project);
$res1 = $client->listLogstores($req1);
var_dump($res1);
# Create a Logstore.
$req2 = new Aliyun_Log_Models_CreateLogstoreRequest($project,$logstore,3,2);
$res2 = $client -> createLogstore($req2);
# Wait until the Logstore takes effect.
sleep(60);
```

```

# Write logs.
$topic = "";
$source = "";
$logitems = array();
for ($i = 0; $i < 5; $i++)
{
    $contents = array('index1'=> strval($i));
    $logItem = new Aliyun_Log_Models_LogItem();
    $logItem->setTime(time());
    $logItem->setContents($contents);
    array_push($logitems, $logItem);
}
$req2 = new Aliyun_Log_Models_PutLogsRequest($project, $logstore, $topic, $source, $logitems);
$res2 = $client->putLogs($req2);
var_dump($res2);
# Pull the data.
# List the shard IDs in the Logstore.
$listShardRequest = new Aliyun_Log_Models_ListShardsRequest($project,$logstore);
$listShardResponse = $client -> listShards($listShardRequest);
foreach($listShardResponse-> getShardIds() as $shardId)
{
    # Obtain the cursor that corresponds to each shard ID.
    $getCursorRequest = new Aliyun_Log_Models_GetCursorRequest($project,$logstore,$shardId,null, time() - 60);
    $response = $client -> getCursor($getCursorRequest);
    $cursor = $response-> getCursor();
    $count = 100;
    while(true)
    {
        # Start to read data from a cursor.
        $batchGetDataRequest = new Aliyun_Log_Models_BatchGetLogsRequest($project,$logstore,$shardId,$count,$cursor);
        var_dump($batchGetDataRequest);
        $response = $client -> batchGetLogs($batchGetDataRequest);
        if($cursor == $response -> getNextCursor())
        {
            break;
        }
        $logGroupList = $response -> getLogGroupList();
        foreach($logGroupList as $logGroup)
        {
            print ($logGroup->getCategory());
            foreach($logGroup -> getLogsArray() as $log)
            {
                foreach($log -> getContentsArray() as $content)
                {
                    print ($content-> getKey().":".$content->getValue()."\t");
                }
                print("\n");
            }
        }
        $cursor = $response -> getNextCursor();
    }
}

# Before you query logs, wait for 1 minute.
sleep(60);
# Query the distribution of logs. Before you query logs, you must ensure that indexes are created. Log Service SDK for PHP does not provide the API operation that you can call to create indexes. You must create indexes in the Log Service console.
$topic = "";
$query="";
$from = time()-3600;
$to = time();
$res3 = NULL;
while (is_null($res3) || (! $res3->isCompleted()))
{
    $req3 = new Aliyun_Log_Models_GetHistogramsRequest($project, $logstore, $from, $to, $topic, $query);
    $res3 = $client->getHistograms($req3);
}
var_dump($res3);
# Query log data.
$res4 = NULL;
while (is_null($res4) || (! $res4->isCompleted()))
{
    $req4 = new Aliyun_Log_Models_GetLogsRequest($project, $logstore, $from, $to, $topic, $query, 5, 0, False);
    $res4 = $client->getLogs($req4);
}
var_dump($res4);

```

4.2.6. Python SDK

Log Service provides developers with Log Service SDK for Python to simplify the use of Log Service.

Download URLs

You can click [here](#) to download Log Service SDK for Python.

You can also download Log Service SDK for Python from [GitHub](#):

<https://github.com/aliyun/aliyun-log-python-sdk>

Step 1: Create an Apsara Stack tenant account

To access Log Service, you must create an account in the Apsara Uni-manager Management Console.

Step 2: Obtain an Apsara Stack AccessKey pair

To use Log Service SDK for Python, you must obtain an AccessKey pair.

Log on to the Apsara Uni-manager Management Console. Select an AccessKey pair for the SDK. The AccessKey pair consists of an AccessKey ID and an AccessKey secret.

The AccessKey pair will be used in the following steps and must be kept confidential. For information about how to use an AccessKey pair in Log Service SDK for Python, see [Basic configurations](#).

Step 3: Create a project and a Logstore in the Log Service console

Before you can use the SDK, you must create a project and a Logstore in the Log Service console.

1. Log on to the Log Service console.
2. In the Projects section, click **Create Project**.
3. Set the **Project Name** and the **Region** parameters, and then click **OK**.
4. In the **Projects** section, click the name of the project. On the page that appears, click the plus sign (+) in the left-side navigation pane.

You can also click **Create** to create a Logstore as prompted after you create a project.

5. In the dialog box that appears, set the required parameters, and then click **OK**.

You must specify the Logstore name, data retention period, and number of shards. Set the number of shards based on your business requirements. In this example, four shards are created.

Note

- The account that you use to create the project and the Logstore must be the same as the account that you used to obtain the AccessKey pair.
- The project name must be unique in Log Service. The Logstore name must be unique in the project.
- After you create the project, you cannot modify its region or migrate the project across regions.

Step 4: Install the Python development environment

Log Service SDK for Python is a pure Python library that supports all operating systems on which Python runs, such as Linux, Mac OS X, and Windows. To install the Python development environment, perform the following steps:

1. Download and install the latest [Python installation package](#).

Note

- Log Service SDK for Python supports Python 2.6, 2.7, 3.3, 3.4, 3.5, and 3.6. You can run the `python -V` command to check the current version of Python.
- Python no longer supports Python 2.6 or Python 3.3. We recommend that you use Python 2.7, Python 3.4, or a later version.

2. Download and install the Python package management tool [pip](#).

After you install pip, run the `pip -V` command to check whether the installation is successful and view the pip version.

Step 5: Install Log Service SDK for Python

Run the following command as an administrator in Shell to install Log Service SDK for Python.

```
pip install -U aliyun-log-python-sdk
```

Step 6: Start a Python program

The following sample code shows how to use Log Service SDK for Python. To interact with Log Service and obtain the relevant output, run the following sample code in a text editor or Python IDE:

For more information, visit [GitHub](#).

```
# encoding: utf-8
import time
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
from aliyun.log.getlogsrequest import GetLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest
def main():
    endpoint = '' # Select the endpoint that matches the region of the project created in the preceding steps.
    accessKeyId = '' # Use your AccessKey ID.
    accessKey = '' # Use your AccessKey secret.
    project = '' # The name of the project created in the preceding steps.
    logstore = '' # The name of the Logstore created in the preceding steps.
    # Note: Configure four shards for the created Logstore.
    # Create a client instance.
    client = LogClient(endpoint, accessKeyId, accessKey)
    # List all Logstores.
    req1 = ListLogstoresRequest(project)
    res1 = client.list_logstores(req1)
    res1.log print()
```

```

topic = ""
source = ""
# Send 10 packets. Each packet contains 10 log entries.
for i in range(10):
    logitemList = [] # LogItem list
    for j in range(10):
        contents = [('index', str(i * 10 + j))]
        logItem = LogItem()
        logItem.set_time(int(time.time()))
        logItem.set_contents(contents)
        logitemList.append(logItem)
    req2 = PutLogsRequest(project, logstore, topic, source, logitemList)
    res2 = client.put_logs(req2)
    res2.log_print()

# List all shards and read the data written within the last minute.
listShardRes = client.list_shards(project, logstore)
for shard in listShardRes.get_shards_info():
    shard_id = shard["shardID"]
    start_time = int(time.time()) - 60
    end_time = start_time + 60
    res = client.get_cursor(project, logstore, shard_id, start_time)
    res.log_print()
    start_cursor = res.get_cursor()
    res = client.get_cursor(project, logstore, shard_id, end_time)
    end_cursor = res.get_cursor()
    while True:
        loggroup_count = 100 # Read 100 packets each time.
        res = client.pull_logs(project, logstore, shard_id, start_cursor, loggroup_count, end_cursor)
        res.log_print()
        next_cursor = res.get_next_cursor()
        if next_cursor == start_cursor:
            break
        start_cursor = next_cursor

# Note: You can call the following API operation to query data only after you enable the indexing feature.
time.sleep(60)
topic = ""
query = "index"
From = int(time.time()) - 600
To = int(time.time())
res3 = None
# Query the number of log entries generated within the last 10 minutes based on the query conditions. Retry the query if not all execution results are correct.
while (res3 is None) or (not res3.is_completed()):
    req3 = GetHistogramsRequest(project, logstore, From, To, topic, query)
    res3 = client.get_histograms(req3)
    res3.log_print()
# Obtain the number of log entries that match the query conditions.
total_log_count = res3.get_total_count()
log_line = 10
# Read 10 log entries each time until all log data is queried. Retry the query three times if not all query results are correct during each query.
for offset in range(0, total_log_count, log_line):
    res4 = None
    for retry_time in range(0, 3):
        req4 = GetLogsRequest(project, logstore, From, To, topic, query, log_line, offset, False)
        res4 = client.get_logs(req4)
        if res4 is not None and res4.is_completed():
            break
        time.sleep(1)
    if res4 is not None:
        res4.log_print()
listShardRes = client.list_shards(project, logstore)
shard = listShardRes.get_shards_info()[0]
# Split a shard.
if shard["status"] == "readwrite":
    shard_id = shard["shardID"]
    inclusiveBeginKey = shard["inclusiveBeginKey"]
    midKey = inclusiveBeginKey[-1] + str((int(inclusiveBeginKey[-1])) + 1)
    client.split_shard(project, logstore, shard_id, midKey)
# Merge shards.
shard = listShardRes.get_shards_info()[1]
if shard["status"] == "readwrite":
    shard_id = shard["shardID"]
    client.merge_shard(project, logstore, shard_id)
# Delete a shard.
shard = listShardRes.get_shards_info()[-1]
if shard["status"] == "readonly":
    shard_id = shard["shardID"]
    client.delete_shard(project, logstore, shard_id)
# Create an external data source. def sample_external_store(client, project): res = client.create_external_store(project, ExternalStoreConfig("rds_store", "cn-qingdao", "rds-vpc", "vpc-m5eq4irclpucpk85frr5j", "i-m5eeo2whsnfg4kzq54ah", "47.104.78.128", "3306", "root", "sfdsfldsfksfldsfs", "meta", "join_meta")); res.log_print() res = client.update_external_store(project, ExternalStoreConfig("rds_store", "cn-qingdao", "rds-vpc", "vpc-m5eq4irclpucpk85frr5j", "i-m5eeo2whsnfg4kzq54ah", "47.104.78.128", "3306", "root", "sfdsfldsfksfldsfs", "meta", "join_meta")); res.log_print() res = client.get_external_store(project, "rds_store"); res.log_print() res = client.list_external_store(project, ""); res.log_print(); res = client.delete_external_store(project, "rds_store")
res.log_print(); # Use Log Service SDK for Python to query data. req4 = GetLogsRequest(project, logstore, From, To, topic, "select count(1)", 10, 0, False) res4 = client.get_logs(req4) # Use Log Service SDK for Python to query data in ApsaraDB RDS for MySQL through the JOIN syntax. req4 = GetLogsRequest(project, logstore, From, To, topic, "select count(1) from " + logstore + " join rds_store on id=id", 10, 0, False) res4 = client

```

```

get_logs(project, logstore, From, To, topic, " | i
nt.get_logs(req4) # Use Log Service SDK for Python to write the query result to RDS. req4 = GetLogsRequest(project, logstore, From, To, topic, " | i
nsert into rds_store select count(1) ", 10,0, False) res4 = client.get_logs(req4)
if __name__ == '__main__':
    main()

```

4.2.7. SDK for Android

The Log Service SDK for Android is used to collect user data from Android platforms and write the data to Log Service.

Download URL at GitHub

<https://github.com/aliyun/aliyun-log-android-sdk>

4.2.8. SDK for C

The Log Service SDK for C is used to access logs in Log Service from multiple platforms, such as the MIPS and OpenWrt systems.

The SDK for C uses libcurl as a network library. The *apr/apr-util* library resolves issues about memory management and cross-platform operations. You can use the SDK for C after you compile a small part of the source code.

In addition, the C Producer Library and C Producer Lite Library provide you with a one-stop log collection solution across different platforms. This solution features simple deployment, high performance, and low resource consumption.

For more information, visit the following GitHub URLs:

- [C Producer Library \(recommended for servers\)](#)
- [C Producer Lite Library \(recommended for IoT and smart devices\)](#)

4.2.9. SDK for Go

Log Service provides developers with the SDK for Go to simplify the use of Log Service.

The Log Service SDK for Go allows developers to perform the following operations:

- Write and consume data in batches.
- Use keywords to query indexes.
- Manage Logtail configurations and machine groups.

You can download the SDK for Go from the following GitHub URL: <https://github.com/aliyun/aliyun-log-go-sdk>.

For information about the usage instructions and sample code, see README.

4.2.10. iOS SDK

Log Service SDK for iOS is implemented based on Log Service APIs. The SDK allows you to write logs to Log Service.

You can download Log Service SDK for iOS from GitHub based on your programming language.

- Swift: <https://github.com/aliyun/aliyun-log-ios-sdk>.
- Object-C: <https://github.com/lujiajing1126/AllyunLogObjc>. AllyunLogObjc is a third-party tested SDK. You can raise issues to the author and receive a quick response.

Swift-based sample code of Log Service SDK for iOS

```

/*
    Use an endpoint, AccessKey ID, and AccessKey secret to create a Log Service client.
    @endPoint: The endpoint. For more information, visit https://help.aliyun.com/document_detail/29008.html.
*/
let myClient = try! LOGClient(endPoint: "",
                             accessKeyID: "",
                             accessKeySecret: "",
                             projectName: "")

/* Create a log group. */
let logGroup = try! LogGroup(topic: "mTopic", source: "mSource")
/* Write a log entry to the log group. */
let log1 = Log()
try! log1.PutContent("K11", value: "V11")
try! log1.PutContent("K12", value: "V12")
try! log1.PutContent("K13", value: "V13")
logGroup.PutLog(log1)
/* Write a log entry to the log group. */
let log2 = Log()
try! log2.PutContent("K21", value: "V21")
try! log2.PutContent("K22", value: "V22")
try! log2.PutContent("K23", value: "V23")
logGroup.PutLog(log2)
/* Send the log entry. */
myClient.PostLog(logGroup, logStoreName: "") { response, error in
    // handle response however you want
    if error?.domain == NSErrorDomain { error?.code == NSErrorDomain {
        print("timed out") // note, 'response' is likely 'nil' if it timed out
    }
}
}

```

Objective-C

For information about how to use Log Service SDK for iOS based on Objective-C, visit [Objective-C](#) at GitHub.

4.2.11. SDK for C ++

The Log Service SDK for C ++ allows you call Log Service API operations in C++ programs. This SDK applies only to Linux platforms.

The Log Service SDK for C ++ supports all Log Service APIs and provides multiple features, such as resource creation and data read/write.

You can download the SDK for C ++ from the following GitHub URL: <https://github.com/aliyun/aliyun-log-cpp-sdk>.

4.2.12. Node.js SDK

Log Service SDK for Node.js allows developers to use Log Service by using Node.js programs. This topic describes how to use Log Service SDK for Node.js to write log data to Log Service.

Procedure

1. Create an Apsara Stack tenant account.
2. Obtain an Apsara Stack AccessKey pair.

The AccessKey pair will be used in the following steps and must be kept confidential. For information about how to use the AccessKey pair in Log Service SDK for Node.js, see [Basic configurations](#).

3. Create a project.
 - i. Use the Node.js web framework Express to create a project.

For more information, see [installing](#).

- ii. Install [morgan](#).

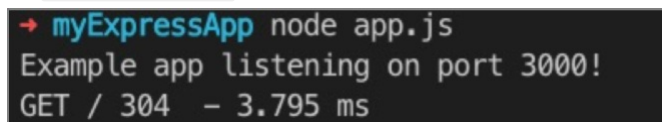
- iii. Create an app.js file in the project.

The app.js file contains the following code:

```
var express = require('express')
var morgan = require('morgan')
var app = express()
const logger = morgan(function (tokens, req, res) {
  return [
    tokens.method(req, res),
    tokens.url(req, res),
    tokens.status(req, res),
    tokens.res(req, res, 'content-length'), '-',
    tokens['response-time'](req, res), 'ms'
  ].join(' ')
})
app.use(logger)
app.get('/', (req, res) => res.send('Hello World!'))
app.listen(3000, () => console.log('Example app listening on port 3000!'))
```

- iv. Run the **node app.js** command to start the project.

Visit <http://localhost:3000> . Then, access logs are generated.



```
➔ myExpressApp node app.js
Example app listening on port 3000!
GET / 304 - 3.795 ms
```

4. Write logs to Log Service.
 - i. Install aliyun-sdk.

Click [here](#) to download aliyun-sdk. Run the **npm install aliyun-sdk** command to install aliyun-sdk.
 - ii. Create a project and a Logstore in the Log Service console.

iii. Call Log Service SDK for Node.js in the project.

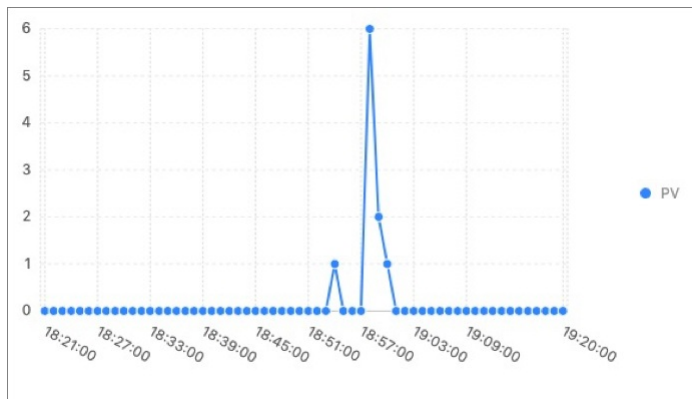
The following sample code is used:

```
var express = require('express')
var morgan = require('morgan')
var ALY = require('aliyun-sdk') // Call aliyun-sdk.
var app = express()
const logger = morgan(function (tokens, req, res) {
  // Obtain log data.
  const log = [
    tokens.method(req, res),
    tokens.url(req, res),
    tokens.status(req, res),
    tokens.res(req, res, 'content-length'), '-',
    tokens['response-time'](req, res), 'ms'
  ]
  var sls = new ALY.SLS({
    "accessKeyId": "LTA*****G9", // The AccessKey ID that is obtained in Step 2
    "secretAccessKey": "MfX*****l0l", // The AccessKey secret that is obtained in Step 2
    endpoint: 'http://cn-hangzhou.sls.aliyuncs.com',
    apiVersion: '2015-06-01'
  })
  var logGroup = {
    logs: [
      {
        time: Math.floor(new Date().getTime()/1000), // Unit: seconds
        contents: [ // Key-value pairs
          {
            key: 'method',
            value: log[0]
          },
          {
            key: 'url',
            value: log[1]
          },
          {
            key: 'status',
            value: log[2]
          },
          {
            key: 'time',
            value: log[5]
          }
        ]
      }
    ],
    topic: 'topicName', //optional
    source: '127.0.0.1' //optional
  };
  var opt = {
    projectName: 'wangdan-test',
    logStoreName: 'test',
    logGroup
  }
  sls.putLogs(opt,function(error, data) {
    console.log(error, data)
  })
  return [
    tokens.method(req, res),
    tokens.url(req, res),
    tokens.status(req, res),
    tokens.res(req, res, 'content-length'), '-',
    tokens['response-time'](req, res), 'ms'
  ].join(' ')
})
app.use(logger)
app.get('/', (req, res) => res.send('Hello World!'))
app.listen(3000, () => console.log('Example app listening on port 3000!'))
```

5. View collected log data.

In the Log Service console, find the Logstore in the project. Click the Logstore name, enter a query statement in the search box on the **Search & Analysis** page, and then click **Search & Analyze** to view log data.

The following figure indicates that most page views (PVs) occurred between 18:30 and 19:00 on September 12 during the specified time range. You can record and display the PV statistics as a chart on your dashboard.



4.3. API Reference

4.3.1. Overview

Log Service allows you to collect and manage logs by using the Apsara Uni-manager Management Console. In addition, Log Service allows you to use the Log Service API to collect logs, query logs, and manage your projects and Logstores.

Object	Operation
Log	Basic terms such as log and log group
Project	ListProject, CreateProject, DeleteProject, GetProject, and UpdateProject GetProjectLogs (Queries all logs in a project)
Config	ListConfig, CreateConfig, DeleteConfig, GetConfig, and UpdateConfig GetAppliedMachineGroups (Queries the machine groups to which a Logtail configuration file is applied)
MachineGroup	ListMachineGroup, CreateMachineGroup, DeleteMachineGroup, GetMachineGroup, and UpdateMachineGroup ApplyConfigToMachineGroup and RemoveConfigFromMachineGroup (Deletes a Logtail configuration file) GetAppliedConfigs (Queries the applied Logtail configuration files)
Logstore	ListLogstore, CreateLogstore, DeleteLogstore, GetLogstore, and UpdateLogstore GetLogs (Queries logs) and GetHistograms (Queries the distribution of logs)
Index	GetIndex, CreateIndex, DeleteIndex, and UpdateIndex
Shard	ListShards, SplitShard, and MergeShards PostLog (Writes logs) GetCursor (Locates logs) PullLogs (Consumes logs)
Consumer group	CreateConsumerGroup, DeleteConsumerGroup, ListConsumerGroup, and UpdateConsumerGroup HeartBeat (Sends heartbeat packets), GetCheckPoint, and UpdateCheckPoint

You can call the API to perform the following operations:

- Collect logs based on [Logtail configurations](#) and [Logtail machine group](#).
- Create [Logstores](#) and write logs to or read logs from the Logstores.

Note

- The Log Service API is a RESTful API based on the HTTP protocol.
- Security verification is required for all API requests. For information about the API request signature method and process, see [Request signatures](#).

4.3.2. Common request headers

The Log Service API is a RESTful API based on the HTTP protocol. The Log Service API supports a set of common request headers that can be used in all API requests. Each Log Service API request must provide a common request header. The following table describes the fields of the common request headers.

Header field	Type	Description
Accept	String	The type of the response that the client expects Log Service to return. Two response types are supported: application/json and application/x-protobuf. This field is optional and valid for only GET requests. The specific value is subject to the definition of each API.
Accept-Encoding	String	The compression algorithm that the client expects Log Service to return. LZ4 and DEFLATE are supported. The value can also be an empty string, indicating that no compression is performed. This field is optional and valid only for GET requests. The specific value is subject to the definition of each API.
Authorization	String	The signature content. For more information, see Request signatures .
Content-Length	Numeric value	The length of the HTTP request body defined in RFC 2616. If the request does not have a body, this field is not required.
Content-MD5	String	The uppercase string that is generated after the request body is calculated based on the MD5 algorithm. If the request does not have a body, this field is not required.
Content-Type	String	The type of the HTTP request body defined in RFC 2616. Log Service API requests support only two types: application and x-protobuf. If the request does not have a body, this field is not required. The specific value is subject to the definition of each API.
Date	String	The time when the request is sent. Only the RFC 822 format is supported. The GMT standard time is used. The formatted string is %a, %d %b %Y %H:%M:%S GMT, for example, Mon, 3 Jan 2010 08:33:47 GMT.
Host	String	The complete hostname of the HTTP request. The value does not include protocol headers such as http://. Example: big-game.regionid.example.com.
x-log-apiversion	String	The API version. The current version is 0.6.0.
x-log-bodyrawsize	Numeric value	The initial size of the request body. If the request does not have a body, the value is 0. If the request body is compressed, the value is the size of the initial data before the compression. Valid values: 0 to 3 × 1024 × 1024. This field is optional and only required when the request body is compressed.
x-log-compresstype	String	The compression type of the API request body. LZ4 and DEFLATE that are defined in RFC 1951 are supported. If you want to use the ZLIB compression type, see RFC 1950. If the request body is not compressed, this field is not required.
x-log-date	String	The time when the request is sent. The time format is the same as that of the Date field. This field is optional. If a request contains this field, the value of this field replaces the value of the Date field for the request verification performed by Log Service. The Date field must be provided regardless of whether a request has an x-log-date field.
x-log-signaturemethod	String	The signature calculation method. Set the value to hmac-sha1.

Note

- The maximum difference between the time represented by the Date field of a request and the time when Log Service receives the request is 15 minutes. If the difference exceeds 15 minutes, Log Service rejects the request. If an x-log-date field is specified in the request, the time difference is calculated based on the value of the x-log-date field.
- If the compression algorithm is specified in x-log-compresstype of a request, the raw data must be compressed and then placed into the HTTP request body. The values of Content-Length and Content-MD5 fields are calculated based on the compressed HTTP body.
- The value of the Date field cannot be specified for the HTTP requests sent by some platforms. As a result, no correct value of the Date field is available to calculate the request signature. In this case, you must specify an x-log-date field and use the value of this field to calculate the request signature. After Log Service receives an API request, it checks whether the request contains an x-log-date field. If yes, Log Service uses the value of the x-log-date field for signature verification. If no, Log Service uses the value of the standard HTTP field Date for signature verification.

4.3.3. Common response headers

The Log Service API is a RESTful API based on the HTTP protocol. Each Log Service API response provides a common response header. The following table describes the fields in the common response headers.

Header field	Type	Description
Content-Length	Numeric value	The length of the HTTP response defined in RFC 2616.
Content-MD5	String	The MD5 value of the HTTP response defined in RFC 2616. This value is an uppercase string after the request body is calculated based on the MD5 algorithm.
Content-Type	String	The type of the HTTP response defined in RFC 2616. Log Service supports two response types: application/json and application/x-protobuf.
Date	String	The time when the request is returned. Only the RFC 822 format is supported. The GMT standard time is used. The formatted string is %a, %d %b %Y %H:%M:%S GMT, for example, Mon, 3 Jan 2010 08:33:47 GMT.
x-log-requestid	String	The unique ID that marks the request. The ID is generated by Log Service. This response header field is irrelevant to applications. It is used to troubleshoot problems. To troubleshoot failed API requests, you can provide this ID to the technical support team of Log Service.

4.3.4. Request signatures

Each HTTP request of a Log Service API operation must pass security verification to ensure log data security. The security verification is performed by using the symmetric encryption algorithm. You can use an AccessKey pair for security verification.

The security verification process of an API request is as follows:

- The requester generates a signature string based on the API request content, including the HTTP header and body.
- The requester uses an AccessKey pair that consists of an AccessKey ID and AccessKey secret to sign the signature string. A digital signature is generated for this API

request.

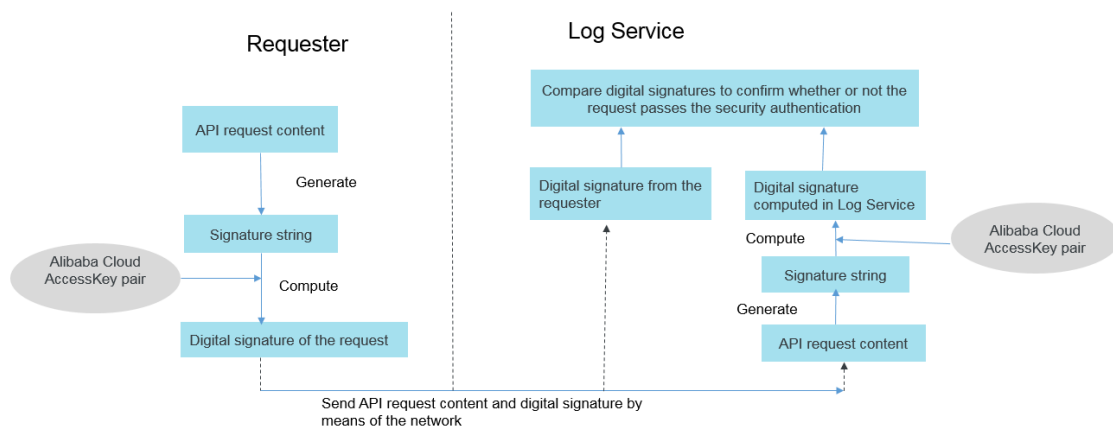
- The requester sends both the API request content and digital signature to Log Service.
- Log Service receives the request, and then repeats steps 1 and 2.

Note Log Service retrieves the AccessKey pair from the backend to calculate the signature string and generate the expected digital signature for this request.

- The server compares the expected digital signature with the digital signature sent from the requester. If both signatures are the same, the request passes the security verification. Otherwise, the request is rejected.

The following figure shows the security verification process.

Security verification process



The security verification process can also be used for the following purposes:

- Identify the user that sends the API request. A user must specify the AccessKey pair that is used to generate the digital signature before the user sends the API request. Then, Log Service can use the AccessKey pair to identify the user and implement access control.
- Check whether the API request is intercepted and tampered with during transmission. Log Service recalculates the signature string based on the received API request. If the API request is intercepted and tampered with during transmission, the API request cannot pass the security verification.

Sign an API request

To pass the security verification for an API request, you must sign the API request on the requester to generate a digital signature. You must also use the Authorization field to transmit the digital signature of the API request over the network. The format of the Authorization field is as follows:

```
Authorization:LOG <AccessKeyId>:<Signature>
```

In the preceding format, the value of the Authorization field contains the AccessKey ID. The AccessKey secret is used to construct the signature. To construct the signature, follow these steps:

- Prepare an AccessKey pair.

To generate a digital signature for an API request, you must use an AccessKey pair. The AccessKey pair consists of an AccessKey ID and an AccessKey secret.

- Generate a signature string for the API request.

The signature string of a Log Service API operation is generated based on the method, header, and body of the HTTP request. The signature string is generated by using the following formula:

```
SignString = VERB + "\n"
            + CONTENT-MD5 + "\n"
            + CONTENT-TYPE + "\n"
            + DATE + "\n"
            + CanonicalizedLOGHeaders + "\n"
            + CanonicalizedLOGResource
```

In the preceding formula, `\n` indicates the newline escape character and the plus sign (`+`) indicates the string concatenation operator. The following table describes the other elements.

Signature string elements

Element	Description	Example
VERB	The method name of the HTTP request.	PUT, GET, and POST
CONTENT-MD5	The MD5 value of the HTTP request body, which must be an uppercase string.	875264590688CA6171F6228AF5BBB3D2
CONTENT-TYPE	The type of the HTTP request body.	application/x-protobuf
DATE	The standard timestamp field of the HTTP request. This field complies with the RFC 1123 time format and uses the GMT standard time.	Mon, 3 Jan 2010 08:33:47 GMT

Element	Description	Example
CanonicalizedLOGHeaders	The string constructed by the custom fields in the HTTP request. The custom fields are prefixed with <code>x-log</code> and <code>x-acs</code> .	<code>x-log-apiversion:0.6.0\nx-log-bodyrawsize:50\nx-log-signaturemethod:hmac-sha1</code>
CanonicalizedResource	The string constructed by using the HTTP request resources.	<code>/logstores/app_log</code>

If an HTTP request does not have a body, the `CONTENT-MD5` and `CONTENT-TYPE` fields are empty strings. In this case, the signature string is generated by using the following formula:

```
SignString = VERB + "\n"
            + "\n"
            + "\n"
            + DATE + "\n"
            + CanonicalizedLOGHeaders + "\n"
            + CanonicalizedResource
```

The `x-log-date` custom request field is used in Log Service API operations. For more information, see [Common request headers](#). If you specify this field in a request, the field value replaces the value of the HTTP standard request field `Date` during signature generation.

The CanonicalizedLOGHeaders string is constructed as follows:

- Convert the names of all HTTP request fields prefixed with `x-log` and `x-acs` into lowercase letters.
- Sort all Log Service custom request fields obtained in the previous step in lexicographical order.
- Delete all spaces on either side of a delimiter between the request header and content.
- Separate all headers and contents with the `\n` delimiter to obtain the CanonicalizedLOGHeader string.

The CanonicalizedResource string is constructed as follows:

- Set the CanonicalizedResource string to an empty string (`""`).
- Enter the Log Service resources that you want to access, for example, `/logstores/logstorename`. If `logstorename` does not exist, this string is left blank.
- If the request contains a query string (`QUERY_STRING`), add a question mark (`?`) and the query string at the end of the CanonicalizedResource string.

In the request, `QUERY_STRING` is the lexicographically sorted string of the request parameters included in the URL. Equal signs (`=`) are used between the names and values of parameters to form a string. The name-value pairs are then sorted in lexicographical order and connected with ampersands (`&`) to form a string.

The formula is described as follows:

```
QUERY_STRING = "KEY1=VALUE1" + "&" + "KEY2=VALUE2"
```

3. Generate a digital signature for the request.

Log Service API operations support only the `hmac-sha1` digital signature algorithm, which is also the default signature algorithm. The signature formula is as follows:

```
Signature = base64(hmac-sha1(UTF8-Encoding-Of(SignString), AccessKeySecret))
```

The HMAC-SHA1 method defined in [RFC 2104](#) is used as the signature method. The AccessKey secret used in the preceding formula must match the AccessKey ID in the final Authorization field. Otherwise, the request cannot be verified by Log Service.

After the digital signature is generated, use the signature to construct a complete security verification header for the API request. Then, enter the security verification header in the HTTP request and send the request.

Examples

The following two examples demonstrate how to sign an API request. In the examples, the AccessKey pair you use to sign the API request is as follows:

```
AccessKeyId = "<yourAccessKeyId>"
AccessKeySecret = "<yourAccessKeySecret>"
```

- Example 1: you want to send the following GET request to list all Logstores in the `ali-test-project` project:

```
GET /logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host: ali-test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

The following signature string is generated for the API request:

```
GET\n\nMon, 09 Nov 2015 06:11:16 GMT\nx-log-apiversion:0.6.0\nx-log-signaturemethod:hmac-sha1\n/logstores? logstoreName=&offset=0&size=1000
```

The `CONTENT-TYPE` and `CONTENT-MD5` fields are empty because the GET request does not have an HTTP body. If the AccessKey secret is used to generate a signature, the signature is as follows:

```
jEYOTCJs2e88o+y5F4/S5IsnBJQ=
```

The digitally signed HTTP request is as follows:

```
GET /logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host: ali-test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
```

- Example 2: You want to write the following log to the Logstore named `test-logstore` in the `ali-test-project` project:

```
topic=""
time=1447048976
source="10.10.10.1"
"TestKey": "TestContent"
```

The following HTTP request must be constructed based on the Log Service API definition:

```
POST /logstores/test-logstore HTTP/1.1
Date: Mon, 09 Nov 2015 06:03:03 GMT
Host: test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Content-MD5: 1DD45FA4A70A9300CC9FE7305AF2C494
Content-Length: 52
x-log-apiversion:0.6.0
x-log-bodyrawsize:50
x-log-compresstype:lz4
x-log-signaturemethod:hmac-sha1
<The log is serialized to byte streams in the ProtoBuffer format.>
```

In this HTTP request, the log is first serialized to byte streams in the ProtoBuffer format and then used as the request body. For more information, see [Data encoding](#). The value of the Content-Type header field value in this request is application/x-protobuf. The value of the Content-MD5 header field is the MD5 value that corresponds to the request body. The signature string of this request is as follows:

```
POST\n1DD45FA4A70A9300CC9FE7305AF2C494\napplication/x-protobuf\nMon, 09 Nov 2015 06:03:03 GMT\nx-log-apiversion:0.6.0\nx-log-bodyrawsize:50\nx-log-compresstype:lz4\nx-log-signaturemethod:hmac-sha1\n/logstores/test-logstore
```

If the AccessKey secret is used to generate a signature, the signature is as follows:

```
XWLGyHGg2F2hcfXWxMLiNkGki6g=
```

The digitally signed HTTP request is as follows:

```
POST /logstores/test-logstore HTTP/1.1
Date: Mon, 09 Nov 2015 06:03:03 GMT
Host: test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Content-MD5: 1DD45FA4A70A9300CC9FE7305AF2C494
Content-Length: 52
x-log-apiversion:0.6.0
x-log-bodyrawsize:50
x-log-compresstype:lz4
x-log-signaturemethod:hmac-sha1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
<The log is serialized to byte streams in the ProtoBuffer format.>
```

4.3.5. Common error codes

When an API request error occurs, Log Service returns an error message, including the HTTP status code and the specific error details in the response body. The error details in the response body are formatted as follows:

```
{
  "errorCode" : <ErrorCode>,
  "errorMessage" : <ErrorMessage>
}
```

Some of the error messages that may be returned by Log Service apply to most APIs and are defined as common error codes. The other error messages apply only to some APIs and are defined as special error codes. The unique error codes of an API are described in the relevant API reference.

Common error codes

HTTP status code	Error code	Error message	Description
411	MissingContentLength	Content-Length does not exist in http header when it is necessary.	The error message returned because the required Content-Length field does not exist in the request header.
415	InvalidContentType	Content-Type {type} is unsupported.	The error message returned because the specified Content-Type is not supported.
400	MissingContentType	Content-Type does not exist in http header when body is not empty.	The error message returned because the Content-Type header field is not specified when the HTTP request body is not empty.
400	MissingBodyRawSize	x-log-bodyrawsize does not exist in header when it is necessary.	The error message returned because the required x-log-bodyrawsize field does not exist in the request header in the compression scenario.
400	InvalidBodyRawSize	x-log-bodyrawsize is invalid.	The error message returned because the x-log-bodyrawsize value is invalid.
400	InvalidCompressType	x-log-compresstype {type} is unsupported.	The error message returned because the compression type specified in the x-log-compresstype parameter is not supported.

HTTP status code	Error code	Error message	Description
400	MissingHost	Host does not exist in http header.	The error message returned because the Host field does not exist in the HTTP request header.
400	MissingDate	Date does not exist in http header.	The error message returned because the Date field does not exist in the HTTP request header.
400	InvalidDateFormat	Date {date} must follow RFC822.	The error message returned because the value of the Date field in the request header does not comply with the RFC 822 standard.
400	MissingAPIVersion	x-log-apiversion does not exist in http header.	The error message returned because the x-log-apiversion field does not exist in the HTTP request header.
400	InvalidAPIVersion	x-log-apiversion {version} is unsupported.	The error message returned because the value of the x-log-apiversion field in the HTTP request header is not supported.
400	MissAccessKeyId	x-log-accesskeyid does not exist in header.	The error message returned because no AccessKey ID exists in the Authorization header field.
401	Unauthorized	The AccessKeyId is unauthorized.	The error message returned because the specified AccessKey ID is unauthorized.
400	MissingSignatureMethod	x-log-signaturemethod does not exist in http header.	The error message returned because the x-log-signaturemethod field does not exist in the HTTP request header.
400	InvalidSignatureMethod	signature method {method} is unsupported.	The error message returned because the signature method specified in the x-log-signaturemethod header field is not supported.
400	RequestTimeTooSkewed	Request time exceeds server time more than 15 minutes.	The error message returned because the request sending time is more than 15 minutes before or after the current server time.
404	ProjectNotExist	Project {name} does not exist.	The error message returned because the specified project does not exist.
401	SignatureNotMatch	Signature {signature} is not matched.	The error message returned because the digital signature of the request does not match the signature computed in Log Service.
403	WriteQuotaExceed	Write quota is exceeded.	The error message returned because the log write quota is exceeded.
403	ReadQuotaExceed	Read quota is exceeded.	The error message returned because the log read quota is exceeded.
500	InternalServerError	Internal server error message.	The error message returned because an internal server error occurred.
503	ServerBusy	The server is busy, please try again later.	The error message returned because the server is busy. Try again later.

The {...} in the error message indicates the specific error information. For example, {name} in the ProjectNotExist error message is replaced by a specific project name.

4.3.6. Project API

4.3.6.1. CreateProject

Creates a project.

Request syntax

```
POST / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "projectName": <ProjectName>,
  "description": <Description>
}
```

Request elements

- Request headers

The CreateProject API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
projectName	String	Yes	The name of the project.
description	String	Yes	The description of the project.

Response elements

- Response headers

The CreateProject API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	ProjectAlreadyExist	Project {project} already exist
400	ParameterInvalid	The body is not valid json string
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
POST / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "projectName": "my-project-test",
  "description": "Description of my-project-test"
}
```

- Sample responses

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-log-requestid: 5B0A619F205DC3F30EDA9322
```

4.3.6.2. DeleteProject

Deletes a project.

Request syntax

```
DELETE / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

Request elements

- Request headers

The DeleteProject API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
projectName	String	Yes	The name of the project. The name is a part of the Host parameter in the header.

Response elements

- Response headers

The DeleteProject API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	The project does not exist: {project}
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
DELETE / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 08:25:04 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 08:25:04 GMT
x-log-requestid: 5B0A6B60BB6EE39764D458B5
```

4.3.6.3. UpdateProject

Modifies a project.

Request syntax

```
PUT / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
```

Request elements

- Request headers

The UpdateProject API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
projectName	String	Yes	The name of the project. The name is a part of the Host parameter in the header.

Parameter	Type	Required	Description
description	String	Yes	The description of the project. The value is an empty string by default.

Response elements

- Response headers

The UpdateProject API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	The project does not exist: <Project>
400	ParameterInvalid	The body is not valid json string
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
PUT / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 40
Connection: Keep-Alive
{
  "description": "Description of my-project-test"
}
```

- Sample responses

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-log-requestid: 5B0A619F205DC3F30EDA9322
```

4.3.6.4. GetProject

Queries a project based on the project name.

Request syntax

```
GET / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

Request elements

- Request headers

The GetProject API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description

Parameter	Type	Required	Description
projectName	String	Yes	The name of the project. The name is a part of the Host parameter in the header.

Response elements

- Response headers

The GetProject API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the request succeeds, the response body contains the details of the project. The following table describes the response parameters.

Parameter	Type	Description
createTime	String	The time when the project was created.
description	String	The description of the project.
lastModifyTime	String	The time when the project was last modified.
owner	String	The ID of the account that was used to create the project.
projectName	String	The name of the project.
status	String	The status of the project.
region	String	The region to which the project belongs.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	The project does not exist: {project}
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
GET / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 08:25:04 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 08:25:04 GMT
x-log-requestid: 5B0A6B60BB6EE39764D458B5
```

4.3.6.5. ListProject

Lists all projects.

Request syntax

```
GET /? offset={offset}&size={size} HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

Request elements

- Request headers

The ListProject API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
offset	Integer	No	The start position of the records to return. Default value: 0.
size	Integer	No	The maximum number of entries to return on each page. Default and maximum value: 500.

Response elements

- Response headers

The ListProject API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the request succeeds, the response body contains the project list. The following table describes the response parameters.

Parameter	Type	Description
count	Integer	The number of returned projects on each page.
total	Integer	The total number of returned projects.
projects	Array	The list of projects.

The following table describes the elements of the projects parameter.

Element	Type	Description
createTime	String	The time when the project was created.
description	String	The description of the project.
lastModifyTime	String	The time when the project was last modified.
owner	String	The ID of the account that was used to create the project.
projectName	String	The name of the project.
status	String	The status of the project.
region	String	The region to which the project belongs.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
GET /? offset=0&size=2&projectName= HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 09:03:33 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```

HTTP/1.1 200
Server: nginx
Content-Type: application/json
Content-Length: 345
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 09:03:33 GMT
x-log-requestid: 5B0A7465AAEA20CA70DE3064
{
  "count": 2,
  "total": 11,
  "projects": [
    {
      "projectName": "project1",
      "status": "Normal",
      "owner": "",
      "description": "",
      "region": "cn-shanghai",
      "createTime": "1524222931",
      "lastModifyTime": "1524539357"
    },
    {
      "projectName": "project123456",
      "status": "Normal",
      "owner": "",
      "description": "",
      "region": "cn-shanghai",
      "createTime": "1471963876",
      "lastModifyTime": "1524539357"
    }
  ]
}

```

4.3.6.6. GetProjectLogs

Queries logs in a project by using an SQL statement.

Request syntax

The following example shows the request syntax of the GetProjectLogs operation:

```

GET /logs query=SELECT avg(latency) as avg_latency FROM where __date__ >'2017-09-01 00:00:00' and __date__ < '2017-09-02 00:00:00'
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

```

Request elements

- Request headers

The GetProjectLogs operation does not have operation-specific request headers. For information about the common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
query	string	Yes	The SQL statement that is used for the query.

Response elements

- Response headers

For information about the common response headers of Log Service API operations, see [Common response headers](#). The following table describes the response headers that are specific to this operation.

Parameter	Type	Description
x-log-progress	String	The status of the query results. Valid values: Incomplete and Complete.
x-log-count	Integer	The total number of logs in the query results.
x-log-processed-rows	Integer	The number of rows that were processed in the query.
x-log-elapsed-millisecond	Integer	The number of milliseconds that were consumed by the query.

- Response parameters

The response body of the GetProjectLogs operation is an array. Each element in the array is a log. The following table describes the response parameters.

Parameter	Type	Description

Parameter	Type	Description
__time__	Integer	The log timestamp. The timestamp follows the UNIX time format. It is the number of seconds that have elapsed since 00:00:00 Thursday, January 1, 1970.
__source__	String	The log source that was specified when logs were written.
[content]	Key-value pair	The original content of logs.

Additional considerations

- The query statement that is used in this API operation is a standard SQL statement.
- You must specify a project in the domain name of the request.
- You must specify a Logstore in the FROM clause of the query statement. A Logstore can be regarded as an SQL table.
- You must specify a time range in the SQL statement by using the __date__ parameter (timestamp) or __time__ parameter (UNIX timestamp).
- If the number of logs in a Logstore significantly changes, Log Service cannot predict the number of times you must call this API operation to obtain complete results. In this case, you must check the value of the x-log-progress parameter in the returned results of each request. You can also call this API operation again to obtain the complete results based on the value of this parameter.

 **Note** Each time you call this operation, the same number of charge units (CUs) are consumed.

Error codes

In addition to [Common error codes](#), the API operation may also return the following operation-specific error codes.

HTTP status code	Error code	Error message	Description
400	ParameterInvalid	parameter is invalid	The error message returned because the specified parameter in the request is invalid.

Example

In this example, the logs whose topic is groupA are queried in a Logstore named app_log. The Logstore belongs to a project named big-game in the China (Hangzhou) region. The time range is from 00:00:00 UT C, September 1, 2014 to 22:00:00 UT C, September 1, 2014, and the keyword is "error". The query starts from the start of the time range. A maximum of 20 logs can be returned.

Sample requests

```
GET /logs/?query=SELECT * FROM <logStoreName> where __line__ = 'abc' and __date__ > '2017-09-01 00:00:00' and __date__ < '2017-09-02 00:00:00'&line=20&offset=0 HTTP/1.1
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.4.0
x-log-signaturemethod: hmac-sha1
```

Sample responses

```
HTTP/1.1 200 OK
Content-MD5: 36F9F7F0339BEAF571581AF1B0AAAFB5
Content-Type: application/json
Content-Length: 269
Date: Wed, 3 Sept. 2014 08:33:47 GMT
x-log-requestid: efag01234-12341-15432f
x-log-progress : Complete
x-log-count : 10000
x-log-processed-rows: 10000
x-log-elapsed-millisecond:5
{
  "progress": "Complete",
  "count": 2,
  "logs": [
    {
      "__time__": 1409529660,
      "__source__": "10.237.0.17",
      "Key1": "error",
      "Key2": "Value2"
    },
    {
      "__time__": 1409529680,
      "__source__": "10.237.0.18",
      "Key3": "error",
      "Key4": "Value4"
    }
  ]
}
```

In the sample response, the value of the x-log-progress parameter is Complete. This value indicates that all logs are queried and the returned results are complete. Two logs meet the specified query conditions. The logs are displayed as the value of the logs parameter. If the value of the x-log-progress parameter in the response is incomplete, you must repeat the request to obtain complete query results.

4.3.7. Logstore operations

4.3.7.1. CreateLogstore

Creates a Logstore.

Request syntax

The request syntax of the CreateLogstore API operation is as follows:

```
POST /logstores HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
  "logstoreName" : <logStoreName>,
  "ttl": <ttl>,
  "shardCount": <shardCount>,
  "autoSplit": <autoSplit>,
  "maxSplitShard": <maxSplitShard>
}
```

Request elements

- Request headers

The CreateLogstore API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore. The name must be unique in a project.
ttl	Integer	Yes	The time to live (TTL) of log data. Unit: days. Valid values: 1 to 3600.
shardCount	Integer	Yes	The number of shards. Valid values: 1 to 100.
enable_tracking	Boolean	No	Specifies whether to enable the WebTracking feature. Default value: false.
autoSplit	Boolean	No	Specifies whether to enable automatic sharding. Default value: false.
maxSplitShard	Integer	No	The maximum number of shards for automatic sharding. Valid values: 1 to 64. Default value: 64. This parameter must be specified if the autoSplit parameter is set to true.

Response elements

- Response headers

The CreateLogstore API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common request headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	LogstoreAlreadyExist	logstore {logstoreName} already exists
500	InternalServerError	Specified Server Error Message
400	LogstoreInfoInvalid	logstore info is invalid
400	ProjectQuotaExceed	Project Quota Exceed

Examples

- Sample requests

```

POST /logstores HTTP/1.1
Header :
{
  x-log-apiversion=0.6.0,
  Authorization=LOG <yourAccessKeyId>:<yourSignature>,
  Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
  Date=Wed, 11 Nov 2015 07:35:00 GMT,
  Content-Length=55,
  x-log-signaturemethod=hmac-sha1,
  Content-MD5=7EF43D0B8F4A807B95E775048C911C72,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
Body :
{
  "logstoreName": "test-logstore",
  "ttl": 1,
  "shardCount": 2,
  "autoSplit": true,
  "maxSplitShard": 64
}

```

- Sample responses

```

HTTP/1.1 200 OK
Header:
{
  Date=Wed, 11 Nov 2015 07:35:00 GMT,
  Content-Length=0,
  x-log-requestid=5642EFA499248C827B012B39,
  Connection=close,
  Server=nginx/1.6.1
}

```

4.3.7.2. DeleteLogstore

Deletes a Logstore, including all shards and indexes of the Logstore.

Request syntax

```

DELETE /logstores/{logstoreName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

```

Request parameters

- Request headers

The DeleteLogstore API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore. The name must be unique in a project.

Response parameters

- Response headers

The DeleteLogstore API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
DELETE /logstores/test_logstore HTTP/1.1
Header :
{
  x-log-apiversion=0.6.0,
  Authorization=LOG <yourAccessKeyId>:<yourSignature>,
  Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
  Date=Wed, 11 Nov 2015 08:09:38 GMT,
  Content-Length=0,
  x-log-signaturemethod=hmac-sha1,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
```

- Sample responses

```
HTTP/1.1 200 OK
Header:
{
  Date=Wed, 11 Nov 2015 08:09:39 GMT,
  Content-Length=0,
  x-log-requestid=5642F7C399248C817B013A07,
  Connection=close,
  Server=nginx/1.6.1
}
```

4.3.7.3. UpdateLogstore

Updates the attributes of a Logstore. Only the time to live (TTL) and shard attributes can be updated.

Request syntax

```
PUT /logstores/{logstoreName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
  "logstoreName": <logstoreName>,
  "ttl": <ttl>,
  "shardCount": <shardCount>,
  "autoSplit": <autoSplit>,
  "maxSplitShard": <maxSplitShard>
}
```

Request elements

- Request headers

The UpdateLogstore API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore. The name must be unique in a project.
ttl	Integer	Yes	The time to live (TTL) of log data. Unit: days. Valid values: 1 to 3600.
shardCount	Integer	Yes	The number of shards. After the attributes of a Logstore are updated, the number of shards is unchanged. You can only call the SplitShard or MergeShards operation to change the number of shards.
enable_tracking	Boolean	No	Specifies whether to enable the WebTracking feature.
autoSplit	Boolean	No	Specifies whether to enable automatic sharding.
maxSplitShard	Integer	No	The maximum number of shards for automatic sharding. Valid values: 1 to 64. This parameter must be specified if the autoSplit parameter is set to true.

Response elements

- Response headers

The UpdateLogstore API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	LogStoreAlreadyExist	logstore {logstoreName} already exists
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	invalid shard count,you can only modify shardCount by split& merge shard

Additional considerations

You can modify the number of shards be only calling the SplitShard or MergeShards API operation.

Examples

• Sample requests

```
PUT /logstores/test-logstore HTTP/1.1
Header:
{
  x-log-apiversion=0.6.0,
  Authorization=LOG <yourAccessKeyId>:<yourSignature>,
  Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
  Date=Wed, 11 Nov 2015 08:28:19 GMT,
  Content-Length=55,
  x-log-signaturemethod=hmac-sha1,
  Content-MD5=757C60FC41CC7D3F60B88E0D916D051E,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
Body :
{
  "logstoreName": "test-logstore",
  "ttl": 1,
  "shardCount": 2,
  "autoSplit": true,
  "maxSplitShard": 64
}
```

• Sample responses

```
HTTP/1.1 200 OK
Header:
{
  Date=Wed, 11 Nov 2015 08:28:20 GMT,
  Content-Length=0,
  x-log-requestid=5642FC2399248C8F7B0145FD,
  Connection=close,
  Server=nginx/1.6.1
}
```

4.3.7.4. GetLogstore

Queries the attributes of a Logstore.

Request syntax

```
GET /logstores/{logstoreName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

• Request headers

The GetLogstore API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore. The name must be unique in a project.

Response elements

- Response headers

The GetLogstore API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

```
{
  "logstoreName" : <logstoreName>,
  "ttl": <ttl>,
  "shardCount": <shardCount>,
  "createTime": <createTime>,
  "lastModifyTime": <lastModifyTime>
}
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist
404	LogstoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
GET /logstores/test-logstore HTTP/1.1
Header :
{
  x-log-apiversion=0.6.0,
  Authorization=LOG <yourAccessKeyId>:<yourSignature>,
  Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
  Date=Wed, 11 Nov 2015 07:53:29 GMT,
  Content-Length=0,
  x-log-signaturemethod= hmac-sha1,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
```

- Sample responses

```
HTTP/1.1 200 OK
Header :
{
  Date=Wed, 11 Nov 2015 07:53:30 GMT,
  Content-Length=107,
  x-log-requestid=5642F3FA99248C817B01352D,
  Connection=close,
  Content-Type=application/json,
  Server=nginx/1.6.1
}
Body :
{
  "logstoreName": test-logstore,
  "ttl": 1,
  "shardCount": 2,
  "createTime": 1447833064,
  "lastModifyTime": 1447833064
}
```

4.3.7.5. ListLogstore

Lists all Logstores in a specified project.

Request syntax

```
GET /logstores HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The ListLogstore API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
offset	Integer	No	The start position of the records to return. Default value: 0.
size	Integer	No	The maximum number of entries to return on each page. Default and maximum value: 500.
logstoreName	String	Yes	The Logstore name that is used as a filtering condition. Fuzzy match is used.

Response elements

- Response headers

The ListLogstore API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After a ListLogstore request succeeds, the response body contains a list of all Logstores in the specified project. The following table describes the response parameters.

Parameter	Type	Description
count	Integer	The number of returned Logstores.
total	Integer	The total number of Logstores.
logstores	String array	The name list of returned Logstores.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	Invalid parameter size, (0.6.0]
400	InvalidLogStoreQuery	logstore Query is invalid

Examples

- Sample requests

```
GET /logstores HTTP/1.1
Header:
{
  x-log-apiversion=0.6.0,
  Authorization=LOG <yourAccessKeyId>:<yourSignature>,
  Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
  Date=Wed, 11 Nov 2015 08:09:39 GMT,
  Content-Length=0,
  x-log-signaturemethod=hmac-sha1,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
```

- Sample responses

```
HTTP/1.1 200 OK
Header:
{
  Date=Wed, 11 Nov 2015 08:09:39 GMT,
  Content-Length=52,
  x-log-requestid=5642F7C399248C8D7B01342F,
  Connection=close,
  Content-Type=application/json,
  Server=nginx/1.6.1
}
Body:
{
  "count":1,
  "logstores":["test-logstore"],
  "total":1
}
```

4.3.7.6. ListShards

Lists all available shards in a Logstore.

Request syntax

```
GET /logstores/<logstorename>/shards HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The ListShards API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.

Response elements

- Response headers

The ListShards API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response elements

The response is an array that consists of shards.

```
[
  {
    "shardID": 0,
    "status": "readwrite",
    "inclusiveBeginKey": "00000000000000000000000000000000",
    "exclusiveEndKey": "80000000000000000000000000000000",
    "createTime": 1453949705
  },
  {
    ...
  },
  {
    ...
  }
]
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

 **Note** The {logstoreName} variable in the preceding error message is replaced with a specific Logstore name.

Examples

- Sample requests

```
GET /logstores/sls-test-logstore/shards
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Thu, 12 Nov 2015 03:40:31 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

- Sample responses

```

Header:
{
  "content-length": "57",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Thu, 12 Nov 2015 03:40:31 GMT",
  "content-type": "application/json",
  "x-log-requestid": "56440A2F99248C050600C74C"
}
Body :
[
  {
    "shardID": 1,
    "status": "readwrite",
    "inclusiveBeginKey": "00000000000000000000000000000000",
    "exclusiveEndKey": "80000000000000000000000000000000",
    "createTime": 1453949705
  },
  {
    "shardID": 2,
    "status": "readwrite",
    "inclusiveBeginKey": "80000000000000000000000000000000",
    "exclusiveEndKey": "ffffffffffffffffffffffffffffffff",
    "createTime": 1453949705
  },
  {
    "shardID": 0,
    "status": "readonly",
    "inclusiveBeginKey": "00000000000000000000000000000000",
    "exclusiveEndKey": "ffffffffffffffffffffffffffffffff",
    "createTime": 1453949705
  }
]

```

4.3.7.7. SplitShard

Splits a specified shard in the read/write state.

Request syntax

```

POST /logstores/<logstorename>/shards/<shardid>? action=split&key=<splitkey> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

```

Request elements

- Request headers

The SplitShard API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
shardid	Integer	Yes	The ID of the shard.
splitkey	String	Yes	The location where the shard is split.

Response elements

- Response headers

Content-Type: application/json

The SplitShard API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The response is an array that consists of three shards. The first shard is the original shard. The second and third shards are the new shards that are created by splitting the original shard.

```
[
  {
    "shardID": 33,
    "status": "readonly",
    "inclusiveBeginKey": "ee000000000000000000000000000000",
    "exclusiveEndKey": "ffffffffffffffffffffffffffffffff",
    "createTime": 1453949705
  },
  {
    "shardID": 163,
    "status": "readwrite",
    "inclusiveBeginKey": "ee000000000000000000000000000000",
    "exclusiveEndKey": "ef000000000000000000000000000000",
    "createTime": 1453949705
  },
  {
    "shardID": 164,
    "status": "readwrite",
    "inclusiveBeginKey": "ef000000000000000000000000000000",
    "exclusiveEndKey": "ffffffffffffffffffffffffffffffff",
    "createTime": 1453949705
  }
]
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	invalid mid hash
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

 **Note** The {logstoreName} variable in the preceding error message is replaced with a specific Logstore name.

Examples

• Sample requests

```
POST /logstores/logstorename/shards/33? action=split&key=ef000000000000000000000000000000
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou.sls.aliyuncs.com",
  "Date": "Thu, 12 Nov 2015 03:40:31 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

• Sample responses

```
Header:
{
  "content-length": "57",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Thu, 12 Nov 2015 03:40:31 GMT",
  "content-type": "application/json",
  "x-log-requestid": "56440A2F99248C050600C74C"
}
Body :
[
  {
    "shardID": 33,
    "status": "readonly",
    "inclusiveBeginKey": "ee000000000000000000000000000000",
    "exclusiveEndKey": "ffffffffffffffffffffffffffffffff",
    "createTime": 1453949705
  },
  {
    "shardID": 163,
    "status": "readwrite",
    "inclusiveBeginKey": "ee000000000000000000000000000000",
    "exclusiveEndKey": "ef000000000000000000000000000000",
    "createTime": 1453949705
  },
  {
    "shardID": 164,
    "status": "readwrite",
    "inclusiveBeginKey": "ef000000000000000000000000000000",
    "exclusiveEndKey": "ffffffffffffffffffffffffffffffff",
    "createTime": 1453949705
  }
]
```

4.3.7.8. MergeShards

Merges two adjacent shards that are in the read/write state. If you specify the ID of a shard in the request, Log Service automatically finds the adjacent shard.

Request syntax

```
POST /logstores/<logstorename>/shards/<shardid>? action=merge HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers
The MergeShards API operation does not have special request headers. For information about common request headers of Log Service API operations, see Common request headers.
- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
shardid	Integer	Yes	The ID of the shard.

Response elements

- Response headers
Content-Type: application/json
The MergeShards API operation does not have special response headers. For information about common response headers of Log Service API operations, see Common response headers.
- Response parameters
An array that consists of three shards is returned. The first shard is the merged shard. The other two shards are the original shards.

```
[
  {
    'shardID': 167,
    'status': 'readwrite',
    'inclusiveBeginKey': 'e0000000000000000000000000000000',
    'createTime': 1453953105,
    'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
  },
  {
    'shardID': 30,
    'status': 'readonly',
    'inclusiveBeginKey': 'e0000000000000000000000000000000',
    'createTime': 0,
    'exclusiveEndKey': 'e7000000000000000000000000000000'
  },
  {
    'shardID': 166,
    'status': 'readonly',
    'inclusiveBeginKey': 'e7000000000000000000000000000000',
    'createTime': 1453953073,
    'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
  }
]
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	can not merge the last shard
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

 **Note** The {Name} variable in the preceding error message is replaced with a specific Logstore name.

Examples

• Sample requests

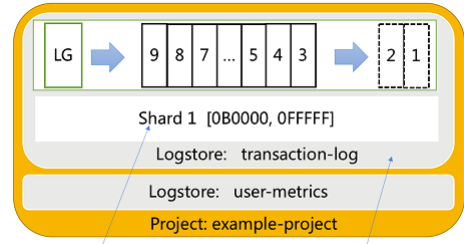
```
POST /logstores/logstorename/shards/30? action=merge
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou.sls.aliyuncs.com",
  "Date": "Thu, 12 Nov 2015 03:40:31 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

• Sample responses

```
Header:
{
  "content-length": "57",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Thu, 12 Nov 2015 03:40:31 GMT",
  "content-type": "application/json",
  "x-log-requestid": "56440A2F99248C050600C74C"
}
Body :
[
  {
    'shardID': 167,
    'status': 'readwrite',
    'inclusiveBeginKey': 'e0000000000000000000000000000000',
    'createTime': 1453953105,
    'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
  },
  {
    'shardID': 30,
    'status': 'readonly',
    'inclusiveBeginKey': 'e0000000000000000000000000000000',
    'createTime': 0,
    'exclusiveEndKey': 'e7000000000000000000000000000000'
  },
  {
    'shardID': 166,
    'status': 'readonly',
    'inclusiveBeginKey': 'e7000000000000000000000000000000',
    'createTime': 1453953073,
    'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
  }
]
```

4.3.7.9. GetCursor

Queries a cursor based on a point in time. You can use a cursor to obtain the location of a specific log. The following figure shows the relationships among a cursor, project, Logstore, and shard.



- Each project has multiple Logstores.
- Each Logstore has multiple shards.
- You can use a cursor to obtain the location of a specific log.

Request syntax

The request syntax of the Get Cursor API operation is as follows:

```
GET /logstores/ay42/shards/2? type=cursor&from=1402341900 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
```

Request elements

- Request headers
The Get Cursor API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
shard	String	Yes	The name of the shard.
type	String	Yes	This parameter must be set to cursor.
from	String	Yes	The time that is used to query a cursor. Set the value to a Unix timestamp or a string such as <code>begin</code> or <code>end</code> .

You can use the `from` parameter to locate the logs within the lifecycle of a shard in a Logstore. If the lifecycle of the Logstore is `[begin_time, end_time)` and the `from` parameter is set to `from_time`, the response is returned as follows:

- `from_time ≤ begin_time` or `from_time = "begin"` : A cursor is returned. The cursor corresponds to the log whose timestamp is `begin_time`.
- `from_time ≥ end_time` or `from_time = "end"` : A cursor is returned. The cursor corresponds to the next log to be written at the current time (No data exists in the current cursor location).
- `from_time > begin_time` and `from_time < end_time` : A cursor is returned. The cursor corresponds to the first packet that Log Service receives later than or at `from_time`.

Note The lifecycle of a Logstore is specified by the `ttl` parameter of the Logstore. For example, the current time is 2018-11-11 09:00:00 and the `ttl` parameter is set to 5. The time interval of consumable data in each shard is `[2018-11-05 09:00:00, 2018-11-11 09:00:00)`. The server time is used.

Response elements

• Response headers

The Get Cursor API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

• Response parameters

```
{
  "cursor": "MTQ0NzI5OTYwNjg5NjYzMjM1Ng=="
}
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	Logstore {Name} does not exist
400	ParameterInvalid	Parameter From is not valid
400	ShardNotExist	Shard {ShardID} does not exist
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	the logstore has no shard

Examples

• Sample requests

```
GET /logstores/sls-test-logstore/shards/0? type=cursor&from=begin
Header:
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Thu, 12 Nov 2015 03:56:57 GMT",
  "x-log-apiversion": "0.6.0",
  "Content-Type": "application/json",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

• Sample responses

```
Header:
{
  "content-length": "41",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Thu, 12 Nov 2015 03:56:57 GMT",
  "content-type": "application/json",
  "x-log-requestid": "56440E0999248C070600C6AA"
}
Body:
{
  "cursor": "MTQ0NzI5OTYwNjg5NjYzMjM1Ng=="
}
```

4.3.7.10. GetCursorTime

Queries the server time based on the cursor.

Request syntax

```
GET /logstores/{logstoreName}/shards/{shard}? cursor={cursor}&type=cursor_time HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The GetCursorTime API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
shard	Integer	Yes	The ID of the shard.
cursor	String	Yes	The cursor that is used to obtain the corresponding timestamp.
type	String	Yes	Set the value to cursor_time.

Response elements

- Response headers

The GetCursorTime API does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

```
{
  "cursor_time": 1554260243
}
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	Logstore {Name} does not exist
400	ParameterInvalid	Parameter From is not valid
400	ShardNotExist	Shard {ShardID} does not exist
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	the logstore has no shard

Examples

- Sample requests

```
GET /logstores/internal-operation_log/shards/0? cursor=MTU0NzQ3MDY4MjM3NjUxMzQ0Ng%3D%3D&type=cursor_time HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-hangzhou.sls.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Wed, 03 Apr 2019 02:57:23 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```
HTTP/1.1 200
Server: nginx
Content-Type: application/json
Content-Length: 26
Connection: close
Access-Control-Allow-Origin: *
Date: Wed, 03 Apr 2019 02:57:23 GMT
x-log-requestid: 5CA42113D2F00378A74B9051
{
  "cursor_time": 1554260243
}
```

4.3.7.11. PullLogs

Pulls logs from a Logstore. You must specify a shard when you call this API operation.

Only Protocol Buffer log data can be pulled from a Logstore. For more information, see [Protocol Buffer format](#).

In scenarios that involve the Storm architecture, election and collaborative consumption can be performed through LoghubClientLib.

Request syntax

```
GET /logstores/ay42/shards/0? type=logs&cursor=MTQ0NzMyOTQwMTEwMjEzMDkwNA==&count=100 HTTP/1.1
Accept: application/x-protobuf
Accept-Encoding: lz4
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

- Request header fields
 - Accept: application/x-protobuf
 - Accept-Encoding: lz4, deflate, or ""
 - For information about common request headers of Log Service API operations, see [Common request headers](#).
- Request parameters

Parameter	Type	Required	Description
type	String	Yes	Set the value to logs.
cursor	String	Yes	The cursor that identifies where data reading starts. This indicates the start point.
count	Integer	Yes	The number of returned log groups. Valid values: 1 to 1000.

Response parameters

- Response header fields
 - x-log-cursor: The next cursor of the currently read data.
 - x-log-count: The number of currently returned logs.
 - For information about common response headers of Log Service API operations, see [Common response headers](#).
- Response elements

Serialized data is returned in the Protocol Buffer format. The data may be compressed.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	Logstore {Name} does not exist
400	ParameterInvalid	Parameter Cursor is not valid
400	ParameterInvalid	ParameterCount should be in [0-1000]
400	ShardNotExist	Shard {ShardID} does not exist
400	InvalidCursor	this cursor is invalid
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
Read data from shard 0.
GET /logstores/sls-test-logstore/shards/0? cursor=MTQ0NzMyOTQwMTEwMjEzMDkwNA==&count=1000&type=log
Header:
{
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "x-log-bodyrawsize": 0,
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "x-log-apiversion": "0.6.0",
  "Host": "ali-test-project.cn-hangzhou-failover-intranet.sls.aliyuncs.com",
  "x-log-signaturemethod": "hmac-sha1",
  "Accept-Encoding": "lz4",
  "Content-Length": 0,
  "Date": "Thu, 12 Nov 2015 12:03:17 GMT",
  "Content-Type": "application/x-protobuf",
  "accept": "application/x-protobuf"
}
```

- Sample responses

```
Header:
{
  "x-log-count" : "1000",
  "x-log-requestid" : "56447FB20351626D7C000874",
  "Server" : "nginx/1.6.1",
  "x-log-bodyrawsize" : "34121",
  "Connection" : "close",
  "Content-Length" : "4231",
  "x-log-cursor" : "MTQ0NzMyOTQwMTEwMjEzMDkwNA==",
  "Date" : "Thu, 12 Nov 2015 12:01:54 GMT",
  "x-log-compresstype" : "lz4",
  "Content-Type" : "application/x-protobuf"
}
Body:
The <log group list in the Protocol Buffer format> after the compression.
```

- Page flip

To flip the page (get the next token) without returning data, you can send HTTP HEAD requests.

4.3.7.12. PostLog

Writes log data to a Logstore.

Only Protocol Buffer log data can be written to a Logstore. For more information, see [Protocol Buffer format](#).

 **Note** Log data is written in the form of log groups.

Log data can be written to a Logstore in either of the following modes:

- **LoadBalance**: In this mode, Log Service automatically writes logs to all writable shards in a Logstore. This mode provides high availability (SLA: 99.95%) in data writes. It applies to scenarios where data writes and reads are irrelevant to shards, for example, where ordering is not considered.
- **KeyHash**: The key field is added to the URL parameter. This determines which shard to store the log data. This field is optional. If you do not specify this field, Log Service automatically writes log data in the LoadBalance mode. For example, Log Service can hash an instance as a producer to a fixed shard based on the name of the instance. This ensures that data is written and consumed by using a strict order in the shard. The key is included only in the MD5 value range of a shard at a time even when the shard is merged or split. For more information, see [Shards](#).

Protocol Buffer log data

The following table describes the fields of Protocol Buffer log data. For more information, see [Data model](#) and [Data encoding](#).

- Log

Field	Type	Required	Description
Time	Integer	Yes	The time when the log was generated.
Contents	List	Yes, this field must contain at least one element.	The list of log fields. For information about the field elements, see the Content field table.

- Content

Field	Type	Required	Description
Key	String	Yes	The name of the custom key.
Value	String	Yes	The value of the custom key.

- LogTag

Field	Type	Required	Description
Key	String	Yes	The name of the custom key.
Value	String	Yes	The value of the custom key.

- LogGroup

Field	Type	Required	Description
Logs	List	Yes	A list of logs. For information about the field elements, see the Log field table.
Topic	String	Yes	A user-defined field in a log. This field identifies a group of logs.
Source	String	Yes	The source of the log, for example, the IP address of the machine where the log is generated.
LogTags	List	Yes	The list of log tags. For information about the field elements, see the LogTags field table.

Request syntax

- LoadBalance mode

```
POST /logstores/<logstorename>/shards/lb HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/x-protobuf
Content-Length: <Content Length>
Content-MD5: <Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-bodyrawsize: <BodyRawSize>
x-log-compresstype: lz4
x-log-signaturemethod: hmac-sha1
<Compressed log data in the Protocol Buffer format>
```

- KeyHash mode

```
POST /logstores/<logstorename>/shards/route? key=14d2f850ad6ea48e46e4547edbbb27e0
Authorization: <AuthorizationString>
Content-Type: application/x-protobuf
Content-Length: <Content Length>
Content-MD5: <Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-bodyrawsize: <BodyRawSize>
x-log-compresstype: lz4
x-log-signaturemethod: hmac-sha1
<Compressed log data in the Protocol Buffer format>
```

Request elements

- Request headers

For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstorename	String	Yes	The name of the Logstore to which logs are written.

Response elements

- Response headers

The PutLogs API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

No response element is returned after the request succeeds.

Additional considerations

- You can call the PutLogs API operation to write a maximum of 3 MB log group data or 4,096 logs each time. The size of values in each log cannot be greater than 1 MB. If these limits are exceeded, the request will fail and no log data will be written.
- Log Service checks the format of log data written by the PutLogs API operation. If a log does not meet the format requirements, the request will fail and no log data will be written.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message	Description
400	PostBodyInvalid	Protobuffer content cannot be parsed.	The error message returned because the logs in the Protocol Buffer format cannot be parsed.
400	InvalidTimestamp	Invalid timestamps are in logs.	The error message returned because the logs contain invalid timestamps.
400	InvalidEncoding	Non-UTF8 characters are in logs.	The error message returned because the logs contain non-UTF-8 characters.
400	InvalidKey	Invalid keys are in logs.	The error message returned because the logs contain invalid keys.
400	PostBodyTooLarge	Logs must be less than or equal to 3 MB and 4096 entries.	The error message returned because the size of logs is greater than 3 MB or the number of logs exceeds 4,096.
400	PostBodyUncompressError	Failed to decompress logs.	The error message returned because the logs fail to be decompressed.
499	PostBodyInvalid	The post data time is out of range	The error message returned because the log timestamp is not within the valid range: [-7 × 24 hours, + 15 minutes].

HTTP status code	Error code	Error message	Description
404	LogStoreNotExist	logstore {Name} does not exist.	The error message returned because the specified Logstore does not exist.

 **Note** The {Name} variable in the preceding error message is replaced with a specific Logstore name.

Examples

• Sample requests

```
POST /logstores/sls-test-logstore
{
  "Content-Length": 118,
  "Content-Type": "application/x-protobuf",
  "x-log-bodyrawsize": 1356,
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Content-MD5": "6554BD042149C844761C2C094A8FECCE",
  "Date": "Thu, 12 Nov 2015 06:54:26 GMT",
  "x-log-apiversion": "0.6.0",
  "x-log-compresstype": "lz4",
  "x-log-signaturemethod": "hmac-sha1",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
<Binary data obtained after logs in the Protocol Buffer format are compressed with the LZ4 algorithm>
```

• Sample responses

```
Header
{
  "date": "Thu, 12 Nov 2015 06:53:03 GMT",
  "connection": "close",
  "x-log-requestid": "5644160399248C060600D216",
  "content-length": "0",
  "server": "nginx/1.6.1"
}
```

4.3.7.13. GetShipperStatus

You can call this operation to query the LogShipper task status.

Request syntax

```
GET /logstores/{logstoreName}/shipper/{shipperName}/tasks? from=1448748198&to=1448948198&status=success&offset=0&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

The request parameters are described in .

GetShipperStatus request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The Logstore name, which must be unique in the same project.
shipperName	String	Yes	The name of the log shipping rule, which is unique in the same Logstore.
from	Integer	Yes	The creation time range of a LogShipper task.
to	Integer	Yes	The creation time range of a LogShipper task.
status	String	No	The default value is null, indicating that tasks in any states will be returned. Currently, tasks in the successful, running, or failed state are returned.
offset	Integer	No	The starting number of LogShipper tasks within a specified time range. The default value is 0.
size	Integer	No	The number of LogShipper tasks within a specified time range. The default value is 100. The maximum value is 500.

Request header

The GetShipperStatus API does not have any special request headers. For more information about the common request headers of Log Service APIs, see [Common request headers](#).

Response header

The GetShipperStatus API does not have any special response headers. For more information about the common response headers of Log Service APIs, see [Common response headers](#).

Response elements

When a request is successful, the response body contains the list of specified LogShipper tasks.

```
{
  "count" : 10,
  "total" : 20,
  "statistics" : {
    "running" : 0,
    "success" : 20,
    "fail" : 0
  }
  "tasks" : [
    {
      "id" : "abcdefghijk",
      "taskStatus" : "success",
      "taskMessage" : "",
      "taskCreateTime" : 1448925013,
      "taskLastDataReceiveTime" : 1448915013,
      "taskFinishTime" : 1448926013
    }
  ]
}
```

The GetShipperStatus response elements are described in .

GetShipperStatus response elements

Name	Type	Description
count	Integer	The number of returned tasks.
total	Integer	The total number of tasks within a specified range.
statistics	JSON	The statistics of task status within a specified range. For more information, see the following table.
tasks	Array	The details of a LogShipper task within a specified range. For more information, see the following table.

For more information about the statistics, see .

Statistic parameter

Name	Type	Description
running	Integer	The number of running tasks within a specified range.
successful	Integer	The number of successful tasks within a specific range.
failed	Integer	The number of failed tasks within a specific range.

For more information about the tasks, see .

Tasks parameters

Name	Type	Description
id	String	The unique ID of a LogShipper task.
taskStatus	String	The LogShipper task status, which can be running, successful, or failed.
taskMessage	String	The error message returned when a LogShipper task fails.
taskCreateTime	Integer	The creation time of a LogShipper task.
taskLastDataReceiveTime	Integer	The latest log data time in a LogShipper task.
taskFinishTime	Integer	The end time of a LogShipper task.

Error codes

As described in , in addition to the [Common error codes](#) of Log Service APIs, the API may also return the following special error codes.

Error codes specific to GetShipperStatus

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ShipperNotExist	shipper {logstoreName} does not exist
500	InternalServerError	internal server error
400	ParameterInvalid	start time must be earlier than end time
400	ParameterInvalid	only support query last 48 hours task status
400	ParameterInvalid	status only contains success/running/fail

Detailed description

You can only query LogShipper task status within the last 24 hours.

Examples

Sample requests:

```
GET /logstores/test-logstore/shipper/test-shipper/tasks? from=1448748198&to=1448948198&status=success&offset=0&size=100 HTTP/1.1
Header:
{
  x-log-apiversion=0.6.0,
  Authorization=LOG 94to3z4l8yupi6ikawqgd370:wFc13ohVJupCi0ZFxD0x4IA68A=,
  Host=ali-test-project.regionid.example.com,
  Date=Wed, 11 Nov 2015 08:28:19 GMT,
  Content-Length=55,
  x-log-signaturemethod= hmac-sha1,
  Content-MD5=757C60FC41CC7D3F60B88E0D916D051E,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
```

Sample responses:

```
HTTP/1.1 200 OK
Header:
{
  Date=Wed, 11 Nov 2015 08:28:20 GMT,
  Content-Length=0,
  x-log-requestid=5642FC2399248C8F7B0145FD,
  Connection=close,
  Server=nginx/1.6.1
}
Body:
{
  "count" : 10,
  "total" : 20,
  "statistics" : {
    "running" : 0,
    "success" : 20,
    "fail" : 0
  }
  "tasks" : [
    {
      "id" : "abcdefghijk",
      "taskStatus" : "success",
      "taskMessage" : "",
      "taskCreateTime" : 1448925013,
      "taskLastDataReceiveTime" : 1448915013,
      "taskFinishTime" : 1448926013
    }
  ]
}
```

4.3.7.14. RetryShipperTask

You can call this operation to rerun the failed LogShipper tasks.

Request syntax

```
PUT /logstores/{logstoreName}/shipper/{shipperName}/tasks HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
["task-id-1", "task-id-2", "task-id-2"]
```

Request parameters

The request parameters are described in .

RetryShipperTask request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The Logstore name, which must be unique in the same project.
shipperName	String	Yes	The name of the log shipping rule, which is unique in the same Logstore.

Request header

The RetryShipperTask API does not have any special request headers. For more information about the common request headers of Log Service APIs, see [Common request headers](#).

Response header

The RetryShipperTask API does not have any special response headers. For more information about the common response headers of Log Service APIs, see [Common response headers](#).

Response elements

The returned HTTP status code is 200.

Error codes

As described in , in addition to the [Common error codes](#) of Log Service APIs, the API may also return the following special error codes.

Error codes specific to RetryShipperTask

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ShipperNotExist	shipper {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	only allow retry 10 tasks every time

Detailed description

A maximum of 10 failed posting tasks can be re-executed at a time.

Examples

Sample requests:

```
PUT /logstores/test-logstore/shipper/test-shipper/tasks HTTP/1.1
Header:
{
  x-log-apiversion=0.6.0,
  Authorization=LOG 94to3z4l8yupi6ikawgqd370:wFc13ohVJupCi0ZFxD0x4IA68A=,
  Host=ali-test-project.regionid.example.com,
  Date=Wed, 11 Nov 2015 08:28:19 GMT,
  Content-Length=55,
  x-log-signaturemethod=hmac-sha1,
  Content-MD5=757C60FC41CC7D3F60B88E0D916D051E,
  User-Agent=sls-java-sdk-v-0.6.0,
  Content-Type=application/json
}
Body :
["task-id-1", "task-id-2", "task-id-2"]
```

Sample responses:

```
HTTP/1.1 200 OK
Header:
{
  Date=Wed, 11 Nov 2015 08:28:20 GMT,
  Content-Length=0,
  x-log-requestid=5642FC2399248C8F7B0145FD,
  Connection=close,
  Server=nginx/1.6.1
}
```

4.3.7.15. GetLogs

Queries logs in a Logstore of a specified project, or queries the logs that meet the specified conditions.

After a log is written to a Logstore, you can call the GetHistograms and GetLogs API operations to query the log. However, the query has a latency, which varies with the type of the log. Based on the log timestamp, Log Service classifies logs into the following two types:

- **Real-time data:** For this type, the difference between the time in a log and the current time on the server is [-180 seconds, 900 seconds]. For example, if the log time is UTC 2014-09-25 12:03:00 and the server receives the log at UTC 2014-09-25 12:05:00, the log is regarded as real-time data.
- **Historical data:** For this type, the difference between the time in a log and the current time on the server is [-7 × 86,400 seconds, -180 seconds]. For example, if the log time is UTC 2014-09-25 12:00:00 and the server receives the log at UTC 2014-09-25 12:05:00, the log is regarded as historical data.

Real-time data can be queried with a maximum latency of three seconds after being written to a Logstore. In most cases, 99.9% of real-time data can be queried in one second.

Request syntax

The request syntax of the GetLogs API operation is as follows:

```
GET /logstores/<logstorename>? type=log&topic=<logtopic>&from=<starttime>&to=<endtime>&query=<querystring>&line=<linenum>&offset=<startindex>&reverse
=<ture|false> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-bodyrawsize: 0
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- **Request headers**

The GetLogs API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- **Request parameters**

Parameter	Type	Required	Description
logstorename	String	Yes	The name of the Logstore whose logs you want to query.
type	String	Yes	The type of data in the Logstore. The value of this parameter must be log.
from	Integer	Yes	The start time of the time range during which the logs to be queried are generated. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.
to	Integer	Yes	The end time of the time range during which the logs to be queried are generated. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.
topic	String	No	The topic of the logs to be queried.
query	String	No	The query statement. see <i>User Guide</i> . The path is Query and analysis > Query syntax and functions > Query syntax .
line	Integer	No	The maximum number of logs to return for the request. Value range: 0 to 100. Default value: 100.
offset	Integer	No	The start point of the returned log for the request. Valid values: 0 and positive integers. Default value: 0.
reverse	Boolean	No	Specifies whether to return logs in reverse order based on the log timestamp, accurate to seconds. If the value is true, logs are returned in reverse order. If the value is false, logs are returned in regular order. Default value: false.

Response elements

- **Response headers**

The GetLogs API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

The response header contains a special element that indicates whether the returned results of the request are complete. The following table describes the response parameters.

Parameter	Type	Description
x-log-progress	String	The status of the query results. Valid values: Incomplete and Complete.
x-log-count	Integer	The total number of logs in the current query results.

- **Response parameters**

After a GetLogs request succeeds, the response body contains log data hits in the query. If a large number of logs (measured in TBs) are queried, the response of the GetLogs operation may be incomplete. The response body of the GetLogs operation is an array. Each element in the array is a log. The following table describes the elements in the array.

Element	Type	Description
__time__	Integer	The log timestamp. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.

Element	Type	Description
__Source__	String	The log source, which is specified when logs are written.
[content]	Key-value pair	The original content of logs.

Additional considerations

- The time range defined by the from and to request parameters is left closed and right open. This means that the time range includes the start time, but does not include the end time. If the from and to parameters for a time range have the same value, the time range is invalid and the function will return an error.
- If the number of logs in the Logstore significantly changes, Log Service cannot predict how many times this API operation needs to be called to obtain the complete results. In this case, you must check the value of the x-log-progress parameter in the returned results of each request. Based on the value, you can decide whether to repeatedly call this API operation to obtain the complete results.

 **Note** The same number of query compute units (CUs) is consumed each time you call this API operation.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message	Description
404	LogStoreNotExist	logstore {Name} does not exist	The error message returned because the specified Logstore does not exist.
400	InvalidTimeRange	request time range is invalid	The error message returned because the specified time interval of the request is invalid.
400	InvalidQueryString	query string is invalid	The error message returned because the specified query string of the request is invalid.
400	InvalidOffset	offset is invalid	The error message returned because the specified offset parameter of the request is invalid.
400	InvalidLine	line is invalid	The error message returned because the specified line parameter of the request is invalid.
400	InvalidReverse	Reverse value is invalid	The error message returned because the specified reverse parameter is invalid.
400	IndexConfigNotExist	logstore without index config	The error message returned because the index feature has not been enabled for the Logstore.

 **Note** The {name} variable in the preceding error message is replaced with a specific Logstore name.

Example

In this example, the logs with the groupA topic are queried in a Logstore named app_log. The Logstore belongs to a project named big-game in the China (Hangzhou) region. The time range is from 2014-09-01 00:00:00 to 2014-09-01 22:00:00 and the keyword is "error". The query starts from the beginning of the time range. A maximum of 20 logs can be returned.

Sample requests

```
GET /logstores/app_log? type=log&topic=groupA&from=1409529600&to=1409608800&query=error&line=20&offset=0 HTTP/1.1
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.4.0
x-log-signaturemethod: hmac-sha1
```

Sample responses

```

HTTP/1.1 200 OK
Content-MD5: 36F9F7F0339BEAF571581AF1B0AAAFB5
Content-Type: application/json
Content-Length: 269
Date: Wed, 3 Sept. 2014 08:33:47 GMT
x-log-requestid: efag01234-12341-15432f
x-log-progress : Complete
x-log-count : 10000
x-log-processed-rows: 10000
x-log-elapsed-millisecond:5
{
  "progress": "Complete",
  "count": 2,
  "logs": [
    {
      "__time__": 1409529660,
      "__source__": "10.237.0.17",
      "Key1": "error",
      "Key2": "Value2"
    },
    {
      "__time__": 1409529680,
      "__source__": "10.237.0.18",
      "Key3": "error",
      "Key4": "Value4"
    }
  ]
}

```

In the sample response, the x-log-progress value is Complete. This indicates that the log query is completed and the returned results are complete. Two logs meet the specified query conditions and are displayed as the value of the logs parameter. If the x-log-progress value is Incomplete, you must repeat the request to obtain the complete results.

4.3.7.16. GetHistograms

Queries the distribution of logs in a Logstore within a specified project, or queries the distribution of logs that meet the specified conditions.

After a log is written to a Logstore, you can call the GetHistograms and GetLogs API operations to query the log. However, the query has a latency, which varies with the type of the log. Based on the log timestamp, Log Service classifies logs into the following two types:

- Real-time data: For this type, the difference between the time in a log and the current time on the server is within the interval [-180 seconds, 900 seconds]. For example, if the log time is UTC 2014-09-25 12:03:00 and the server receives the log at UTC 2014-09-25 12:05:00, the log is regarded as real-time data. The latency for real-time data is three seconds, from the time when the data is written to the Logstore to the time when the data can be queried.
- Historical data: For this type, the difference between the time in a log and the current time on the server is [-7 × 86,400 seconds, -180 seconds). For example, if the log time is UTC 2014-09-25 12:00:00 and the server receives the log at UTC 2014-09-25 12:05:00, the log is regarded as historical data.

Request syntax

The request syntax of the GetHistograms API operation is as follows:

```

GET /logstores/<logstorename>? type=histogram&topic=<logtopic>&from=<starttime>&to=<endtime>&query=<querystring> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-bodyrawsize: 0
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

```

Request elements

- Request headers

The GetHistograms API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstorename	String	Yes	The name of the Logstore whose logs you want to query.
type	String	Yes	The type of data in the Logstore. The value of this parameter must be histogram.
from	Integer	Yes	The start time of the time range during which the logs to be queried are generated. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.
to	Integer	Yes	The end time of the time range during which the logs to be queried are generated. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.
topic	String	No	The topic of the logs to be queried.

Parameter	Type	Required	Description
query	String	No	The query statement. For more information about the query syntax, see <i>User Guide</i> . The path is Query and analysis > Query syntax and functions > Query syntax .

Response elements

- Response headers

The GetHistograms API does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After a GetHistograms request succeeds, the response body contains the distribution of the log hits returned by the query along the timeline. The response results evenly divide the time range into several subintervals and returns the number of log hits for each subinterval. The following table describes the response parameters.

Parameter	Type	Description
progress	String	The status of the query results. Valid values: Incomplete and Complete.
count	Integer	The total number of logs in the current query results.
histograms	Array	The distribution of the current query results in the subintervals. For information about the parameters of each element in the histograms array, see the following table.

This table describes the elements in the histograms parameter.

Element	Type	Description
from	Integer	The start time of the subinterval. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.
to	Integer	The end time of the subinterval. This parameter indicates the number of seconds that have elapsed since the Unix epoch 00:00:00 UTC on January 1, 1970.
count	Integer	The number of logs that meet the query conditions for this subinterval in the current query results.
progress	String	Indicates whether the current query results in the subinterval are complete. Valid values: Incomplete and Complete.

Additional considerations

- The preceding time ranges or subintervals are left closed and right open. This means that each time range or subinterval includes the start time, but does not include the end time. If the from and to parameters for a time range have the same value, the time range is invalid and the function will return an error.
- The time range is consistently divided into subintervals in the response. If the time range is unchanged, the subintervals divided in the response is also unchanged.
- If the number of logs in the Logstore significantly changes, Log Service cannot predict how many times this API operation needs to be called to obtain the complete results. In this case, you must check the value of the progress parameter in the returned results of each request. Based on the value, you can decide whether to repeatedly call this API operation to obtain the complete results.

 **Note** The same number of query compute units (CUs) is consumed each time you call this API operation.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message	Description
404	LogStoreNotExist	logstore {Name} does not exist	The error message returned because the specified Logstore does not exist.
400	InvalidTimeRange	request time range is invalid	The error message returned because the specified time interval of the request is invalid.
400	InvalidQueryString	query string is invalid	The message returned when the query string of the request is invalid.

 **Note** The {name} variable in the preceding error message is replaced with a specific Logstore name.

Example

In this example, the distribution of logs with the groupA topic is queried in a Logstore named app_log. The Logstore belongs to a project named big-game in the China (Hangzhou) region. The time range is from 2014-09-01 00:00:00 to 2014-09-01 22:00:00 and the keyword is "error".

- Sample requests

```
GET /logstores/app_log? type=histogram&topic=groupA&from=1409529600&to=1409608800&query=error HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

- Sample responses

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-MD5: E6AD9C21204868C2DE84EE3808AAA8C8
Content-Type: application/json
Date: Wed, 3 Sept. 2014 08:33:47 GMT
Content-Length: 232
x-log-requestid: efag01234-12341-15432f
[
  {
    {
      "from": 1409529600,
      "to": 1409569200,
      "count": 2,
      "progress": "Complete"
    },
    {
      "from": 1409569200,
      "to": 1409608800,
      "count": 1,
      "progress": "Incomplete"
    }
  ]
```

In this sample response, the server divides the entire histogram into two equal time ranges: [2014-09-01 00:00:00, 2014-09-01 11:00:00) and [2014-09-01 11:00:00, 2014-09-01 22:00:00). The first query returns incomplete data because the amount of data exceeds the limit. The sample response indicates three hits. The results for the time range [2014-09-01 00:00:00 2014-09-01 11:00:00) are complete, including two hits. The results for the other time range are incomplete, including one hit. To obtain the complete results, you must repeatedly call the preceding sample request until the progress value in the response changes to Complete. The sample response is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-MD5: E6AD9C21204868C2DE84EE3808AAA8C8
Content-Type: application/json
Date: Wed, 3 Sept. 2014 08:33:48 GMT
Content-Length: 232
x-log-requestid: afag01322-1e241-25432e
[
  {
    {
      "from": 1409529600,
      "to": 1409569200,
      "count": 2,
      "progress": "Complete"
    },
    {
      "from": 1409569200,
      "to": 1409608800,
      "count": 2,
      "progress": "complete"
    }
  ]
```

4.3.7.17. UpdateIndex

Updates the indexes of a Logstore.

Request syntax

```
PUT /logstores/<logstoreName>/index HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
{
  "line": <full text index>,
  "keys": <key-value index>,
  "ttl": <ttl>
}
```

Request elements

- Request headers

The UpdateIndex API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
TTL	Integer	No	The time to live (TTL) of the indexes. Valid values: 7, 30, and 90. Default value: 0.
keys	Object	No	The key-value index, where the key refers to the field name and the value refers to the field value.
line	Object	No	The full-text index.

You must specify the keys parameter or the line parameter, or both. The following table describes the full-text index attributes.

Attribute	Type	Required	Description
token	Array	Yes	The list of delimiters.
caseSensitive	Boolean	No	Specifies whether the fields are case-sensitive.
chn	Boolean	No	Specifies whether the fields contain Chinese characters.
include_keys	Array	No	The list of included fields. This parameter and the exclude_keys parameter cannot be specified at the same time.
exclude_keys	Array	No	The list of the excluded fields. This parameter and the include_keys parameter cannot be specified at the same time.

The following table describes the key-value index attributes.

Attribute	Type	Required	Description
type	String	Yes	The type of the fields.
alias	String	No	The alias of the field.
chn	Boolean	No	Specifies whether the fields contain Chinese characters. This parameter takes effect only when the type parameter is set to text.
token	Array	Required only when the type parameter is set to text	The list of delimiters. This parameter takes effect only when the type parameter is set to text.
caseSensitive	Boolean	No	Specifies whether the fields are case-sensitive. This parameter takes effect only when the type parameter is set to text.
doc_value	Boolean	No	Specifies whether to enable analytics for the field.

Response elements

- Response headers

The UpdateIndex API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	ParameterInvalid	log store index is not created
400	ParameterInvalid	key config must has type
400	IndexInfoInvalid	required field token is lacking or of error format

HTTP status code	Error code	Error message
400	IndexInfoInvalid	required fields line/keys are lacking or of error format
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests


```
PUT /logstores/logstore-4/index HTTP/1.1
Header:
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Mon, 07 May 2018 09:47:20 GMT
Content-Type: application/json
Content-MD5: 1860B805A6AA5288B97B32CF3B519112
Content-Length: 316
Connection: Keep-Alive

Body:
{
  "line": {
    "token": [
      " ",
      " ",
      " ",
      "\"",
      ":",
      "=",
      "(",
      ")",
      "[",
      "]",
      "{",
      "}",
      "?",
      "@",
      "$",
      "<",
      ">",
      "/",
      ":",
      "\n",
      "\t",
      "\r"
    ]
  },
  "keys": {
    "agent": {
      "doc_value": true,
      "caseSensitive": true,
      "alias": "agent_alias",
      "type": "text",
      "token": [
        " ",
        " ",
        " ",
        "\"",
        ":",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "$",
        "<",
        ">",
        "/",
        ":",
        "\n",
        "\t",
        "\r"
      ]
    }
  },
  "ttl": 90
}
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Mon, 07 May 2018 09:47:21 GMT
x-log-requestid: 5AF020A98CBAA2EF52AB66C9
```

4.3.7.18. CreateIndex

Creates indexes for a specified Logstore.

Request syntax

```
POST /logstores/<logstoreName>/index HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: <x-log-bodyrawsize>
User-Agent: <User-Agent>
x-log-apiversion: <APIVersion>
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <Content-Length>
{
  "line": <full text index>,
  "keys": <key-value index>
}
```

Request elements

- Request headers

The CreateIndex API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

-

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
keys	Object	No	The key-value index, where the key refers to the field name and the value refers to the field value.
line	Object	No	The full-text index.

You must specify the keys parameter or the line parameter, or both. The following table describes the full-text index attributes.

Attribute	Type	Required	Description
caseSensitive	Boolean	No	Specifies whether the fields are case-sensitive.
chn	Boolean	No	Specifies whether the fields contain Chinese characters.
token	Array	Yes	The list of delimiters.
include_keys	Array	No	The list of included fields. This parameter and the exclude_keys parameter cannot be specified at the same time.
exclude_keys	Array	No	The list of excluded fields. This parameter and the include_keys parameter cannot be specified at the same time.

The following table describes the key-value index attributes.

Attribute	Type	Required	Description
type	String	Yes	The type of the field.
alias	String	No	The alias of the field.
chn	Boolean	No	Specifies whether the fields contain Chinese characters. This parameter takes effect only when the type parameter is set to text.
token	Array	Required only when the type parameter is set to text	The list of delimiters. This parameter takes effect only when the type parameter is set to text.
caseSensitive	Boolean	No	Specifies whether the fields are case-sensitive. This parameter takes effect only when the type parameter is set to text.

Attribute	Type	Required	Description
doc_value	Boolean	No	Specifies whether to enable analytics for the field.

Response elements

- Response headers

The CreateIndex API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	IndexInfoInvalid	required field token is lacking or of error format
400	IndexAlreadyExist	log store index is already created
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

 **Note** The {Project} and {logstoreName} variables in the preceding error messages are replaced with a specific project name and Logstore name.

Examples

Sample requests

```
POST /logstores/my-logstore/index HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Mon, 07 May 2018 09:43:16 GMT
Content-Type: application/json
Content-MD5: 22876515FC311F857AD6C7902F6A0148
Content-Length: 316
Connection: Keep-Alive

{
  "line": {
    "token": [
      " ",
      " ",
      " ",
      " ",
      "\"",
      ":",
      "=",
      "(",
      ")",
      "[",
      "]",
      "{",
      "}",
      "?",
      "@",
      "$",
      "<",
      ">",
      "/",
      ":",
      "\n",
      "\t",
      "\r"
    ]
  },
  "keys": {
    "agent": {
      "doc_value": true,
      "caseSensitive": true,
      "alias": "agent_alias",
      "type": "text",
      "token": [
        " ",
        " ",
        " ",
        " ",
        "\"",
        ":",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "$",
        "<",
        ">",
        "/",
        ":",
        "\n",
        "\t",
        "\r"
      ]
    }
  }
}
```

4.3.7.19. DeleteIndex

Deletes the indexes of a Logstore.

Request syntax

```
DELETE /logstores/<logstoreName>/index HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

Request parameters

- Request headers

The DeleteIndex API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.

Response elements

- Response headers

The DeleteIndex API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	ParameterInvalid	log store index not created
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
DELETE /logstores/my-logstore/index HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 06 May 2018 13:22:57 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 06 May 2018 13:22:57 GMT
x-log-requestid: 5AEF01B1A458E41C2F8326A8
```

4.3.7.20. GetIndex

Queries the indexes of a Logstore.

Request syntax

The request syntax of the GetIndex API operation is as follows:

```
GET /logstores/<logstoreName>/index HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

Request elements

- Request headers

The GetIndex API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common response headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.

Response elements

- Response headers

The GetIndex API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

When the GetIndex API operation succeeds, the response body contains the indexes of the specified project and Logstore. The following table describes the response body parameters.

Parameter	Type	Description
index_mode	String	The type of the indexes.
keys	Dictionary	The key-value index, where the key refers to the field name and the value refers to the field value.
line	object	The full-text index.
storage	String	The storage type. The default value is pg, which refers to PostgreSQL.
ttr	Integer	The time to live (TTL) of the index file. Valid values: 7, 30, and 90. Unit: days.
lastModifyTime	Integer	The time when the indexes were last modified. The value is a Unix timestamp.

- Full-text index attributes

Parameter	Type	Description
caseSensitive	Boolean	Indicates whether the indexes are case sensitive.
chn	Boolean	Indicates whether the fields contain Chinese characters.
token	Array	The list of delimiters.
include_keys	Array	The list of included fields.
exclude_keys	Array	The list of excluded fields.

- Key-value index attributes

Parameter	Type	Description
type	String	Indicates the type of the field.
alias	String	The alias of the field.
chn	Boolean	Indicates whether the fields contain Chinese characters. This parameter takes effect only when the type parameter is set to text.
token	Array	The list of delimiters. This parameter takes effect only when the type parameter is set to text.
caseSensitive	Boolean	Indicates whether the fields are case-sensitive. This parameter takes effect only when the type parameter is set to text.
doc_value	Boolean	Indicates whether to enable analytics for the index.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message	Description
400	IndexConfigNotExist	index config doesn't exist	The error message returned because the specified index does not exist.
404	ProjectNotExist	The Project does not exist : {Project}	The error message returned because the specified project does not exist.
404	LogStoreNotExist	logstore {logstoreName} does not exist	The error message returned because the specified Logstore does not exist.
500	InternalServerError	Specified Server Error Message	The error message returned because an error occurs on the specified server.

Examples

- Sample requests

```
GET /logstores/logstore-4/index HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 06 May 2018 13:08:42 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Type: application/json
Content-Length: 712
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 06 May 2018 13:08:42 GMT
x-log-requestid: 5AEEFE5A8B8AEB5E6C82B395
{
  "index_mode": "v2",
  "keys": {
    "agent": {
      "alias": "",
      "caseSensitive": false,
      "chn": false,
      "doc_value": true,
      "token": [
        ",",
        " ",
        "!",
        "\"",
        "'",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        ";",
        "@",
        "&"
      ]
    }
  }
}
```

```

        "<",
        ">",
        "/",
        ":",
        "\\n",
        "\\t",
        "\\r"
    ],
    "type": "text"
},
"bytes": {
    "alias": "",
    "doc_value": true,
    "type": "long"
},
"remote_ip": {
    "alias": "",
    "caseSensitive": false,
    "chn": false,
    "doc_value": true,
    "token": [
        ",",
        " ",
        "!",
        "\"",
        ":",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "$",
        "<",
        ">",
        "/",
        ":",
        "\\n",
        "\\t",
        "\\r"
    ],
    "type": "text"
},
"response": {
    "alias": "",
    "doc_value": true,
    "type": "long"
}
},
"line": {
    "caseSensitive": false,
    "chn": false,
    "token": [
        ",",
        " ",
        "!",
        "\"",
        ":",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "$",
        "<",
        ">",
        "/",
        ":",
        "\\n",
        "\\t",
        "\\r"
    ]
},
"storage": "pg",
"ttl": 30,
"lastModifyTime": 1524155379
}

```


4.3.7.21. PutWebtracking

Merges multiple logs into one log during a single collection request. In the WebTracking mode, only one log can be written to each request. This API operation applies to scenarios where you need to collect a large amount of logs. Before you can call the PutWebtracking API operation, you must turn on the WebTracking switch for the corresponding Logstore. We recommend that you call this API operation to collect logs from web pages or clients.

Request syntax

```
POST {project}.{endpoint}/logstores/{logstoreName}/track HTTP/1.1
x-log-apiversion: 0.6.0
x-log-bodyrawsize: 1234
x-log-compresstype: gzip
{
  "__topic__": "topic",
  "__source__": "source",
  "__logs__": [
    {
      "key1": "value1",
      "key2": "value2"
    },
    {
      "key1": "value1",
      "key2": "value2"
    }
  ],
  "__tags__": {
    "tag1": "value1",
    "tag2": "value2"
  }
}
```

Request elements

- Request headers

Only the following three fields are required in the request header:

- o x-log-apiversion: 0.6.0
- o x-log-bodyrawsize: 1234
- o x-log-compresstype: gzip

The first two fields are required, and the third one is optional. For more information, see [Common request headers](#).

- Request parameters

- o URL parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
endpoint	String	Yes	The endpoint of Log Service. For more information, see Obtain an endpoint .
project	String	Yes	The name of the project.

- o Request body parameters

Parameter	Type	Required	Description
__topic__	String	No	The topic of the log.
__source__	String	No	The source of the log.
__logs__	list	Yes	The content of the log. Each element is a JSON object that indicates a log.  Note The time in a log collected by using the WebTracking feature is the time when the log arrives at Log Service. You do not need to specify the __time__ field for each log. If this field is specified, it is overwritten by the time when the log arrives at Log Service.
__tags__	object	No	The tag of the log.

Response elements

- Response headers

The PutWebtracking API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

No response parameter is returned after the request succeeds.

Examples

- Sample requests

```
POST my-project.cn-hangzhou.log.aliyuncs.com/logstores/logstore-test/track HTTP/1.1
x-log-apiversion: 0.6.0
x-log-bodyrawsize: 1234
x-log-compress-type: gzip
{
  "topic": "topic",
  "source": "source",
  "logs": [
    {
      "foo": "bar",
      "foo1": "bar1"
    },
    {
      "bar": "foo",
      "bar1": "foo1"
    }
  ],
  "tags": {
    "tag1": "value1",
    "tag2": "value2"
  }
}
```

- Sample responses

```
HTTP/1.1 200 OK
Header:
{
  Date=Wed, 20 May 2019 07:35:00 GMT,
  Content-Length=0,
  x-log-requestid=5642EFA499248C827B012B39,
  Connection=close,
  Server=nginx/1.6.1
}
```

4.3.8. Logtail machine group operations

4.3.8.1. CreateMachineGroup

Creates a machine group to which the Logtail configuration is distributed and applied.

Request syntax

```
POST /machinegroups HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/json
Content-Length: <Content Length>
Content-MD5: <Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
  "groupName": "testgroup",
  "groupType": "",
  "groupAttribute": {
    "externalName": "testgroup",
    "groupTopic": "testgrouptopic"
  },
  "machineIdentifyType": "ip",
  "machineList": [
    "test-ip1",
    "test-ip2"
  ]
}
```

Request elements

- Request headers

The CreateMachineGroup API does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Request body parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group. The name must be unique in a project.
groupType	String	No	The type of the machine group. Default value: null.
machineIdentifyType	String	Yes	The type of the machine identifier. Valid values: ip and userdefined-id.
groupAttribute	object	Yes	The attribute of the machine group. Default value: null.
machineList	Array	Yes	The specific machine identifier, which can be an IP address or user-defined identifier.

The following table describes the attributes that are specified by the groupAttribute parameter.

Attribute	Type	Required	Description
groupTopic	String	No	The topic of the machine group. Default value: null.
externalName	String	No	The external identifier on which the machine group depends. Default value: null.

Response elements

- Response headers

The CreateMachineGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	MachineGroupAlreadyExist	group {GroupName} already exists
400	InvalidParameter	invalid group resource json
500	InternalServerError	Internal server error

Examples

- Sample requests

```
POST /machinegroups HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 17:57:33 GMT",
  "Content-Length": "187",
  "x-log-signaturemethod": "hmac-sha1",
  "Content-MD5": "82033D507DEAD72067BB58DFDCB590D",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/json",
  "x-log-bodyrawsize": "0"
}
Body :
{
  "groupName": "test-machine-group",
  "groupType": "",
  "machineIdentifyType": "ip",
  "groupAttribute": {
    "groupTopic": "testtopic",
    "externalName": "testgroup"
  },
  "machineList": [
    "127.0.0.1",
    "127.0.0.2"
  ]
}
```

- Sample responses

```

HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 17:57:33 GMT",
  "Content-Length": "0",
  "x-log-requestid": "5642300D99248CB76D005D36",
  "Connection": "close",
  "Server": "nginx/1.6.1"
}

```

4.3.8.2. DeleteMachineGroup

Deletes a machine group. After you delete a machine group, the Logtail configuration applied to the machine group is also deleted.

Request syntax

```

DELETE /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

```

Request elements

- Request headers

The DeleteMachineGroup operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group.

Response elements

- Response headers

The DeleteMachineGroup operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
500	InternalServerError	internal server error

Examples

- Sample requests

```

DELETE /machinegroups/test-machine-group-4 HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 19:13:28 GMT",
  "Content-Length": "0",
  "x-log-signaturemethod": "hmac-sha1",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/x-protobuf",
  "x-log-bodyrawsize": "0"
}

```

- Sample responses

```

HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 19:13:28 GMT",
  "Content-Length": "0",
  "x-log-requestid": "564241D899248C827B000CFE",
  "Connection": "close",
  "Server": "nginx/1.6.1"
}

```

4.3.8.3. UpdateMachineGroup

Updates the information of a machine group. If the Logtail configurations are applied to a machine group, the configurations are automatically applied to or removed from a machine when the machine is added to or removed from the machine group.

Request syntax

```
PUT /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/json
Content-Length: <Content Length>
Content-MD5: <Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
  "groupName": "test-machine-group",
  "groupType": "",
  "groupAttribute": {
    "externalName": "testgroup",
    "groupTopic": "testgrouptopic"
  },
  "machineIdentifyType": "ip",
  "machineList": [
    "test-ip1",
    "test-ip2"
  ]
}
```

Request elements

- Request headers

The UpdateMachineGroup API does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

- URL parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group.

- o

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group. The name must be unique in a project.
groupType	String	No	The type of the machine group. Default value: null.
machineIdentifyType	String	Yes	The type of the machine identifier. Valid values: ip and userdefined-id.
groupAttribute	Object	Yes	The attribute of the machine group. Default value: null.
machineList	Array	Yes	The list of machine identifiers, which can be IP addresses or user-defined identifiers.

The groupAttribute parameter consists of the following parameters.

Parameter	Type	Required	Description
groupTopic	String	No	The topic of the machine group. Default value: null.
externalName	String	No	The identifier of the external management system on which the machine group depends. Default value: null.

Response elements

- Response headers

The UpdateMachineGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
400	InvalidParameter	invalid group resource json
500	InternalServerError	internal server error

Examples

• Sample requests

```
PUT /machinegroups/test-machine-group HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 18:41:43 GMT",
  "Content-Length": "194",
  "x-log-signaturemethod": "hmac-sha1",
  "Content-MD5": "2CEBAEBE53C078891527CB70A855BAF4",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/json",
  "x-log-bodyrawsize": "0"
}
Body :
{
  "groupName": "test-machine-group",
  "groupType": "",
  "machineIdentifyType": "userdefined",
  "groupAttribute": {
    "groupTopic": "testtopic2",
    "externalName": "testgroup2"
  },
  "machineList": [
    "uu_id_1",
    "uu_id_2"
  ]
}
```

• Sample responses

```
HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 18:41:43 GMT",
  "Content-Length": "0",
  "x-log-requestid": "56423A6799248CA57B00035C",
  "Connection": "close",
  "Server": "nginx/1.6.1"
}
```

4.3.8.4. ListMachineGroup

Lists machine groups.

Request syntax

```
GET /machinegroups? offset=1&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

• Request headers

The `ListMachineGroup` API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
offset	Integer	No	The start position of the records to return. Default value: 0.
size	Integer	No	The maximum number of entries to return on each page. Default and maximum value: 500.

Parameter	Type	Required	Description
groupName	String	No	The group machine name that is used as a filtering condition. Fuzzy match is supported.

Response elements

- Response headers

The ListMachineGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the request succeeds, the response body contains a list of all machine groups in the specified project. The following table describes the response parameters.

Parameter	Type	Description
count	Integer	The number of returned machine groups on each page.
total	Integer	The total number of returned machine groups.
machinegroups	JSON array	The list of returned machine groups.

```
{
  "machinegroups": [
    "test-machine-group",
    "test-machine-group-2"
  ],
  "count": 2,
  "total": 2
}
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
500	InternalServerError	internal server error

Examples

- Sample requests

```
GET /machinegroups? groupName=test-machine-group&offset=0&size=3 HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 18:34:44 GMT",
  "Content-Length": "0",
  "x-log-signaturemethod": "hmac-sha1",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/x-protobuf",
  "x-log-bodyrawsize": "0"
}
```

- Sample responses

```
HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 18:34:44 GMT",
  "Content-Length": "83",
  "x-log-requestid": "564238C499248C8F7B0001DE",
  "Connection": "close",
  "Content-Type": "application/json",
  "Server": "nginx/1.6.1"
}
Body :
{
  "machinegroups": [
    "test-machine-group",
    "test-machine-group-2"
  ],
  "count": 2,
  "total": 2
}
```

4.3.8.5. GetMachineGroup

Queries the details of a machine group.

Request syntax

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The GetMachineGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group.

Response elements

- Response headers

The GetMachineGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

Parameter	Type	Description
groupName	String	The name of the machine group. The name must be unique in a project.
groupType	String	The type of the machine group. Valid values: null and Armory. Default value: null.
machineIdentifyType	String	The type of the machine identifier. Valid values: ip and userdefined-id.
groupAttribute	JSON object	The attribute of the machine group. Default value: null.
machineList	JSON array	The list of machine identifiers, which can be IP addresses or user-defined identifiers.
createTime	Integer	The time when the machine group was created.
lastModifyTime	Integer	The time when the machine group was last updated.

The groupAttribute parameter consists of the following parameters.

Name	Type	Required	Description
groupTopic	String	No	The topic of the machine group.
externalName	String	No	The identifier of the external management system (Armory) on which the machine group depends.

```
{
  "groupName": "test-machine-group",
  "groupType": "",
  "groupAttribute": {
    "externalName": "testgroup",
    "groupTopic": "testtopic"
  },
  "machineIdentifyType": "ip",
  "machineList": [
    "127.0.0.1",
    "127.0.0.2"
  ],
  "createTime": 1447178253,
  "lastModifyTime": 1447178253
}
```

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	MachineGroupNotExist	group {GroupName} does not exist
500	InternalServerError	internal server error

Examples

• Sample requests

```
GET /machinegroups/test-machine-group HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 18:15:24 GMT",
  "Content-Length": "0",
  "x-log-signaturemethod": "hmac-sha1",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/x-protobuf",
  "x-log-bodyrawsize": "0"
}
```

• Sample responses

```
HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 18:15:23 GMT",
  "Content-Length": "239",
  "x-log-requestid": "5642343B99248CB36D0060B8",
  "Connection": "close",
  "Content-Type": "application/json",
  "Server": "nginx/1.6.1"
}
Body :
{
  "groupName": "test-machine-group",
  "groupType": "",
  "groupAttribute": {
    "externalName": "testgroup",
    "groupTopic": "testtopic"
  },
  "machineIdentifyType": "ip",
  "machineList": [
    "127.0.0.1",
    "127.0.0.2"
  ],
  "createTime": 1447178253,
  "lastModifyTime": 1447178253
}
```

4.3.8.6. ApplyConfigToMachineGroup

Applies a Logtail configuration to a machine group.

Request syntax

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

• Request headers

The ApplyConfigToMachineGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
GroupName	String	Yes	The name of the machine group.
ConfigName	String	Yes	The name of the Logtail configuration.

Response elements

• Response headers

The ApplyConfigToMachineGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

• Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
404	ConfigNotExist	config {ConfigName} does not exist
500	InternalServerError	internal server error

Examples

• Sample requests

```
PUT /machinegroups/sample-group/configs/logtail-config-sample
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Mon, 09 Nov 2015 09:44:43 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

• Sample responses

```
{
  "date": "Mon, 09 Nov 2015 09:44:43 GMT",
  "connection": "close",
  "x-log-requestid": "56406B0B99248CAA230BA094",
  "content-length": "0",
  "server": "nginx/1.6.1"
}
```

4.3.8.7. RemoveConfigFromMachineGroup

Deletes a Logtail configuration from a machine group.

Request syntax

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

• Request headers

The RemoveConfigFromMachineGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
GroupName	String	Yes	The name of the machine group.
ConfigName	String	Yes	The name of the Logtail configuration.

Response elements

• Response headers

The RemoveConfigFromMachineGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

• Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
404	ConfigNotExist	config {ConfigName} does not exist
500	InternalServerError	internal server error

Examples

• Sample requests

```
DELETE /machinegroups/sample-group/configs/logtail-config-sample
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Mon, 09 Nov 2015 09:48:48 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

• Sample responses

```
{
  "date": "Mon, 09 Nov 2015 09:48:48 GMT",
  "connection": "close",
  "x-log-requestid": "56406C0099248CAA230BE135",
  "content-length": "0",
  "server": "nginx/1.6.1"
}
```

4.3.8.8. ListMachines

Lists your machines in a machine group that is connected with Log Service.

Request syntax

```
GET /machinegroups/{groupName}/machines HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

• Request headers

The ListMachines API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group.
offset	Integer	No	The start position of the returned records. Default value: 0.
size	Integer	No	The maximum number of entries to return on each page. Default and maximum value: 500.

Response elements

• Response headers

The ListMachines API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

• Response parameters

Parameter	Type	Description
count	Integer	The number of returned machines on each page.
total	Integer	The total number of machines.
machines	JSON array	The list of returned machines.

The machines parameter consists of the following attributes.

Attribute	Type	Description
ip	String	The IP address of the machine.
machine-uniqueid	String	The universally unique identifier (UUID) of the machine.
userdefined-id	String	The user-defined identifier of the machine.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
500	InternalServerError	internal server error

Additional considerations

The API operation lists only the machines that are normally connected with Log Service.

Examples

• Sample requests

```
GET /machinegroups/test-machine-group-5/machines? offset=0&size=3 HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 19:04:57 GMT",
  "Content-Length": "0",
  "x-log-signaturemethod": "hmac-sha1",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/x-protobuf",
  "x-log-bodyrawsize": "0"
}
```

• Sample responses

```
HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 19:04:58 GMT",
  "Content-Length": "324",
  "x-log-requestid": "56423FD999248C827B000A57",
  "Connection": "close",
  "Content-Type": "application/json",
  "Server": "nginx/1.6.1"
}
Body :
{
  "machines": [
    {
      "ip": "10.101.166.116",
      "machine-uniqueid": "",
      "userdefined-id": "",
      "lastHeartbeatTime": 1447182247
    },
    {
      "ip": "10.101.165.193",
      "machine-uniqueid": "",
      "userdefined-id": "",
      "lastHeartbeatTime": 1447182246
    },
    {
      "ip": "10.101.166.91",
      "machine-uniqueid": "",
      "userdefined-id": "",
      "lastHeartbeatTime": 1447182248
    }
  ],
  "count": 3,
  "total": 8
}
```

4.3.8.9. GetAppliedConfigs

Queries the Logtail configurations that are applied to a machine group.

Request syntax

```
GET /machinegroups/{groupName}/configs HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

• Request headers

The GetAppliedConfigs API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group.

Response elements

- Response headers

The GetAppliedConfigs API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After a GetAppliedConfigs request succeeds, the response body contains a list of all machines in the specified machine group. The following table describes the response parameters.

Parameter	Type	Description
count	Integer	The number of returned configurations on each page.
configs	String array	The list of returned configurations.

Error codes

In addition to the [Common error codes](#) of Log Service API operations, the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group (GroupName) does not exist
500	InternalServerError	internal server error

Examples

- Sample requests

```
GET /machinegroups/test-machine-group/configs HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Tue, 10 Nov 2015 19:45:48 GMT",
  "Content-Length": "0",
  "x-log-signaturemethod": "hmac-sha1",
  "User-Agent": "sls-java-sdk-v-0.6.0",
  "Content-Type": "application/x-protobuf",
  "x-log-bodyrawsize": "0"
}
```

- Sample responses

```
HTTP/1.1 200 OK
Header :
{
  "Date": "Tue, 10 Nov 2015 19:45:48 GMT",
  "Content-Length": "53",
  "x-log-requestid": "5642496C99248C8C7B00173F",
  "Connection": "close",
  "Content-Type": "application/json",
  "Server": "nginx/1.6.1"
}
Body :
{
  "configs": [
    "two",
    "three",
    "test_logstore"
  ],
  "count": 3
}
```

4.3.9. Logtail configuration operations

4.3.9.1. CreateConfig

Creates a Logtail configuration.

Request syntax

```

POST /configs HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
  "configName": "testcategory1",
  "inputType": "file",
  "inputDetail": {
    "logType": "common_reg_log",
    "logPath": "/var/log/httpd/",
    "filePattern": "access*.log",
    "localStorage": true,
    "timeFormat": "%Y/%m/%d %H:%M:%S",
    "logBeginRegex": ".*",
    "regex": "(\\w+) (\\s+)",
    "key": ["key1", "key2"],
    "filterKey": ["key1"],
    "filterRegex": ["regex1"],
    "fileEncoding": "utf8",
    "topicFormat": "none"
  },
  "outputType": "LogService",
  "outputDetail": {
    "logstoreName": "perfcounter"
  }
}

```

Request elements

- Request headers

The CreateConfig API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

For information about the request parameters and Logtail configuration examples, see [Logtail configurations](#).

Response elements

- Response headers

The CreateConfig API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response elements

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	ConfigAlreadyExist	config {Configname} already exists
400	InvalidParameter	invalid config resource json
500	InternalServerError	internal server error

The configuration fails to be created if an error occurs, for example, the configuration already exists, the format is incorrect, the required parameters are missing, or the quota is exceeded.

Examples

- Sample requests

```

POST /configs HTTP/1.1
Header :
{
  'Content-Length': 737,
  'Host': 'ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com',
  'x-log-bodyrawsize': 737,
  'Content-MD5': 'FBA01ECF7255BE143379BC70C56BBF68',
  'x-log-signaturemethod': 'hmac-sha1',
  'Date': 'Mon, 09 Nov 2015 07:45:30 GMT',
  'x-log-apiversion': '0.6.0',
  'User-Agent': 'log-python-sdk-v-0.6.0',
  'Content-Type': 'application/json',
  'Authorization': 'LOG <yourAccessKeyId>:<yourSignature>'
}
Body:
{
  "configName": "sample-logtail-config",
  "inputType": "file",
  "inputDetail": {
    "logType": "common_reg_log",
    "logPath": "/var/log/httpd/",
    "filePattern": "access*.log",
    "localStorage": true,
    "timeFormat": "%d/%b/%Y:%H:%M:%S",
    "logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
    "regex": "([\\d\\.]) + \\S+ \\S+ \\[(\\S+) \\S+\\] \"([\\w+ ]+\" ([^\" ]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\" \"([^\"]*)\".*",
    "key": ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "browser"],
    "filterKey": [],
    "filterRegex": [],
    "topicFormat": "none",
    "fileEncoding": "utf8"
  },
  "outputType": "LogService",
  "outputDetail": {
    {
      "logstoreName": "sls-test-logstore"
    }
  }
}

```

- Sample responses

```

HTTP/1.1 200 OK
Header
{
  'date': 'Mon, 09 Nov 2015 07:45:30 GMT',
  'connection': 'close',
  'x-log-requestid': '56404f1a99248ca26c002180',
  'content-length': '0',
  'server': 'nginx/1.6.1'
}

```

4.3.9.2. ListConfig

Lists all configurations of a project. You can set the relevant parameters in the syntax to list the configurations on multiple pages.

Request syntax

```

GET /configs? offset=0&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

```

Request elements

- Request headers

The List Config API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
offset	Integer	No	The start position of entries to return. Default value: 0.
size	Integer	No	The maximum number of entries to return on each page. Default value: 500.

Response elements

- Response headers

The List Config API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The response body of the ListConfig API operation contains a list of all configurations in the project. The following table describes the response parameters.

Parameter	Type	Description
count	Integer	The number of returned configurations on each page.
total	Integer	The total number of returned configurations.
configs	String array	The list of returned configurations.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	config {Configname} does not exist
500	InternalServerError	internal server error

Examples

- Sample requests

```
GET /configs? offset=0&size=10 HTTP/1.1
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Mon, 09 Nov 2015 09:19:13 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

- Sample responses

```
Header :
{
  "content-length": "103",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Mon, 09 Nov 2015 09:19:13 GMT",
  "content-type": "application/json",
  "x-log-requestid": "5640651199248CAA2300C2BA"
}
Body:
{
  "count": 3,
  "configs":
  [
    "logtail-config-sample",
    "logtail-config-sample-2",
    "logtail-config-sample-3"
  ],
  "total": 3
}
```

4.3.9.3. GetAppliedMachineGroups

Lists the machines that apply a Logtail configuration.

Request syntax

```
GET /configs/{configName}/machinegroups HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The GetAppliedMachineGroups API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Logtail configurations](#).

- Request parameters

Parameter	Type	Required	Description
ConfigName	String	Yes	The name of the Logtail configuration.

Response elements

- Response headers

The GetAppliedMachineGroups API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the request succeeds, the response body contains a list of all machines in the specified machine group. The following table describes the response parameters.

Parameter	Type	Description
count	Integer	The number of returned machine groups.
machinegroups	String array	The list of returned machine groups.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group [GroupName] does not exist
500	InternalServerError	internal server error

Examples

- Sample requests

```
GET /configs/logtail-config-sample/machinegroups
Header:
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Mon, 09 Nov 2015 09:51:38 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

- Sample responses

```
Header :
{
  "content-length": "44",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Mon, 09 Nov 2015 09:51:38 GMT",
  "content-type": "application/json",
  "x-log-requestid": "56406CAA99248CAA230BE828"
}
Body:
{
  "count": 1,
  "machinegroups":
  [
    "sample-group"
  ]
}
```

4.3.9.4. GetConfig

Queries the details of a Logtail configuration.

Request syntax

```
GET /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The GetConfig API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
ConfigName	String	Yes	The name of the Logtail configuration.

Response elements

- Response headers

The GetConfig API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common request headers](#).

- Response parameters

For information about the response parameters and Logtail configuration examples, see [Logtail configurations](#).

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	Config {Configname} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
GET /configs/logtail-config-sample
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Mon, 09 Nov 2015 08:29:15 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

- Sample responses

```

Header :
{
  "content-length": "730",
  "server": "nginx/1.6.1",
  "connection": "close",
  "date": "Mon, 09 Nov 2015 08:29:15 GMT",
  "content-type": "application/json",
  "x-log-requestid": "5640595B99248CAA23004A59"
}

Body :
{
  "configName": "logtail-config-sample",
  "outputDetail": {
    "endpoint": "http://cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
    "logstoreName": "sls-test-logstore"
  },
  "outputType": "LogService",
  "inputType": "file",
  "inputDetail": {
    "regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \\\"(\\w+) ([^\"\\s]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \\\"([^\"]*)\\\" \\\"([^\"]*)\\\"\\.\\*\\.\\*",
    "filterKey": [],
    "logPath": "/var/log/httpd/",
    "logBeginRegex": "\\d+\\.\\.\\d+\\.\\.\\d+\\.\\.\\d+ - .*",
    "logType": "common_reg_log",
    "topicFormat": "none",
    "localStorage": true,
    "key": [
      "ip",
      "time",
      "method",
      "url",
      "request_time",
      "request_length",
      "status",
      "length",
      "ref_url",
      "browser"
    ],
    "filePattern": "access*.log",
    "timeFormat": "%d/%b/%Y:%H:%M:%S",
    "filterRegex": []
  },
  "createTime": 1447040456,
  "lastModifyTime": 1447050456
}

```

4.3.9.5. DeleteConfig

Deletes a Logtail configuration. When you call this API operation, the Logtail configuration is deleted even if it is applied to the Machine Group.

Request syntax

```
DELETE /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request elements

- Request headers

The DeleteConfig API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
ConfigName	String	Yes	The name of the logtail configuration.

Response elements

- Response headers

The DeleteConfig API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	config {Configname} does not exist
400	InvalidParameter	invalid config resource json
500	InternalServerError	internal server error

Examples

• Sample requests

```
DELETE /configs/logtail-config-sample
Header :
{
  "Content-Length": 0,
  "x-log-signaturemethod": "hmac-sha1",
  "x-log-bodyrawsize": 0,
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "Date": "Mon, 09 Nov 2015 09:28:21 GMT",
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

• Sample responses

```
Header :
{
  "date": "Mon, 09 Nov 2015 09:28:21 GMT",
  "connection": "close",
  "x-log-requestid": "5640673599248CAA230836C6",
  "content-length": "0",
  "server": "nginx/1.6.1"
}
```

4.3.9.6. UpdateConfig

Updates Logtail configurations. If the Logtail configurations are applied to a machine group, the configurations of the machine group are also updated.

Request syntax

```
PUT /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
  "configName": "testcategory1",
  "inputType": "file",
  "inputDetail": {
    "logType": "common_reg_log",
    "logPath": "/var/log/httpd/",
    "filePattern": "access.log",
    "localStorage": true,
    "timeFormat": "%Y/%m/%d %H:%M:%S",
    "logBeginRegex": ".*",
    "regex": "(\\w+) (\\s+)",
    "key": ["key1", "key2"],
    "filterKey": ["key1"],
    "filterRegex": ["regex1"],
    "topicFormat": "none"
  },
  "outputType": "LogService",
  "outputDetail": {
    "logstoreName": "perfcounter"
  }
}
```

Request elements

• Request headers

The UpdateConfig API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

For information about the request parameters and Logtail configuration examples, see [Logtail configurations](#).

Response elements

- Response headers

The UpdateConfig API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the request succeeds, the HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	config {Configname} does not exist
400	InvalidParameter	invalid config resource json
400	BadRequest	config resource configname does not match with request
500	InternalServerError	internal server error

The configuration fails to be created if an error occurs, for example, the format is incorrect, the required parameters are missing, or the quota is exceeded.

Examples

- Sample requests

```
PUT /configs/logtail-config-sample
Header :
{
  "Content-Length": 737,
  "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
  "x-log-bodyrawsize": 737,
  "Content-MD5": "431263EB105D584A5555762A81E869C0",
  "x-log-signaturemethod": "hmac-sha1",
  "Date": "Mon, 09 Nov 2015 09:14:32 GMT",
  "x-log-apiversion": "0.6.0",
  "User-Agent": "log-python-sdk-v-0.6.0",
  "Content-Type": "application/json",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
Body :
{
  "outputDetail": {
    "logstoreName": "sls-test-logstore"
  },
  "inputType": "file",
  "inputDetail": {
    "regex": "([\\d\\.]) + \\S+ \\S+ \\[([\\S+]) \\S+\\] \"([\\w+])\" ([^\"\\s]*)\" ([\\d\\.])+ ([\\d+]) ([\\d+]) ([\\d+|-]) \"([^\"]*)\" \"([^\"]*)\"\\.\"",
    "filterKey": [],
    "logPath": "/var/log/nginx/",
    "logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
    "logType": "common_reg_log",
    "topicFormat": "none",
    "localStorage": true,
    "key": [
      "ip",
      "time",
      "method",
      "url",
      "request_time",
      "request_length",
      "status",
      "length",
      "ref_url",
      "browser"
    ],
    "filePattern": "access*.log",
    "timeFormat": "%d/%b/%Y:%H:%M:%S",
    "filterRegex": []
  },
  "outputType": "LogService",
  "configName": "logtail-config-sample"
}
```

- Sample responses

```
{
  "date": "Mon, 09 Nov 2015 09:14:32 GMT",
  "connection": "close",
  "x-log-requestid": "564063F899248CAA2300B778",
  "content-length": "0",
  "server": "nginx/1.6.1"
}
```

4.3.10. Consumer group operations

4.3.10.1. CreateConsumerGroup

Creates a consumer group in a Logstore.

Request syntax

```
POST /logstores/<logstoreName>/consumergroups HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
Content-Length: <ContentLength>
{
  "consumerGroup": <consumerGroup>,
  "timeout": <timeout>,
  "order": <order>
}
```

Request elements

- Request headers

The CreateConsumerGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.
consumerGroup	String	Yes	The name of the consumer group. The name must be unique in a project.
timeout	Integer	Yes	The timeout period. If Log Service does not receive a heartbeat message from a consumer within the timeout period, the consumer is deleted.
order	Boolean	Yes	Specifies whether to consume data in sequence from a single shard. Valid values: ◦ true: The data in a single shard is consumed in sequence. The data in the original shard is first consumed after the shard is split. Then, the data in the new shards is consumed at the same time. ◦ false: The data in a single shard is randomly consumed.

Response elements

- Response headers

The CreateConsumerGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	ConsumerGroupAlreadyExist	consumer group already exist
400	JsonInfoInvalid	consumerGroup or timeout is of error format
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
POST /logstores/my-logstore/consumergroups HTTP/1.1
Header:
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Fri, 04 May 2018 08:02:22 GMT
Content-Type: application/json
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
Content-Length: 65
Connection: Keep-Alive
Body:
{
  "consumerGroup": "consumer-group-1",
  "timeout": 300,
  "order": true
}
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Fri, 04 May 2018 08:15:11 GMT
x-log-requestid: 5AEC168FA796F4195BF404CB
```

4.3.10.2. DeleteConsumerGroup

Deletes a consumer group.

Request syntax

```
DELETE /logstores/<logstoreName>/consumergroups/<consumerGroupName> HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
```

Request elements

- Request headers

The DeleteConsumerGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore to which the consumer group belongs.
consumerGroup	String	Yes	The consumer group to be deleted.

Response elements

- Response headers

The DeleteConsumerGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Additional considerations

If you attempt to delete a consumer group that does not exist, the system will return a success message.

Examples

• Sample requests

```
DELETE /logstores/logstore-test/consumergroups/consumer-group-1 HTTP/1.1
Header:
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Fri, 04 May 2018 08:02:22 GMT
Content-Type: application/json
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
Content-Length: 65
Connection: Keep-Alive
```

• Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Fri, 04 May 2018 08:15:11 GMT
x-log-requestid: 5AEC168FA796F4195BF404CB
```

4.3.10.3. GetCheckPoint

Queries the checkpoints of one or all shards from which data is consumed by a consumer group.

Request syntax

```
GET /logstores/<logstoreName>/consumergroups/<consumerGroup>? shard=<shardId> HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Connection: Keep-Alive
```

Request elements

• Request headers

The GetCheckPoint API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore to which the consumer group belongs.
consumerGroup	String	Yes	The name of the consumer group.
shardId	Integer	No	The ID of the shard. - If the specified shard does not exist, an empty list is returned. - If the shard ID is not specified, the checkpoints of all shards are returned.

Response elements

• Response headers

The GetCheckPoint API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

• Response parameters

After the request succeeds, the response body contains the checkpoint of the shard from which data is consumed by the specified consumer group. The following table describes the response parameters.

Parameter	Type	Description
shard	Integer	The ID of the shard.
checkpoint	String	The value of the checkpoint.
updateTime	Integer	The time when the checkpoint was last updated.

Parameter	Type	Description
consumer	String	The consumer that consumes data from the shard.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
404	ConsumerGroupNotExist	consumer group not exist
500	InternalServerError	Specified Server Error Message

Examples

• Sample requests

```
GET /logstores/my-logstore/consumergroups/consumer_group_test? shard=0
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Fri, 04 May 2018 09:26:53 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

• Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Type: application/json
Content-Length: 111
Connection: close
Access-Control-Allow-Origin: *
Date: Fri, 04 May 2018 09:26:53 GMT
x-log-requestid: 5AEC275D1FFC036AB254B20C
[
  {
    "shard": 0,
    "checkpoint": "MTUyNDElNTM3OTM3MzkWODQ5Ng==",
    "updateTime": 1524224984800922,
    "consumer": "consumer_1"
  }
]
```

4.3.10.4. HeartBeat

Sends a heartbeat packet to Log Service for a consumer.

Request syntax

```
POST /logstores/<logstoreName>/consumergroups/<consumerGroup>? type=heartbeat&consumer=<consumer> HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
Content-Length: <ContentLength>
<shard ID list>
```

Request elements

• Request headers

The HeartBeat API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

• Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore. The name must be unique in a project.

Parameter	Type	Required	Description
consumerGroup	String	Yes	The name of the consumer group. The name must be unique in a project.
consumer	String	Yes	The consumer.
{Shard ID List}	Array	Yes	The ID list of shards from which data is consumed.

Response elements

- Response headers

The HeartBeat API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the request succeeds, the HTTP status code 200 is returned. An ID list is also returned, containing all shards from which data are consumed by the consumer.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	NotExistConsumerWithBody	non-exist consumer with non-empty body of heartbeat message
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
404	ConsumerGroupNotExist	consumer group not exist
500	InternalServerError	Specified Server Error Message

Additional considerations

The heartbeat packet can be sent only when the connection between the consumer and Log Service is established.

Examples

- Sample requests

```
POST /logstores/my-logstore/consumerGroups/consumer_group_test? type=heartbeat&consumer=consumer_1 HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 06 May 2018 09:44:10 GMT
Content-Type: application/json
Content-MD5: 8D5162CA104FA7E79FE80FD92BB657FB
Content-Length: 3
Connection: Keep-Alive
[0]
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Type: application/json
Content-Length: 5
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 06 May 2018 09:44:11 GMT
x-log-requestid: 5AEECE6B1FFC0366B2553FB5
[0,1]
```

4.3.10.5. ListConsumerGroup

Lists all consumer groups of a Logstore.

Request syntax

```
GET /logstores/<logstoreName>/consumerGroups HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

Request elements

- Request headers

The ListConsumerGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.

Response elements

- Response headers

The ListConsumerGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

After the ListConsumerGroup request succeeds, the response body contains a list of all consumer groups that read log data from the specified Logstore. The following table describes the response parameters.

Parameter	Type	Description
consumerGroup	String	The name of the consumer group.
timeout	Integer	The timeout period. If the specified Logstore does not receive a heartbeat packet from the consumer group during the timeout period, the consumer group is deleted.
order	Boolean	Indicates whether log data was consumed in sequence.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

- Sample requests

```
GET /logstores/my-logstore/consumerGroups HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Fri, 04 May 2018 08:47:30 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Type: application/json
Connection: close
Access-Control-Allow-Origin: *
Date: Fri, 04 May 2018 08:47:31 GMT
x-log-requestid: 5AEC1E23048191954B42EAB9
[
  {
    "name": "test-consumer-group",
    "timeout": 30,
    "order": true
  }
]
```

4.3.10.6. UpdateCheckPoint

Updates the checkpoint of a shard in a Logstore of a project for a consumer group.

Request syntax

```
POST /logstores/<logstoreName>/consumergroups/<consumerGroupName>? forceSuccess=<forceSuccess>&type=checkpoint&consumer=<consumer> HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "shard": <shardID>,
  "checkpoint": <checkpoint>
}
```

Request elements

- Request headers

The UpdateCheckpoint API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore to which the consumer group belongs.
consumerGroup	String	Yes	The name of the consumer group.
forceSuccess	Boolean	No	Specifies whether to forcibly update the consumer group.
consumer	String	No	The consumer.
shard	Integer	Yes	The ID of the shard.
checkpoint	String	Yes	The value of the checkpoint.

Response elements

- Response headers

The UpdateCheckpoint API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common error codes](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	InvalidShardCheckpoint	shard checkpoint not encoded by base64
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} does not exist
404	ConsumerGroupNotExist	consumer group not exist
404	ShardNotExist	shard not exist
500	InternalServerError	Specified Server Error Message

Additional considerations

If the consumer is not specified, the checkpoint can be updated only when the forceSuccess parameter is set to true.

Examples

- Sample requests

```
POST /logstores/logstore-4/consumergroups/consumer_group_test? forceSuccess=false&type=checkpoint&consumer=consumer_1 HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 06 May 2018 09:14:53 GMT
Content-Type: application/json
Content-MD5: 59457BAFB3F85A5117BDE243947583B0
Content-Length: 55
Connection: Keep-Alive
{
  "shard": 0,
  "checkpoint": "MTUyNDE1NTM3OTM3MzkwODQ5Ng=="
}
```

- Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 06 May 2018 09:14:54 GMT
x-log-requestid: 5AEEC78E1FFC0366B24F1C63
```

4.3.10.7. UpdateConsumerGroup

Updates the attributes of a consumer group.

Request syntax

```
PUT /logstores/<logstoreName>/consumergroups/<consumerGroupName> HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
Content-Length: <ContentLength>
{
  "timeout": <timeout>,
  "order": <order>
}
```

Request elements

- Request headers

The UpdateConsumerGroup API operation does not have special request headers. For information about common request headers of Log Service API operations, see [Common request headers](#).

- Request parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore to which the consumer group belongs.
consumerGroup	String	Yes	The name of the consumer group.
timeout	Integer	No	The timeout period. If Log Service does not receive a heartbeat packet during the timeout period, the consumer group is deleted.
order	Boolean	No	Specifies whether to consume data in sequence from a single shard. Valid values: ◦ true: The data in a single shard is consumed in sequence. The data in the original shard is first consumed after the shard is split. Then, the data in the new shards is consumed at the same time. ◦ false: The data in a single shard is randomly consumed.

 **Note** You must specify the `timeout` or `order` parameter, or specify both parameters.

Response elements

- Response headers

The UpdateConsumerGroup API operation does not have special response headers. For information about common response headers of Log Service API operations, see [Common response headers](#).

- Response parameters

The HTTP status code 200 is returned.

Error codes

In addition to [Common response headers](#), the API operation may also return the following special error codes.

HTTP status code	Error code	Error message
400	JsonInfoInvalid	timeout is of error value type
404	ProjectNotExist	The Project does not exist : {Project}
404	ConsumerGroupNotExist	consumer group not exist
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message

Examples

• Sample requests

```
PUT /logstores/logstore-test/consumerGroups/consumer-group-1 HTTP/1.1
Header:
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Fri, 04 May 2018 08:02:22 GMT
Content-Type: application/json
Content-MD5: F58544E4D022CC28A93D0B7CC208A5AA
Content-Length: 65
Connection: Keep-Alive
Body:
{
  "order": false,
  "timeout": 100
}
```

• Sample responses

```
HTTP/1.1 200
Server: nginx/1.12.1
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Fri, 04 May 2018 08:15:11 GMT
x-log-requestid: 5AEC168FA796F4195BF404CB
```

4.3.11. RAM or STS

4.3.11.1. Overview

Use RAM to authorize a RAM user to access Log Service resources of an Alibaba Cloud account

You can create projects, Logstores, Logtail configurations, and machine group in Log Service. You have full permissions on these resources. You can call all relevant API operations to use the resources.

However, if you use your Alibaba Cloud account to create a RAM user, the RAM user is unauthorized to use your resources in Log Service. To allow the RAM user to use your resources, you must grant relevant permissions on the resources to the RAM user.

You can use RAM policies to grant a RAM user the following permissions on Log Service resources:

• Upload data to a specified Logstore

If you grant this permission to a RAM user, the RAM user can upload data to the specified Logstore by using the relevant API or SDK. The policy is as follows:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:Post*",
        "log:BatchPost*"
      ],
      "Resource": ["acs:log:*:*:project/<The name of the specified project>/logstore/<The name of the specified Logstore>"],
      "Effect": "Allow"
    }
  ]
}
```

• Query data of a specified Logstore in the console

If you grant this permission to a RAM user, the RAM user can query logs, pull logs, and view the Logstore list after logging on to the Log Service console. The policy is as follows:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": ["log:List*"],
      "Resource": ["acs:log:*:*:project/<The name of the specified project>/*"],
      "Effect": "Allow"
    },
    {
      "Action": ["log:Get*"],
      "Resource": ["acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>"],
      "Effect": "Allow"
    }
  ]
}
```

You can skip this topic if you do not need to grant access to your Log Service resources across accounts. Skipping this topic does not affect your understanding and use of the other topics in the document.

Related topics:

- [Resource list](#)
- [Action list](#)
- [Authentication rules](#)

4.3.11.2. Resource list

This topic describes the resources whose permissions you can grant to a RAM user. You can use an Alibaba Cloud account to grant a RAM user specific permissions on Log Service resources.

The following table lists the resources and relevant expressions.

Resource	Expression in policies
Project and Logstore	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
Project, Logstore, and LogShipper	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/shipper/\${shipperName}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/*
Project and Logtail	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailconfig}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/*
Project and machine group	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/*
Project and consumer group	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/*
Project and saved search	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/savedsearch/\${savedSearchName}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/savedsearch/*
Project and dashboard	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/dashboard/\${dashboardName}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/dashboard/*
Project and alert	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/alert/\${alarmName}
	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/alert/*
All types of resources	acs:log:\${regionName}:\${projectOwnerAliUid}:*
	acs:log:*:\${projectOwnerAliUid}:*

Note The resources in Log Service have hierarchical relationships. Projects are root resources. Logstores, Logtail, and machine groups are parallel sub-resources of projects. LogShipper and consumer groups are sub-resources of Logstores.

The following table describes the parameters that are used in policies.

Parameter	Description
<code>\${regionName}</code>	The name of a region.
<code>\${projectOwnerAliUid}</code>	The ID of an Alibaba Cloud account.
<code>\${projectName}</code>	The name of a project.
<code>\${logstoreName}</code>	The name of a Logstore.

Parameter	Description
<code>\${logtailconfig}</code>	The name of a Logtail configuration.
<code>\${machineGroupName}</code>	The name of a machine group.
<code>\${shipperName}</code>	The name of a LogShipper rule.
<code>\${consumerGroupName}</code>	The name of a consumer group.
<code>\${savedSearchName}</code>	The name of a saved search.
<code>\${dashboardName}</code>	The name of a dashboard.
<code>\${alarmName}</code>	The name of an alert rule.

4.3.11.3. Action list

This topic describes the actions that a RAM user can perform on Log Service resources after you grant the relevant permissions to the RAM user in the RAM console.

Actions

In the RAM console, you can authorize a RAM user to perform the following actions on Log Service resources.

Action	Description
CreateProject	Creates a project.
GetLogs	Queries logs in a Logstore of a project.
GetHistograms	Queries the distribution of logs in a Logstore of a project.
GetLogstore	Queries the attributes of a Logstore.
ListLogstore	Lists all Logstores in a project.
CreateLogstore	Creates a Logstore in a project.
DeleteLogstore	Deletes a Logstore, including all shards and indexes in the Logstore.
UpdateLogstore	Updates the attributes of a Logstore.
GetCursorOrData (GetCursor and PullLogs)	Queries a cursor based on a point in time. It can also be used to query logs based on a cursor and quantity.
ListShards	Lists all available shards in a Logstore.
PostLog	Writes log data to a Logstore.
CreateConfig	Creates a Logtail configuration in a project.
UpdateConfig	Updates a Logtail configuration.
DeleteConfig	Deletes a Logtail configuration.
GetConfig	Queries the details of a Logtail configuration.
UpdateConfig	Lists all Logtail configurations in a project. You can set the relevant parameters to show the configurations on multiple pages.
CreateMachineGroup	Creates a machine group to specify the servers from which logs are collected.
UpdateMachineGroup	Updates the information of a machine group.
DeleteMachineGroup	Deletes a machine group.
GetMachineGroup	Queries the details of a machine group.
ListMachineGroup	Lists machine groups.
ListMachines	Lists your machines in a machine group that is connected with Log Service.
ApplyConfigToMachineGroup	Applies a Logtail configuration to a machine group.
RemoveConfigFromMachineGroup	Deletes a Logtail configuration from a machine group.
GetAppliedConfigs	Queries the name of Logtail configuration applied to a machine group.
CreateConsumerGroup	Creates a consumer group in a Logstore.
UpdateConsumerGroup	Modifies the attributes of a consumer group.
DeleteConsumerGroup	Deletes a consumer group.
ListConsumerGroup	Lists all consumer groups of a Logstore.

Action	Description
UpdateCheckpoint	Updates the checkpoint of a shard for a consumer group.
HeartBeat	Sends a heartbeat packet to Log Service for a consumer.
GetCheckpoint	Queries the checkpoints of one or all shards from which data is consumed by a consumer group.
CreateIndex	Creates one or more indexes for a Logstore.
DeleteIndex	Deletes one or more indexes from a Logstore.
GetIndex	Queries one or more indexes of a Logstore.
UpdateIndex	Updates one or more indexes of a Logstore.
log:CreateSavedSearch	Creates a saved search.
log:UpdateSavedSearch	Updates a saved search.
log:GetSavedSearch	Queries a saved search.
log:DeleteSavedSearch	Deletes a saved search.
log:ListSavedSearch	Queries the saved search list.
log:CreateDashboard	Creates a dashboard.
log:UpdateDashboard	Updates a dashboard.
log:GetDashboard	Queries a dashboard.
log:DeleteDashboard	Deletes a dashboard.
log:ListDashboard	Queries the dashboard list.
log:CreateJob	Creates a job, for example, an alert or subscription.
log:UpdateJob	Updates a job.
log:CreateApp	Grants a RAM user the permission to create an application, for example, Cost Manager or Log Audit Service.
log:UpdateApp	Grants a RAM user the permission to update an application, for example, Cost Manager or Log Audit Service.
log:GetApp	Grants a RAM user the permission to view an application, for example, Cost Manager or Log Audit Service.
log>DeleteApp	Grants a RAM user the permission to delete an application, for example, Cost Manager or Log Audit Service.

4.3.11.4. Authentication rules

When a RAM user calls Log Service API operations to access the resources of an Alibaba Cloud account, Log Service authenticates the RAM user. This ensures that the resource owner has granted the RAM user the relevant permissions on the resources. This topic describes the authentication rules that apply when a RAM user calls Log Service APIs to access the resources of an Alibaba Cloud account.

Logstore authentication rules

Each Log Service API operation determines which permissions to check based on the relevant resources and the API syntax. The following table describes the Logstore authentication rules for the relevant API operations.

Operation	Resource
log:GetLogStore	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
log:ListLogStores	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/*
log:CreateLogStore	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/*
log>DeleteLogStore	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
log:UpdateLogStore	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}

LogHub authentication rules

The following table describes the LogHub authentication rules of the relevant data write and consumption API operations. The GetCursor and GetLogs API operations are included in the log:GetCursorOrData API operation.

Operation	Resource
log:GetCursorOrData	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
log:ListShards	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
log:PostLogStoreLogs	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}

Configuration authentication rules

Operation	Resource
log:CreateConfig	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/*
log:UpdateConfig	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailConfigName}
log>DeleteConfig	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailConfigName}
log:GetConfig	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailConfigName}
log:ListConfig	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/*

Machine group authentication rules

Operation	Resource
log:CreateMachineGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/*
log:UpdateMachineGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
log>DeleteMachineGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
log:GetMachineGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
log:ListMachineGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/*
log:ListMachines	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}

Interactive API operations for configurations and machine groups

Operation	Resource
log:ApplyConfigToGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailConfigName} acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
log:RemoveConfigFromGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailConfigName} acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
log:GetAppliedMachineGroups	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailConfigName}
log:GetAppliedConfigs	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}

4.3.12. Description of common resources

4.3.12.1. Encrypt data

Log Service allows you to use Key Management Service (KMS) to encrypt data for secure storage. This topic describes the data encryption mechanism of Log Service and how to encrypt data by using KMS.

Data encryption mechanism

Log Service encrypts data by using KMS. The data encryption mechanism has the following characteristics:

- Log Service supports the Advanced Encryption Standard (AES) encryption algorithm.
- You can create and manage a customer master key (CMK) in the KMS console to ensure the security of the CMK.
- Log Service supports the following encryption types:
 - Service key-based encryption: Log service generates an independent service key for each Logstore. The service key never expires.
 - Bring Your Own Key (BYOK) encryption: You can create a CMK in the KMS console and grant the relevant permissions to Log Service. When Log Service calls a KMS API operation, this CMK is used to create a key for data encryption. If the CMK is deleted or disabled, the BYOK key becomes invalid.

Notice

- You can configure data encryption for a Logstore only when you call the [CreateProject](#) operation to create the Logstore. The algorithm or type of data encryption cannot be changed after the Logstore is created.
- If the CMK created in the KMS console becomes invalid, all read and write requests on the Logstore fail.
- The CMK generated by the service key of Log Service does not rotate.

Service key-based encryption

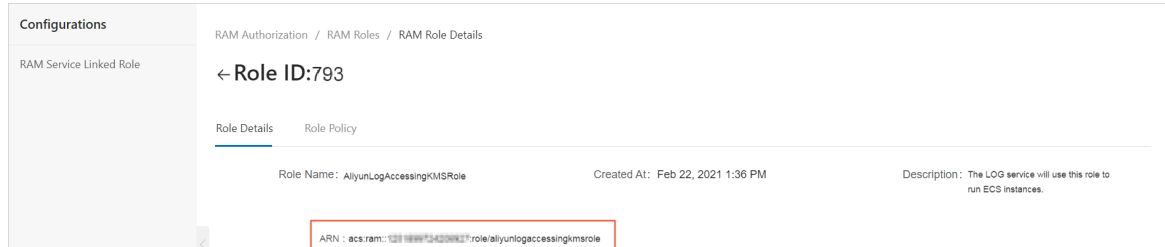
When you call the [CreateProject](#) operation to create a Logstore, add the `encrypt_conf` node to encrypt data by using the following sample code:

```
encrypt_conf = {
  "enable": True, # If you set the value to True, data encryption is enabled.
  "encrypt_type": "aes_gcm", # The encryption algorithm. You can set the value to sm4_gcm or aes_gcm.
}
```

Bring Your Own Key (BYOK) encryption

- In the KMS console, create a BYOK.

For more information, see the [Key Management Service \(KMS\) > Create a CMK](#) topic in *User Guide*.
- Create a level -1 organization.
 - Log on to the Apsara Uni-manager Management Console. For more information, see the [User Guide > Log on to the Apsara Uni-manager Management Console](#) topic.
 - Create an organization. For more information, see the [User Guide > Enterprise > Organizations > Create an organization](#) topic in *User Guide*.
- Create a RAM role for a level -1 organization.
 - In the top navigation bar of the Apsara Uni-manager Management Console, click **Configurations**.
 - On the **RAM Service Linked Role** tab, click **Create RAM Role**.
 - In the **Organization Name** drop-down list, select the created organization. In the **Service Name** drop-down list, select **Log Service**, and click **OK**.
 - On the **RAM Service Linked Role** page, find the **AliyunLogAccessingKMSRole** service role, and click **Details** to obtain the ARN of the RAM role.



- When you call the [CreateProject](#) operation to create a Logstore, add the `encrypt_conf` node to encrypt data by using the following sample code:

```
encrypt_conf = {
  "enable": True, # If you set the value to True, data encryption is enabled.
  "encrypt_type": "aes_gcm", # The encryption algorithm. You can set the value to sm4_gcm or aes_gcm.
  "user_cmk_info": {
    "cmk_key_id": "", # The ID of the CMK that you created in Step 1.
    "arn": "", # The ARN of the RAM role that is obtained in Step 3.
    "region_id": "" # The ID of the region where the CMK resides.
  }
}
```

4.3.12.2. Data model

This topic describes the data model of Log Service.

Project

A project is a basic management unit in Log Service and is used to isolate and control resources. You can use a project to manage all logs and log sources of an application.

Logstore

A Logstore is a unit that collects, stores, and consumes logs in Log Service. Each Logstore belongs to a single project. Each project can have multiple Logstores. You can create multiple Logstores for a project based on your business requirements. We recommend that you create an independent Logstore for each type of logs in an application. For example, you have a game application named big-game. The server stores three types of logs: `operation_log`, `application_log`, and `access_log`. You can create a project named big-game and then create three Logstores in the project to collect, store, and query the three types of logs.

Log

A log is the minimum data unit that is processed in Log Service. Log Service uses a semi-structured data model to define a log. The data model is as follows:

- **Topic:** a user-defined field that marks multiple logs. For example, access logs can be marked based on the sites. This field is an empty string by default. An empty string is also a valid topic.
- **Time:** a reserved field in a log, used to indicate the time when the log is generated. This value is a Unix timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
- **Content:** a field that records the content of a log. The content consists of one or more content items. Each content item is a key-value pair.
- **Source:** a field that indicates the source of a log, for example, the IP address of the machine where the log is generated. This field is empty by default.

The following table describes the requirements for the values of these fields.

Log fields

Field	Requirement
time	An integer in the Unix timestamp format. Unit: seconds.
topic	A UTF-8 encoded string of up to 128 characters.
source	A UTF-8 encoded string of up to 128 characters.
content	One or more key-value pairs. A key is a UTF-8 encoded string of up to 128 characters. It can contain only letters, underscores (_), and digits. It cannot start with a digit. A value is a UTF-8 encoded string of up to 1,048,576 (1024 × 1024) bytes.

Note The key in the content field cannot be any of the following keywords: `__time__`, `__source__`, `__topic__`, `__partition_time__`, and `__extract_others__`.

Topic

Logs in a Logstore can be grouped by topics. Topics can be specified when you write logs to a Logstore. For example, you can use your user ID as the topic of a log when you write the log to a Logstore. If you do not need to group logs in a Logstore, you can specify the same topic for all logs.

Note An empty string is a valid topic. The topic of a log is an empty string by default.

Logs are used in various formats in actual scenarios. The following example shows how to map a raw NGINX access log to the log data model of Log Service. In this example, the IP address of your NGINX server is 10.10.10.1. A raw log on this server is as follows:

```
10.1.1.1 - - [01/Mar/2012:16:12:07 +0800] "GET /Send? AccessKeyId=<yourAccessKeyId> HTTP/1.1" 200 5 "-" "Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2"
```

The following table describes how to map the raw log to the log data model of Log Service.

Log data model of Log Service

Field	Field value	Description
topic	""	The default value (empty string) is used.
time	1330589527	The time when the log was generated. The time is converted from the timestamp of the raw log. Unit: seconds.
source	"10.10.10.1"	The IP address of the server is used as the log source.
content	Key-value pair	The content of the log.

You can decide how to extract the raw log content and combine the extracted content into key-value pairs. The following table describes custom key-value pairs.

Custom key-value pairs

Key	Value
ip	"10.1.1.1"
method	"GET"
status	"200"
length	"5"
ref_url	"-"
browser	"Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2"

LogGroup

A group of logs.

Logs

A collection of multiple logs.

LogGroupList

A collection of log groups that are used to return results.

Encoding method

The following table describes the content encoding method that is supported by Log Service. The RESTful API layer is indicated by Content-Type.

Supported encoding method

Encoding method	Content-Type	Description
ProtoBuf	The data model is encoded by Protocol Buffer.	application/x-protobuf

The following sample code shows how to use Protocol Buffer to define the objects in the data model:

```
message Log
{
  required uint32 Time = 1; // Unix time format
  message Content
  {
    required string Key = 1;
    required string Value = 2;
  }
  repeated Content Contents = 2;
}
message LogGroup
{
  repeated Log Logs = 1;
  optional string Reserved = 2; // Reserved fields
  optional string Topic = 3;
  optional string Source = 4;
}
message LogGroupList
{
  repeated LogGroup logGroupList = 1;
}
```

Note When you use the Protocol Buffer format, key-value pairs must be unique. Otherwise, an undefined behavior occurs.

4.3.12.3. Data encoding

Log Service uses Protocol Buffer as the standard log format. You can write logs to Log Service in the Protocol Buffer format.

Protocol Buffer is used as a format to interchange structured data. Before you write logs to Log Service by calling relevant API operations, you must serialize raw log data to Protocol Buffer data streams as follows:

```
message Log
{
  required uint32 time = 1; // UNIX Time Format
  message Content
  {
    required string key = 1;
    required string value = 2;
  }
  repeated Content contents = 2;
}
message LogGroup
{
  repeated Log logs = 1;
  optional string reserved = 2; // An internal field. You do not need to specify this field.
  optional string topic = 3;
  optional string source = 4;
}
message LogGroupList
{
  repeated LogGroup logGroupList = 1;
}
```

Note

- When you use the Protocol Buffer format, key-value pairs must be unique. Otherwise, an undefined behavior occurs.
- For more information about the Protocol Buffer format, visit [GitHub](#).
- For more information about the API operations that write logs to Log Service, see [PostLog](#).

4.3.12.4. Logstores

A Logstore is a data storage unit. You can create up to 10 Logstores for each project by default. The name of a Logstore must be unique in the project to which the Logstore belongs. A Logstore is the portal for all log data. You can read data from or write data to the Logstore.

The naming rules for Logstores are as follows:

- The name can only contain lowercase letters, digits, hyphens (-), and underscores (_).
- The name must start and end with a lowercase letter or a digit.
- The name must be 2 to 63 characters in length.

Example

```
{
  "logstoreName" : "access_log",
  "ttl": 1,
  "shardCount": 2,
  "createTime":1439538649,
  "lastModifyTime":1439538649
}
```

The following table describes the Logstore parameters.

Logstore parameters

Parameter	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore. The name must be unique in the project to which the Logstore belongs.
ttl	Integer	Yes	The time to live (TTL) of log data. Unit: days. Minimum value: 1.
shardCount	Integer	Yes	The number of shards in the Logstore.
createTime	Integer	No	The time when the Logstore was created in Log Service. This parameter is displayed only in the output result.
lastModifyTime	Integer	No	The time when the Logstore is updated in Log Service. This parameter is displayed only in the output result.

4.3.12.5. Shards

A shard is a basic read/write unit in each Logstore. You can specify the number of shards in each Logstore.

To read data from a shard, you must specify the shard. To write data to a shard, you do not need to specify the shard. Instead, you can use the load balancing mode. In this mode, data is written to multiple shards based on automatic balancing. This ensures high availability of the write operation.

The following table describes the parameters of a shard.

Shard parameters

Parameter	Type	Required	Description
shardID	Integer	Yes	The unique ID of a shard in the Logstore. The shard ID is automatically generated by the system.

4.3.12.6. Logtail configurations

You can create up to 100 Logtail configurations for a project. The name of a Logtail configuration must be unique in a project.

You can use a Logtail configuration to specify the location, method, and parameters for log collection.

The naming rules for Logtail configurations are as follows:

- The name can only contain lowercase letters, digits, hyphens (-), and underscores (_).
- The name must start and end with a lowercase letter or a digit.
- The name must be 2 to 128 characters in length.

Example

```
{
  "configName": "testcategory1",
  "inputType": "file",
  "inputDetail": {
    "logType": "common_reg_log",
    "logPath": "/var/log/httpd/",
    "filePattern": "access.log",
    "localStorage": true,
    "timeFormat": "%Y/%m/%d %H:%M:%S",
    "logBeginRegex": ".*",
    "regex": "(\\w+) (\\s+)",
    "key": ["key1", "key2"],
    "filterKey": ["key1"],
    "filterRegex": ["regex1"],
    "topicFormat": "none"
  },
  "outputType": "sls",
  "outputDetail": {
    "logstoreName": "perfcounter"
  },
  "createTime": 1400123456,
  "lastModifyTime": 1400123456
}
```

The following table describes the configuration parameters.

Configuration parameters

Parameter	Type	Required	Description
configName	String	Yes	The name of the Logtail configuration. The name must be unique in a project.
inputType	String	Yes	The input type. Default value: file.
inputDetail	json	Yes	See the following table.
outputType	String	Yes	The output type. Only LogService is supported.
outputDetail	String	Yes	See the following table.
createTime	Integer	No	The time when the Logtail configuration was created.
lastModifyTime	Integer	No	The time when the Logtail configuration was last updated on Log Service.

The following table describes the elements of the inputDetail parameter.

inputDetail elements

Element	Type	Required	Description
logType	String	Yes	The log type. Valid value: common_reg_log.
logPath	String	Yes	The parent directory where the log resides, for example, <code>/var/logs/</code> .
filePattern	String	Yes	The pattern of the log file, for example, <code>access*.log</code> .
localStorage	Boolean	Yes	Specifies whether to enable the local cache. Logs up to 1 GB can be locally cached when the link to Log Service is disconnected.
timeFormat	String	Yes	The format of the log time, for example, <code>%Y/%m/%d %H:%M:%S</code> .
logBeginRegex	String	Yes	The regular expression that specifies the pattern of the first line of a log. This parameter is applicable to multiline logs.
regex	String	Yes	The regular expression that is used to extract logs.
key	Array	Yes	The key that is generated after logs are extracted.
filterKey	Array	Yes	The key that is used to filter logs. A log meets the requirements only when the key value matches the regular expression specified in the corresponding filterRegex column.
filterRegex	Array	Yes	The regular expression that is specified for each filterKey. The length of filterRegex must be the same as that of filterKey.
topicFormat	String	No	Specifies part of the log file path as the topic, for example, <code>/var/log/(.*)log</code> . The default value is none, indicating that the topic is null.
preserve	Boolean	No	Valid values: true and false. The value true indicates that the directory monitoring never times out. The value false indicates that the timeout period for directory monitoring is 30 minutes. Default value: true.
preserveDepth	Integer	No	Specifies the depth of the monitored directories if the preserve parameter is set to false. Maximum value: 3.

The following table describes the elements of the outputDetail parameter.

outputDetail elements

Element	Type	Required	Description
logstoreName	String	Yes	The name of the Logstore.

4.3.12.7. Logtail machine group

After a machine with Logtail is started, the machine is automatically associated with the user specified in the Logtail configurations. The machine can be identified by using the following three methods:

- IP address: the IP address corresponding to the hostname. This is the simplest method, but the IP address may be duplicated in environments such as Virtual Private

Cloud (VPC).

- Universally unique identifier (UUID): the UUID of the desktop management interface (DMI) device. For more information, visit [RFC 4122](#).
- User-defined ID: the user-defined machine identifier in the Logtail directory.

The attributes of each machine are as follows:

```
{
  "ip" : "testip1",
  "machine-uniqueid" : "testuuid1",
  "userdefined-id" : "testuserdefinedid1",
  "lastHeartbeatTime":1397781420
}
```

The following table describes the machine attribute parameters.

Parameters

Parameter	Type	Description
ip	String	The IP address corresponding to the machine hostname.
machine-uniqueid	String	The unique primary key marked by the machine. The primary key is uploaded by Logtail.
userdefined-id	String	The user-defined machine identifier that is uploaded by Logtail.
lastHeartbeatTime(output-only)	Integer	The last heartbeat time of the machine. The value is a Unix timestamp representing the number of seconds that have elapsed since the epoch time.

machinegroup

The machinegroup parameter indicates the machine group that belongs to the current user in the project. The machine group can be identified by an IP address or user-defined ID. IP addresses are more easily identified than user-defined IDs. However, user-defined IDs can identify machine groups in VPC even when repeated IP addresses exist. You can select either of the two machine identification methods based on your business requirements.

The naming rules of machine groups are as follows:

- The name can contain only lowercase letters, digits, hyphens (-), and underscores (_).
- The name must start and end with a lowercase letter or digit.
- The name must be 2 to 128 characters in length.

Example

```
{
  "groupName" : "testgroup",
  "groupType" : "",
  "groupAttribute" : {
    "externalName" : "testgroup",
    "groupTopic": "testgrouptopic"
  },
  "machineIdentifyType": "ip",
  "machineList" : [
    "ip1",
    "ip2"
    ...
  ],
  "createTime": 1431705075,
  "lastModifyTime" : 1431705075
}
```

The following table lists the parameters of a machine group.

Machine group parameters

Parameter	Type	Required	Description
groupName	String	Yes	The name of the machine group. The name must be unique in a project.
groupType	String	No	The type of the machine group. Default value: null.
machineIdentifyType	String	Yes	The machine identification type. Valid values: ip or userdefined-id.
groupAttribute	object	Yes	The attribute of the machine group. Default value: null.
machineList	Array	Yes	The list of machine identifiers. Valid values: IP addresses or user-defined identifiers.
createTime(output-only)	Integer	No	The time when the machine group was created.
lastModifyTime(output-only)	Integer	No	The time when the machine group was last updated on Log Service.

The following table describes the parameters of groupAttribute.

groupAttribute parameters

Parameter	Type	Required	Description
groupTopic	String	No	The topic of the machine group. Default value: null.
externalName	String	No	The external management identifier on which the machine group depends. Default value: null.

5.CSG

5.1. Preparations

5.1.1. Log on to the API Tool console

You can log on to the API Tool console from the Apsara Uni-manager Management Console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment personnel.
 - A browser is available. We recommend that you use Google Chrome.
1. Enter the endpoint of the Apsara Uni-manager Management Console in the address bar and press Enter.
 2. Enter your username and password.

Obtain the username and password that are used to log on to the Apsara Uni-manager Management Console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your account. For security purposes, your password must meet the minimum complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
 4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
 - You have enabled MFA and bound an MFA device.
- Enter a six-digit MFA verification code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Others > API Tool**.

5.1.2. View the information about an API operation

This topic describes how to view the information about API operations. Before you use SDKs, you need to obtain service names, API operation names, and API versions.

Context

The API operations on the API Directory page are only for reference. For information about how to call API operations, see developer guide of the corresponding cloud service. If the information on the API Directory page is inconsistent with that in the developer guide of the corresponding cloud service, the information in the developer guide prevails.

1. [Log on to the API Tool console](#).
2. In the left-side navigation pane, click **API Directory**.
3. On the **API Directory** page, select **sgw** from the drop-down list, enter a keyword of an operation name in the search box, and then click Search.
4. View and record the service name, operation name, and API version.

5.1.3. Create an AccessKey pair

AccessKey credentials are assigned by Resource Access Management (RAM) or Security Token Service (STS). You can use one type of AccessKey credentials for authentication when you call API operations. This topic describes how to obtain an AccessKey pair.

Context

An AccessKey ID and an AccessKey secret are used to implement symmetric encryption and verify the identity of request senders. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt a signature string.

AccessKey credentials can be assigned by RAM or STS:

- An AccessKey pair that is assigned by RAM to a third party consists of an AccessKey ID and an AccessKey secret.
- An AccessKey 3-tuple that is assigned by STS to a third party consists of an AccessKey ID, an AccessKey secret, and a security token. For more information, see [Obtain an STS AccessKey 3-tuple](#).

The Apsara Uni-manager Management Console provides AccessKey pairs for personal accounts or organizations. Only the administrators of level-1 organizations and operations administrators can obtain the AccessKey pairs of organizations. We recommend that you use the AccessKey pairs of personal accounts to call the operations provided by the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, configure the header parameters that are described in the following table for access control. Otherwise, an error that indicates insufficient permissions can occur.

 **Warning** The AccessKey pair of a personal account is managed and controlled by the permission system of the Apsara Uni-manager Management Console. If you need to obtain the AccessKey pair of an organization, make sure that your operation is allowed by an administrator. This ensures security.

Parameter	Description
x-acs-regionid	The ID of the region. Example: cn-hangzhou-*
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For information about how to obtain an organization ID, see the "GetOrganizationList" section in Apsara Uni-manager Management Console Developer Guide.
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For information about how to obtain a resource set ID, see the "ListResourceGroup" section in Apsara Uni-manager Management Console Developer Guide.
x-acs-instanceid	The ID of the instance that you want to manage. If you want to obtain the ID of an Elastic Compute Service (ECS) instance, you can obtain the ID from the ECS instance list.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. Click the profile picture in the upper-right corner of the page and click **User Information**.
3. In the **AccessKey Pair** section, view the AccessKey pair of the personal account.

AccessKey Create AccessKey Pair

! The AccessKey ID and AccessKey Secret are the keys used to access the cloud service resources. They have full permissions on the account. Please keep them properly.

AccessKey ID	AccessKey Secret	Status	Created At	Actions
W1YRFvKH*****	*****	Enable	Sep 18, 2021, 11:05:57	Disable Delete
IsJsAUyX*****	*****	Enable	Sep 10, 2021, 12:28:35	Disable Delete

? **Note** An AccessKey pair consists of an AccessKey ID and an AccessKey secret. These credentials provide full permissions on cloud resources within your account. You must keep the AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization list, click the name of the level-1 organization whose AccessKey pair you want to obtain.
5. Click **Management AccessKey** on the right side of the page.
6. In the dialog box that appears, view the AccessKey pair of the organization.

5.1.4. Obtain an STS AccessKey pair

This topic describes how to use Security Token Service (STS) to call API operations in Apsara Stack.

Background information

STS allows you to manage temporary credentials to your Apsara Stack resources. You can use STS to call API operations in Apsara Stack.

An STS AccessKey 3-tuple consists of an AccessKey ID, an AccessKey secret, and a security token. When you use STS to call API operations, you must specify an STS AccessKey 3-tuple. For more information about STS, see [What is STS](#).

Procedure

1. Obtain the `RAM role`.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. Move the pointer over the profile picture in the upper-right corner of the page and click **Personal Information**.
 - iii. Click **View Current Role Policy**.

In the **View Policies of Current Role** dialog box, view your `RAM role`.
2. Obtain the STS AccessKey 3-tuple.

Note

In the following code, replace the variable parameters with your actual information. Delete the angle brackets (<>) in the final code.

The following list describes the variable parameters:

- <yourRegionID>: Replace it with the region ID.
- <yourAccessKeyID>: Replace it with your AccessKey ID.
- <yourAccessKeySecret>: Replace it with your AccessKey secret.
- <yourRoleSessionName>: Replace it with the name of the current session.
- <yourRoleArn>: Replace it with your RAM role.
- "sts.aliyuns.com": Replace it with the endpoint of STS.

```
public static void main(String[] args) {
    // Create a DefaultAcsClient instance and initialize it.
    // Specify the region ID, AccessKey ID, and AccessKey secret.
    DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKeyID>", "<yourAccessKeySecret>");
    HttpClientConfig clientConfig = HttpClientConfig.getDefault();
    clientConfig.setIgnoreSSLCerts(true);
    clientConfig.setProtocolType(ProtocolType.HTTPS);
    profile.setHttpClientConfig(clientConfig);
    IAcsClient client = new DefaultAcsClient(profile);
    String assumeRole = callAssumeRole(client);
    if (StringUtils.isEmpty(assumeRole)) return;
    JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
    JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
    // Obtain the STS AccessKey 3-tuple.
    String stsAK = credentials.getString("AccessKeyId");
    String stsSK = credentials.getString("AccessKeySecret");
    String stsToken = credentials.getString("SecurityToken");
}

private static String callAssumeRole(IAcsClient client) {
    CommonRequest request = new CommonRequest();
    // Replace "sts.aliyuns.com" with the endpoint of STS. The format of an STS endpoint is sts-vpc.${global:region}.${global:internet-domain}.
    request.setSysDomain("sts.aliyuns.com");
    request.setSysProduct("Sts");
    request.setSysAction("AssumeRole");
    request.setSysVersion("2015-04-01");
    request.setSysMethod(MethodType.POST);
    request.setHttpContentType(FormatType.FORM);
    // Replace "<yourRoleSessionName>" with the name of the current session.
    request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
    // Replace "<yourRoleArn>" with the RAM role ARN that is obtained in the previous step.
    request.putQueryParameter("RoleArn", "<yourRoleArn>");
    try {
        CommonResponse response = client.getCommonResponse(request);
        return response.getData();
    } catch (ServerException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    } catch (ClientException e) {
        System.out.println(e.getErrCode());
        System.out.println(e.getErrMsg());
    }
    return null;
}
```

3. Use the STS AccessKey 3-tuple to call an API operation of a cloud service. The following sample code provides an example on how to call the

`DescribeGatewaysRequest` operation of Cloud Storage Gateway (CSG).

Note

In the following code, replace the variable parameters with your actual information. Delete the angle brackets (<>) in the final code.

The following list describes the variable parameters:

- <yourRegionID>: Replace it with the region ID.
- <yourStsAccesskeyID>: Replace it with your STS AccessKey ID.
- <yourStsAccessKeySecret>: Replace it with your STS AccessKey secret.
- <yourStsToken>: Replace it with your security token.
- <yourResourceGroupID>: Replace it with the resource set ID.
- <yourRegionID>: Replace it with the actual organization ID.

```

import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.sgw.model.v20180511.DescribeGatewaysRequest;
import com.aliyuncs.sgw.model.v20180511.DescribeGatewaysResponse;
public class Main {
    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace it with the region ID.
            "<yourAccessKeyID>", // Replace it with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace it with your AccessKey secret.
        // Specify the endpoint of the current service. Replace the variable parameter with the endpoint.
        profile.addEndpoint("<yourRegionID>", "<Product>", "<Endpoint>");
        // Configure the request parameters.
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setIgnoreSSLCerts(true); // Ignore this parameter.
        clientConfig.setReadTimeoutMillis(<readTimeout>); // Optional. Specify the read timeout period. Unit: milliseconds.
        clientConfig.setConnectionTimeoutMillis(<connectionTimeout>); // Optional. Specify the connection timeout period. Unit: milliseconds.
        clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>); // Optional. Specify the maximum number of requests per host.
        clientConfig.setMaxRequests(<maxRequests>); // Optional. Specify the maximum number of requests.
        clientConfig.setProtocolType(ProtocolType.HTTP); // Optional. Specify the request protocol. Default value: HTTP.
        /* Optional. Specify the IP address and port of the proxy.
        clientConfig.setHttpProxy("http://192.168.XX:33XX");
        Or
        clientConfig.setHttpsProxy("https://192.168.XX:33XX");
        */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the required parameters.
        DescribeGatewaysRequest request = new DescribeGatewaysRequest();
        // Configure service-related parameters.
        request.setPageSize(10);
        request.setPageNumber(1);
        // Send the request and handle the response or exception.
        try {
            DescribeGatewaysResponse response = client.getAcsResponse(request);
            for (DescribeGatewaysResponse.Gateway gateway : response.getGateways()) {
                System.out.println(gateway.getGatewayId());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}

```

5.1.5. Obtain the values of common header parameters

You must set header parameters when you call API operations. This topic describes the header parameters and how to obtain the values of the parameters. This topic also describes the return values of some parameters after you call API operations.

Header parameters

Parameter	Description
x-acs-regionid	The ID of the region. For more information about how to obtain region IDs, see Obtain a region ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain organization IDs, see Obtain an organization ID .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain resource set IDs, see Obtain a resource set ID .
x-acs-instanceid	The ID of the gateway on which you want to perform operations. For more information about how to obtain gateway IDs, see Obtain a gateway ID .

Query a region ID

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Regions**.
4. Click the name of the organization that you want to view.
5. View the region ID in the **Regions** section on the right.

Obtain an organization ID

You must call an API operation to obtain an organization ID. For more information about how to obtain organization IDs, see the *GetOrganizationList* section in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a resource set ID

You must call an API operation to obtain a resource set ID. For more information about how to obtain resource set IDs, see the *ListResourceGroup* section in *Apsara Uni-manager Management Console Developer Guide*.

Obtain a gateway ID

You can obtain the ID of a gateway in the cloud service console of the Apsara Uni-manager Management Console. Perform the following operations to obtain a gateway ID:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Storage > Object Storage Service**.
3. In the left-side navigation pane, click **Cloud Storage Gateways**.
4. On the Cloud Storage Gateways page, view the **gateway ID**.

Response parameters

The following response parameters are returned by default when you call operations to query the instances of cloud services.

Parameter	Description
ResourceGroupName	The name of the resource set.
ResourceGroup	The ID of the resource set.
DepartmentName	The name of the organization.
Department	The ID of the organization.

5.1.6. Obtain the endpoint of Cloud Storage Gateway

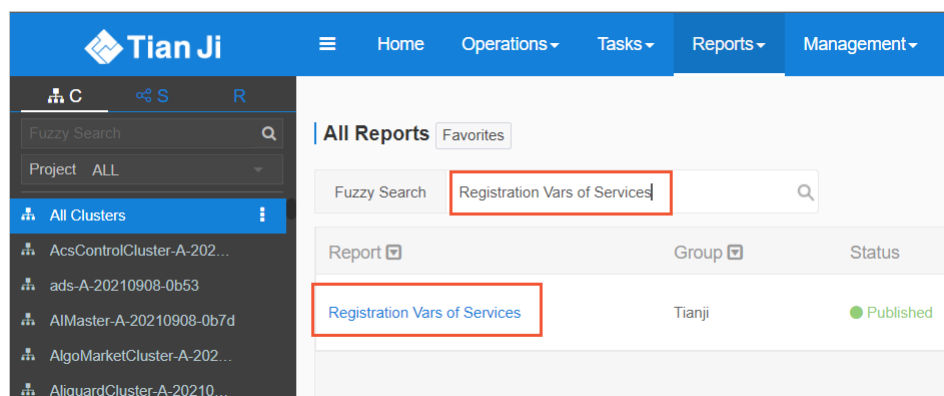
This topic describes how to obtain the endpoint of VPN Gateway.

Background information

To use the POP gateway to call API operations in Apsara Stack, you must specify the endpoint of a cloud service. You can obtain service endpoints from Apsara Stack onsite O&M engineers. You can also perform the following steps to obtain an endpoint.

Procedure

1. Log on to Apsara Infrastructure Management.
 - i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, click **Products**. Then, choose **Platforms > Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.
3. On the **All Reports** page, search for `Registration Vars of Services`. In the report list that appears, click the name of the report.



4. On the **Registration Vars of Services** page, click the  icon next to **Service** to search for `csg-controller`.
5. Find `csg-controller`. Right-click the value in the **Service Registration** column and select **Show More**.

On the **Details** page, view the value of `csg-console-pop-endpoint`, which is the `endpoint`.

5.1.7. Obtain Cloud Storage Gateway SDKs

This topic describes how to obtain Cloud Storage Gateway SDKs.

You can go to the [Apsara Stack documentation](#) page and download the Cloud Storage Gateway SDKs in the **Other Resources** section at the lower part of the page.

5.2. Call examples of SDK for Java

5.2.1. Quick start

This topic describes how to obtain and use Cloud Storage Gateway (CSG) SDK for Java.

Prerequisites

- An authorized account and an AccessKey pair are obtained. The AccessKey pair consists of an AccessKey ID and an AccessKey secret. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of CSG is obtained. For more information, see [Obtain the endpoint of Cloud Storage Gateway](#).
- Java Development Kit (JDK) 1.8 or later is installed before you can use Apsara Stack SDK for Java.

Install CSG SDK for Java

If you use Apache Maven to manage a Java project, you need to only import the JAR file of Apsara Stack SDK for Java to the pom.xml file of the project.

To use a CSG SDK, you must install the SDK core library and the CSG SDK. We recommend that you use version 4.5.17 or later for the SDK core library. In this example, the CSG SDK uses version 3.1.6.

Use CSG SDK for Java

To use CSG SDK for Java, perform the following steps:

1. Create and initialize a DefaultAcsClient instance.
2. Create an API request and set the required parameters.
3. Initiate the API request and process the response or exception.

Note

In the following code, replace the variable parameters with your actual information. Delete the angle brackets (<>) in the final code.

The following list describes the variable parameters:

- <yourRegionID>: Replace it with the region ID.
- <yourAccessKeyID>: Replace it with your AccessKey ID.
- <yourAccessKeySecret>: Replace it with your AccessKey secret.
- <Product>: Replace it with the name of the current service.
- <Endpoint>: Replace it with the endpoint of the current service.
- <readTimeout>: Replace it with a read timeout period based on your business requirements. Unit: milliseconds.
- <connectionTimeout>: Replace it with a connection timeout period based on your business requirements. Unit: milliseconds.
- <maxRequestsPerHost>: Replace it with the maximum number of requests that are supported by each host.
- <maxRequests>: Replace it with the maximum number of requests.
- "http://192.168.XX:33XX": Replace it with the proxy IP address and port number.

```

import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.sgw.model.v20180511.DescribeGatewaysRequest;
import com.aliyuncs.sgw.model.v20180511.DescribeGatewaysResponse;
public class Main {
    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile(
            "<yourRegionID>", // Replace it with the region ID.
            "<yourAccessKeyId>", // Replace it with your AccessKey ID.
            "<yourAccessKeySecret>"); // Replace it with your AccessKey secret.
        // Specify the endpoint of the current service. Replace the variable parameter with the endpoint.
        profile.addEndpoint("<yourRegionID>", "<Product>", "<Endpoint>");
        // Configure the request parameters.
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setIgnoreSSLCerts(true); // Ignore this parameter.
        clientConfig.setReadTimeoutMillis(<readTimeout>); // Optional. Specify the read timeout period. Unit: milliseconds.
        clientConfig.setConnectionTimeoutMillis(<connectionTimeout>); // Optional. Specify the connection timeout period. Unit: milliseconds.
        clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>); // Optional. Specify the maximum number of requests per host.
        clientConfig.setMaxRequests(<maxRequests>); // Optional. Specify the maximum number of requests.
        clientConfig.setProtocolType(ProtocolType.HTTP); // Optional. Specify the request protocol. Default value: HTTP.
        /* Optional. Specify the IP address and port of the proxy.
        clientConfig.setHttpProxy("http://192.168.XX:33XX");
        Or
        clientConfig.setHttpsProxy("https://192.168.XX:33XX");
        */
        profile.setHttpClientConfig(clientConfig);
        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the required parameters.
        DescribeGatewaysRequest request = new DescribeGatewaysRequest();
        // Configure service-related parameters.
        request.setPageSize(10);
        request.setPageNumber(1);
        // Send the request and handle the response or exception.
        try {
            DescribeGatewaysResponse response = client.getAcsResponse(request);
            for (DescribeGatewaysResponse.Gateway gateway : response.getGateways()) {
                System.out.println(gateway.getGatewayId());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}

```

5.2.2. Use CSG SDK for Java

5.2.2.1. Install SDK for Java

This topic describes how to install and call Cloud Storage Gateway (CSG) SDK for Java.

Apsara Stack SDK for Java supports Java Development Kit (JDK) 1.8 or later. You can install Apsara Stack SDK for Java by adding Maven dependencies or downloading [Apsara Stack SDK](#).

To use the development tool kit for CSG, you must install the core library of Apsara Stack SDK for Java and CSG SDK for Java. We recommend that you use version 4.5.17 or later for the core library. In this example, the version of CSG SDK is 3.1.6.

To use CSG SDK for Java, you must install the core library of Apsara Stack SDK for Java and add the following dependencies to the pom.xml file. For more information about the specific SDK version, see [Apsara Stack SDK](#).

```

<!-- Alibaba Cloud SDK for Java -->
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.17</version>
</dependency>
<!-- nas java sdk -->
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-sgw</artifactId>
    <version>3.1.6</version>
</dependency>

```

5.2.2.2. Configure authentication credentials

When you use Apsara Stack SDK to access Apsara Stack services, you must provide an Apsara Stack tenant account for authentication. This topic describes how to configure authentication credentials.

Apsara SDK for Java supports authentication through AccessKey pairs. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

If you want to use your AccessKey pair to access Apsara Stack, you must configure the AccessKey pair as the credentials when you initialize the client.

Note

Do not disclose any code containing your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.

```
DefaultProfile profile = DefaultProfile.getProfile(
    "<yourRegionID>",           // Replace it with the region ID.
    "<yourAccessKeyID>",       // Replace it with your AccessKey ID.
    "<yourAccessKeySecret>"); // Replace it with your AccessKey secret.
```

Note

In the preceding code, replace the variable parameters with your actual information. Delete the angle brackets (<>) in the final code.

5.2.2.3. Configure HttpClientConfig

This topic describes how to configure HttpClientConfig.

You can use the following code to configure HttpClientConfig. You must select and configure parameters based on your business requirements.

Note

In the following code, replace the variable parameters with your actual information. Delete the angle brackets (<>) in the final code.

The following list describes the variable parameters:

- <readTimeout>: Specify a read timeout period based on your business requirements. Unit: milliseconds.
- <connectionTimeout>: Specify a connection timeout period based on your business requirements. Unit: milliseconds.
- <maxRequestsPerHost>: Specify the maximum number of requests for each server based on your business requirements. The value is in the INT format.
- <maxRequests>: Specify the maximum number of requests based on your business requirements. The value is in the INT format.
- "http://192.168.XX:33XX": Replace it with the IP address and port of the proxy.

```
// Obtain the HttpClientConfig object.
HttpClientConfig clientConfig = HttpClientConfig.getDefault();
// Specify whether to ignore the certificate.
clientConfig.setIgnoreSSLCerts(true);
// Optional. Specify the read timeout period. Unit: milliseconds.
clientConfig.setReadTimeoutMillis(<readTimeout>);
// Optional. Specify the connection timeout period. Unit: milliseconds.
clientConfig.setConnectionTimeoutMillis(<connectionTimeout>);
// Optional. Specify the maximum number of requests for each server. The value is in the INT format.
clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>);
// Optional. Specify the maximum number of requests. The value is in the INT format.
clientConfig.setMaxRequests(<maxRequests>);
// Optional. Specify the protocol type.
clientConfig.setProtocolType(ProtocolType.HTTP);
/** Optional. Specify whether to configure a proxy based on your business requirements.
clientConfig.setHttpProxy("http://192.168.XX.XX:33XX");
Or
clientConfig.setHttpsProxy("https://192.168.XX.XX:33XX");
Replace it with the IP address and port of the proxy.
*/
profile.setHttpClientConfig(clientConfig);
```

5.2.2.4. Initiate a request

This topic describes how to use Cloud Storage Gateway (CSG) SDK for Java to initiate a request.

1. Create an AcsClient instance.

```
IAcsClient client = new DefaultAcsClient(profile);
```

2. Create a request.

The naming convention for the Request class is ``${apiName}Request``, where ``${apiName}`` indicates the API name, such as `DescribeGateways`.

When multiple service SDKs are used, different requests may have the same name. You can differentiate the requests based on package names.

```
DescribeGatewaysRequest request = new DescribeGatewaysRequest();
// Configure service-related parameters.
request.setPageSize(10);
request.setPageNumber(1);
```

3. Initiate the request and handle the response.

```
DescribeGatewaysResponse response;
try {
    response = client.getAcsResponse(request);
    for (DescribeGatewaysResponse.Gateway gateway:response.getGateways()) {
        System.out.println(gateway.getGatewayId());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
```

In normal cases, all returned fields are serialized into the response. You can directly use `response.getXXX()` to obtain the response fields.

```
String name = response.getName();
```

If an exception occurs or you need to check the original HTTP response, use `doAction()` to obtain the original response.

```
HttpResponse response = client.doAction(request);
```

5.2.2.5. Handle errors

This topic describes how to handle the errors of Cloud Storage Gateway (CSG) SDK for Java.

If the server or SDK encounters an error when you use CSG SDK for Java, it returns an error response. The error response contains an error code and an error message.

In most cases, you do not need to handle the errors returned by CSG SDK for Python. You only need to handle the errors returned by the server based on error messages.

- `ServerException` indicates a server error.
- `ClientException` indicates an SDK error.

For example, when the following error occurs, you can modify the AccessKey ID based on the error message to solve the problem:

```
com.aliyuncs.exceptions.ClientException: InvalidAccessKeyId.NotFound : Specified access key is not found.
```

If you want to handle the SDK errors returned by CSG SDK for Java, refer to the following example to write your code:

```
try {
    FooResponse response = client.getAcsResponse(request);
    // Handle the response.
    //...
} catch (ServerException e) {
    // You can add your own error handling logic here.
    // For example, print the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
    // If you are unable to troubleshoot the problem, submit a ticket and provide RequestId to Alibaba Cloud technical support.
    System.out.println("ResponseId=" + e.getRequestId());
} catch (ClientException e) {
    // You can add your own error handling logic here.
    // For example, print the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
}
```

5.3. Call examples of SDK for Python

5.3.1. Quick start

This topic describes how to obtain and use Cloud Storage Gateway (CSG) SDK for Python.

Prerequisites

- An authorized account and an AccessKey pair are obtained. The AccessKey pair consists of an AccessKey ID and an AccessKey secret. For more information, see [Obtain an AccessKey pair](#).
- The endpoint of CSG is obtained. For more information, see [Obtain the endpoint of Cloud Storage Gateway](#).

Install CSG SDK for Python

1. Install the SDK core library.

- If you use Python 2.x, run the following command to install the core library:

```
pip install aliyun-python-sdk-core
```

- If you use Python 3.x, run the following command to install the core library:

```
pip3 install aliyun-python-sdk-core-v3
```

2. Install CSG SDK

Before you install CSG SDK, you must download CSG SDK. For more information, see [Obtain Cloud Storage Gateway SDK](#).

- If you use Python 2.x, decompress the SDK package and run the following command to install CSG SDK:

```
python setup.py install
```

- If you use Python 3.x, decompress the SDK package and run the following command to install CSG SDK:

```
python3 setup.py install
```

Use CSG SDK for Python

To use the SDK for Python, perform the following steps:

1. Create a client instance. When you create a client instance, you must obtain the region ID, AccessKey ID, and AccessKey secret.
2. Create an API request and set the required parameters.
3. Initiate the API request and process the response or exception.

Note In the following code, replace the variable parameters with your actual information. Delete the angle brackets (<>) in the final code.

The following list describes the variable parameters:

- <yourRegionID>: Replace it with the region ID.
- <yourAccessKeyID>: Replace it with your AccessKey ID.
- <yourAccessKeySecret>: Replace it with your AccessKey secret.
- <sgw.aliyuncs.com>: Replace it with the endpoint of the service.

```
#!/usr/bin/env python
# coding=utf-8
from aliynsdksdkcore import client
from aliynsdksdkcore.profile import region_provider
from aliynsdksdkcore.request.v20180511 import DescribeGatewaysRequest
# global config
region_id = '<yourRegionID>' # Replace it with the region ID.
access_key = '<yourAccessKeyID>' # Replace it with the AccessKey ID.
access_secret = '<yourAccessKeySecret>' # Replace it with the AccessKey secret.
# product config
product_name = 'sgw' # Specify the service name.
endpoint = '<sgw.aliyuncs.com>' # Replace it with the endpoint of the service.
# create client
region_provider.modify_point(product_name, region_id, endpoint)
clt = client.AcsClient(access_key, access_secret, region_id)
# setup request
request = DescribeGatewaysRequest.DescribeGatewaysRequest()
request.set_endpoint(endpoint)
request.set_content_type("application/json")
request.set_accept_format("JSON")
# get response
response = clt.do_action_with_exception(request)
print(response)
```

5.3.2. Use CSG SDK for Python

5.3.2.1. Install CSG SDK for Python

This topic describes how to install and call Cloud Storage Gateway (CSG) SDK for Python.

Apsara Stack SDK for Python supports Python 2.6.x, Python 2.7.x, Python 3.x, or later. You can install Apsara Stack SDK for Python by using pip.

Note To use the development tool kit for CSG, you must install the core library of Apsara Stack SDK for Python and CSG SDK for Python.

- If you use Python 2.x, run the following command to install the core library:

```
pip install aliyun-python-sdk-core
```

- If you use Python 3.x, run the following command to install the core library:

```
pip3 install aliyun-python-sdk-core-v3
```

Install CSG SDK

Before you install CSG SDK for Python, you must download the SDK package. For more information, see [Obtain Cloud Storage Gateway SDKs](#).

- If you use Python 2.x, decompress the SDK package and run the following command to install the SDK:

```
python setup.py install
```

- If you use Python 3.x, decompress the SDK package and run the following command to install the SDK:

```
python3 setup.py install
```

5.3.2.2. Configure authentication credentials

When you use Apsara Stack SDK to access Apsara Stack services, you must provide an authorized account for authentication.

Apsara Stack SDK for Python supports authentication with an AccessKey pair. For more information about how to obtain an AccessKey pair, see [Obtain an AccessKey pair](#).

If you want to use your AccessKey pair to access Apsara Stack, you must configure the AccessKey pair as the credentials when you initialize the client.

Notice Do not disclose any code containing your AccessKey pair. For example, do not commit the code to public GitHub projects. Otherwise, your account may be compromised.

```
clt = client.AcsClient(secret_id, secret_key, region_id)
```

5.3.2.3. Initiate HTTPS requests

Apsara Stack SDK for Python allows you to initiate API requests over HTTP and HTTPS.

You can specify a protocol (HTTP or HTTPS) for a specific request, or set a global default protocol.

Note The default protocol of the service takes precedence over the global default protocol.

Add OpenSSL support

The HTTPS protocol support of Apsara Stack SDK for Python is based on the OpenSSL support in Python. To use Apsara Stack SDK for Python to make requests over HTTPS, you need to add OpenSSL support in Python. OpenSSL supports TLS 1.2 or later.

Run the `python -c "import ssl"` command to check whether OpenSSL is supported in Python. If

```
ImportError: No module named ssl
```

is not displayed in the command output, OpenSSL is already supported.

If OpenSSL is not supported, run the following command to install it:

```
pip install pyopenssl
```

Example: Set the protocol type of a request to HTTPS.

The following code shows how to set HTTPS as the protocol for a request:

```
request = DescribeGatewaysRequest.DescribeGatewaysRequest()
request.set_protocol_type("https")
# Valid value: https or http
```

Example: Set a global default protocol.

The following code shows how to set a global protocol for all service requests:

```
import aliyunsdkcore.request
aliyun_sdkcore.request.set_default_protocol_type("https")
# Create a request and call client.do_action_with_exception() to send the request.
```

5.3.2.4. Initiate a request

This topic describes how to use Cloud Storage Gateway (CSG) SDK for Python to initiate a request.

Procedure

1. Install CSG SDK

```
from aliyunsdkcore import client
from aliyunsdkcore.profile import region_provider
from aliyunsdksgwhybrid.request.v20180511 import DescribeGatewaysRequest
```

2. Create an AcsClient instance.

```
client.region_provider.modify_point(product_name, region_id, endpoint)
clt = client.AcsClient(access_key, access_secret, region_id)
```

3. Create a request object.

```
request = DescribeGatewaysRequest.DescribeGatewaysRequest()
request.set_endpoint(endpoint)
request.set_content_type("application/json")
request.set_accept_format("JSON")
```

4. Initiate the request and process the response.

```
response = clt.do_action_with_exception(request)
print response
```

5.3.2.5. Handle errors

If the server or SDK encounters an error when you use CSG SDK for Python, it returns an error response. The error response contains an error code and an error message.

In most cases, you do not need to handle the errors returned by CSG SDK for Python. You only need to handle the errors returned by the server based on error messages.

- `ServerException` indicates a server error.
- `ClientException` indicates an SDK error.

For example, when the following error occurs, you can modify the AccessKey ID based on the error message to solve the problem:

```
aliyun_sdkcore.acs_exception.exceptions.ServerException: HTTP Status: 404 Error:InvalidAccessKeyId.NotFound Specified access key is not found.
```

If you want to handle the SDK errors returned by CSG SDK for Python, refer to the following example to write your code:

```
try:
    response = client.do_action_with_exception(request)
except ServerException as e:
    # You can add your own error handling logic here.
    # For example, display the error message.
    print e.get_http_status()
    print e.get_error_code()
    print e.get_error_msg()
```

5.4. API reference

5.4.1. List of operations by function

The following tables list the API operations that you can call in Cloud Storage Gateway (CSG).

Gateway

API	Description
CreateGateway	Creates a gateway.
DescribeGateways	Queries a list of gateways.
DescribeGateway	Queries the basic information of a gateway.
ModifyGateway	Modifies the basic information of a gateway.
DeleteGateway	Deletes a gateway.
UpgradeGateway	Upgrades a gateway.
RebootGateway	Restarts a gateway.
DescribeTasks	Queries the information of a task.

Share

API	Description
DeployCacheDisk	Creates a share.
DescribeGatewayFileShares	Obtains the configurations of a share for a gateway.
ModifyGatewayFileShare	Modifies a share.
DeleteGatewayFileShares	Deletes a share from a gateway.
CreateGatewaySMBUser	Creates an SMB user.
DescribeGatewaySMBUsers	Obtains a list of SMB users for a gateway.
DeleteGatewaySMBUser	Deletes an SMB user from a gateway.

Cache disk

API	Description
DescribeGatewayCaches	Queries a list of cache disks.
ExpandCacheDisk	Scales up a cache disk.
DeleteGatewayCacheDisk	Deletes a cache disk.

5.4.2. Common parameters

Common request parameters

Common parameters are required in all Cloud Storage Gateway (CSG) API requests.

Common request parameters

Parameter	Type	Required	Description
Format	String	No	The type of the returned value. Valid value: JSON.
Version	String	Yes	The version number of the API. Specify the version number in the YYYY-MM-DD format. Example: 2019-05-10.
AccessKeyId	String	Yes	The AccessKey ID provided to you by Apsara Stack.
AccessKeySecret	String	Yes	The AccessKey secret provided to you by Apsara Stack.

Parameter	Type	Required	Description
RegionId	String	Yes	The region ID provided to you by Apsara Stack.
Product	String	Yes	The name of the service. Set the value to sgw.

You can obtain the values of the `Product` and `Version` parameters in the Apsara Uni-manager Management Console. Perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Others > API Tool**.
3. In the left-side navigation pane, click **API Directory**.
4. In the **Operation Name** search box, enter the name of an operation to query its information.

Common response parameters

API responses use the HTTP response format:

- A 2xx status code indicates a successful call.
- A 4xx or 5xx status code indicates a failed call.

Responses are returned in the JSON format. Every response returns a unique RequestID regardless of whether the call is successful.

5.4.3. Gateway

5.4.3.1. CreateGateway

Creates a gateway.

When you call the operation, take note of the following information:

- A virtual private cloud (VPC) and a vSwitch are created in the region where you want to create a file gateway.
- An Elastic Compute Service (ECS) instance is available in the region where you want to create a file gateway. The ECS instance can be connected over the VPC.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateGateway	The operation that you want to perform. Set the value to CreateGateway.
Name	String	Yes	alex***	The name of the gateway. The name must be 1 to 60 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter.
Description	String	No	Test	The description of the gateway. The description must be 0 to 255 characters in length.
VSwitchId	String	No	vsw-bp1krhkgbx3nahkb9s***	• The vSwitch must reside in the same VPC as the ECS instance that you want to mount. • If no gateway is available in the zone where the vSwitch resides, create a vSwitch in another zone.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
GatewayId	String	gw-0001xv7je357zm9u6***	The ID of the gateway.
Message	String	successful	The description of the status code.
RequestId	String	8E69E1A1-9076-4C8C-8ADF-ACD253886E22	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

5.4.3.2. DescribeGateways

Queries the information of multiple gateways, including the ID and name.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeGateways	The operation that you want to perform. Set the value to DescribeGateways.
PageNumber	Integer	No	1	The number of the page to return. - Pages start from page 1. - Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. - Maximum value: 50 - Default value: 10

Response parameters

Parameter	Type	Example	Description
TotalCount	Integer	1	The total number of gateways.
RequestId	String	438A1BEF-9ED5-4B39-B11F-3FEF5F49D559	The ID of the request.
Message	String	successful	The description of the status code.
PageSize	Integer	10	The number of entries returned per page.
PageNumber	Integer	1	The page number of the returned page.
Gateways	Array		The list of gateways.
Gateway			
Status	String	Running	The status of the gateway. Valid values: - Initialized: The gateway is initialized. - Deploying: The gateway is being deployed. - Running: The gateway is running. - Starting: The gateway is being started. - Rebooting: The gateway is being restarted. - Deleting: The gateway is being deleted. - Failed: The gateway failed. - Upgrading: The gateway is being upgraded.
CreatedTime	Long	1577179604	The time at which the gateway was created. The timestamp. Unit: seconds.
InnerIp	String	172.16.0.2	The mount target of the gateway.
GatewayId	String	gw-0001b2odzvl3b1mhi***	The ID of the gateway.
GatewayClass	String	Standard	The specification of the gateway. Valid values: Standard
Name	String	fio-weekly***	The name of the gateway.
EcsInstanceId	String	gw-0001b2onkdzvl3b***	The ID of the Elastic Compute Service (ECS) instance.
Type	String	File	The type of the gateway.
VpcId	String	vpc-bp1p0usrgr5z9e7lwr***	The ID of the virtual private cloud (VPC).
ActivatedTime	Long	1577179769	The time at which the gateway was activated. The timestamp. Unit: seconds.

Parameter	Type	Example	Description
GatewayVersion	String	1.2.6	The version of the gateway.
LastErrorKey	String	internal.error	The key of the last error.
Location	String	Cloud	The location of the gateway. Valid values: Cloud: The gateway is deployed in Alibaba Cloud.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Success	Boolean	true	Indicates whether the request is successful.

5.4.3.3. DescribeGateway

Queries the information of a gateway, including the status and type.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeGateway	The operation that you want to perform. Set the value to DescribeGateway.
GatewayId	String	Yes	gw-000eg44nmxbfwbvq***	The ID of the gateway.

Response parameters

Parameter	Type	Example	Description
ActivatedTime	Long	1578986214	The time at which the gateway was activated. The timestamp. Unit: seconds.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
CreatedTime	Long	1532099461	The time at which the gateway was created. The timestamp. Unit: seconds.
EcsInstanceId	String	i-uf62406gy5vyao4y8***	The ID of the Elastic Compute Service (ECS) instance.
GatewayId	String	gw-0000ewknguw5o96vn***	The ID of the gateway.
GatewayVersion	String	1.0.39	The version of the gateway.
InnerIp	String	172.19.55.65	The mount target of the gateway.
Location	String	Cloud	The location of the gateway. Valid values: Cloud: The gateway is deployed in Alibaba Cloud.
Message	String	successful	The description of the status code.
Name	String	alex***	The name of the gateway.
RequestId	String	736C7733-FCED-4AFB-8407-C325F210CE03	The ID of the request.
Status	String	Initialized	The status of the gateway. Valid values: • Initialized: The gateway is initialized. • Deploying: The gateway is being deployed. • Running: The gateway is running. • Starting: The gateway is being started. • Rebooting: The gateway is being restarted. • Deleting: The gateway is being deleted. • Failed: The gateway failed. • Upgrading: The gateway is being upgraded.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request is successful.
Type	String	File	The type of the gateway.
VpcId	String	vpc-uf6j237kxmsw08auey***	The ID of the virtual private cloud (VPC).
GatewayClass	String	Standard	The specification of the gateway. Standard

5.4.3.4. ModifyGateway

Modifies the information of a gateway, including the name and description.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyGateway	The operation that you want to perform. Set the value to ModifyGateway.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.
Name	String	Yes	alex***	The new name of the gateway. The name must be 1 to 60 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter.
Description	String	No	Test	The new description of the gateway. The description must be 0 to 255 characters in length.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
RequestId	String	220E7407-DF77-428D-8955-B02CF8F82D07	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

5.4.3.5. DeleteGateway

Deletes a gateway.

You can delete a gateway that is only in the Initialized, Activated, Running, Unknown, Failed, or Stopped state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteGateway	The operation that you want to perform. Set the value to DeleteGateway.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.

Parameter	Type	Example	Description
RequestId	String	151BB31D-571D-4BB9-9A02-D4210E5BF5DB	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
TaskId	String	t-0000iwdggghaordhk***	The ID of the task.

5.4.3.6. UpgradeGateway

Upgrades a gateway.

When you call the operation, take note of the following information:

- You can upgrade a gateway that is only in the Running state.
- Services may be interrupted during the upgrade process.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpgradeGateway	The operation that you want to perform. Set the value to UpgradeGateway.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.

Response parameters

Parameter	Type	Example	Description
RequestId	String	CB300009-3850-4107-B291-59BDB4EDA183	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
TaskId	String	t-0001xv7je3580412***	The ID of the task.

5.4.3.7. RebootGateway

Restarts a gateway.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RebootGateway	The operation that you want to perform. Set the value to RebootGateway.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.

Response parameters

Parameter	Type	Example	Description
RequestId	String	CB300009-3850-4107-B291-59BDB4EDA183	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
TaskId	String	t-0001xv7je3580412***	The ID of the task.

5.4.3.8. DescribeTasks

Queries the information of a task, including the progress and name.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeTasks	The operation that you want to perform. Set the value to DescribeTasks.
TargetId	String	Yes	sync-0001xv757xl9sn548***	The ID of the gateway or synchronization group.
PageNumber	Integer	No	1	The number of the page to return. - Pages start from page 1. - Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. - Maximum value: 50 - Default value: 10
TaskId	String	No	t-0001xv7je357xl9n***	The ID of the task. If you leave the parameter empty, all the tasks of the gateway are enumerated.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
RequestId	String	8E9C6D43-4603-4F05-BE76-677A05EDFB7A	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Tasks	Array		The list of tasks.
SimpleTask			
CreatedTime	Long	1578367229	The time at which the task was created. The timestamp. Unit: seconds.
MessageKey	String	xxxxxxxxxx	The output that was returned when an error occurred.
Name	String	task.name.operate_fast_sync.delete	The name of the task.
Progress	Integer	100	The progress of the task. Valid values: 0 to 100.
StateCode	String	task.state.completed	The status code of the task. Valid values: - task.state.Initialized: The task is initialized. - task.state.Running: The task is running. - task.state.completed: The task is completed. - task.state.Failed: The task failed.
TaskId	String	t-0001xv7je357xl9n***	The ID of the task.
UpdatedTime	Long	1578367230	The time at which the task information was last modified. The timestamp. Unit: seconds.
TotalCount	Integer	1	The total number of tasks.

5.4.4. Shares

5.4.4.1. DeployCacheDisk

Creates a share.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeployCacheDisk	The operation that you want to perform. Set the value to DeployCacheDisk.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.
DiskCategory	String	Yes	cloud_efficiency	The type of the cache disk. Valid values: - cloud_efficiency - cloud_ssd
SizeInGB	Integer	Yes	40	The capacity of the cache disk.
CacheConfig	String	Yes	CacheConfig parameter description	The configurations of the share.

Response parameters

Parameter	Type	Example	Description
RequestId	String	CB300009-3850-4107-B291-59BDB4EDA183	The ID of the request.
Success	Boolean	true	If true is returned, the request is successful. If false is returned, the request failed.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
TaskId	String	t-0001xv7je3580412***	The ID of the task.

5.4.4.2. DescribeGatewayFileShares

Queries the configuration information of a share for a gateway.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeGatewayFileShares	The operation that you want to perform. Set the value to DescribeGatewayFileShares.
GatewayId	String	Yes	gw-000eg44nmxbfwbvq***	The ID of the gateway.
IndexId	String	No	NFSd-million	The ID of the share. If you leave the parameter empty, the configurations of all the shares for the gateway are enumerated.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
FileShares	Array		The list of shares.
FileShare			
Address	String	172.16.0.44	The IP address of the VPC.
BeLimit	Integer	0	The maximum upload speed. 0 indicates that the upload speed is not limited. Unit: bytes/s.

Parameter	Type	Example	Description
Browsable	Boolean	true	Indicates whether the share is browsable. Valid values: true and false. The value true indicates that the share is browsable. The value false indicates that the share is not browsable.
CacheMode	String	Cache	The data synchronization mode of the share. Valid values: - Cache: cache mode - Sync: replication mode
DirectIO	Boolean	false	Indicates whether direct I/O is enabled.
DiskId	String	d-bp1cvy8o2569jj31***	The ID of the cache disk.
Enabled	Boolean	true	Indicates whether the share is enabled. Valid values: true and false. The value true indicates that the share is enabled. The value false indicates that the share is disabled.
FeLimit	Integer	0	The maximum write speed. 0 indicates that the upload speed is not limited. Unit: bytes/s.
FileNumLimit	Long	10200202	The maximum number of files supported by the file system.
FsSizeLimit	Long	70368744177664	The capacity of the file system. Unit: bytes.
IgnoreDelete	Boolean	false	Indicates whether the ignore deletions feature is enabled.
InPlace	Boolean	false	Indicates whether the fragment optimization feature is enabled.
InRate	Long	0	The speed of accelerated caching. Unit: bytes/s.
IndexId	String	NFSale***	The ID of the share.
LagPeriod	Long	5	The upload latency. Unit: seconds.
LocalPath	String	/dev/vdb	The path of the cache disk.
Name	String	alex***	The name of the share.
OssBucketName	String	bucketA	The name of the Object Storage Service (OSS) bucket.
OssBucketSsl	Boolean	false	Indicates whether Secure Sockets Layer (SSL) was used to connect to the OSS bucket.
OssEndpoint	String	oss-cn-hangzhou-internal.aliyuncs.com	The endpoint of the OSS bucket.
OssHealth	String	BucketHealthy	The status of the OSS bucket. Valid values: - BucketHealthy: The OSS bucket is connected as expected. - BucketAccessDenied: The OSS bucket cannot be accessed due to insufficient permissions. - BucketMiscFailure: The OSS bucket cannot be accessed due to an error. - BucketNetworkFailure: The OSS bucket cannot be accessed due to network failure. - BucketNotExist: The OSS bucket does not exist. - Null: Unknown.
OssUsed	Long	0	The amount of data stored in the cloud. Unit: bytes.
OutRate	Long	0	The upload speed. Unit: bytes/s.
PathPrefix	String	testprefix	The directory prefix of the OSS bucket.
PollingInterval	Integer	0	The interval between two consecutive reverse synchronization tasks.

Parameter	Type	Example	Description
Protocol	String	SMB	The type of the share. Valid values: • SMB • NFS
RemainingMetaSpace	Long	21308227584	The capacity of the metadata area. Unit: bytes.
RemoteSync	Boolean	false	Indicates whether the reverse synchronization feature is enabled.
RoClientList	String	user1	The IP addresses or CIDR blocks that you allow to only read data from the gateway. The parameter is returned only when the Protocol parameter is set to NFS.
RoUserList	String	user2	The IP addresses or CIDR blocks that you allow to only read data from the gateway. The parameter is returned only when the Protocol parameter is set to SMB.
RwClientList	String	user3	The IP addresses or CIDR blocks that you allow to read data from and write data to the gateway. The parameter is returned only when the Protocol parameter is set to NFS.
RwUserList	String	user4	The IP addresses or CIDR blocks that you allow to read data from and write data to the gateway. The parameter is returned only when the Protocol parameter is set to SMB.
Size	Long	85899345920	The capacity of the cache disk. Unit: bytes.
Squash	String	root_squash	The user mapping.
State	String	clean	The synchronization status of data. Valid values: • clean: The synchronization is complete. • dirty: The synchronization is not complete.
TotalDownload	Long	0	The amount of data downloaded from the OSS bucket to the gateway. Unit: bytes.
TotalUpload	Long	0	The amount of data uploaded from the OSS bucket to the gateway. Unit: bytes.
Used	Long	0	The consumed capacity of the cache disk. Unit: bytes.
Message	String	successful	The description of the status code.
RequestId	String	9A3C3C32-3F37-4658-9D38-945D0799CA0E	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

5.4.4.3. ModifyGatewayFileShare

Modifies a share.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyGatewayFileShare	The operation that you want to perform. Set the value to ModifyGatewayFileShare.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.
IndexId	String	Yes	NFStest_nfs	The ID of the share. You can call the DescribeGatewayFileShares operation to query the share ID.
CacheConfig	String	Yes	CacheConfig parameter description	The configurations of the share.

Response parameters

Parameter	Type	Example	Description
RequestId	String	CB300009-3850-4107-B291-59BDB4EDA183	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
TaskId	String	t-0001xv7je3580412***	The ID of the task.

5.4.4.4. DeleteGatewayFileShares

Deletes a share from a gateway.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteGatewayFileShares	The operation that you want to perform. Set the value to DeleteGatewayFileShares.
GatewayId	String	Yes	gw-000eg44nmxbfwbvq2**	The ID of the gateway.
IndexId	String	Yes	NFSalex***	The ID of the share. You can call the DescribeGatewayFileShares operation to query the share ID.
Force	Boolean	No	false	Specifies whether to forcibly delete the share if data synchronization is not complete. Valid values: - true: The share is forcibly deleted. Proceed with caution. - false: The share is not forcibly deleted. Default value: false

Response parameters

Parameter	Type	Example	Description
RequestId	String	t-000czaedx5koz578vk**	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
TaskId	String	t-000c6hshwukcfn3k**	The ID of the task.

5.4.4.5. CreateGatewaySMBUser

Adds an SMB user and password to a gateway.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateGatewaySMBUser	The operation that you want to perform. Set the value to CreateGatewaySMBUser.
GatewayId	String	Yes	gw-000eg44nmxbfwbvq2**	The ID of the gateway.

Parameter	Type	Required	Example	Description
Password	String	Yes	123***	The password of the SMB user. The password cannot be empty. It can contain letters, digits, and the following characters: !, @, #, \$, %, ^, &, and *.
Username	String	Yes	alex	The name of the SMB user. The name must be 1 to 32 characters in length and can contain letters, digits, hyphens (-), and underscores (_).

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
RequestId	String	F8B59F29-453D-49BF-8673-EEB8F9F2D5C6	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
TaskId	String	t-000eg44nmxbsh3qk***	The ID of the task.

5.4.4.6. DescribeGatewaySMBUsers

Queries the SMB users of a gateway.

Request parameters

Parameter	Type	Required	Example	Description
GatewayId	String	Yes	gw-000eg44nmxbfwbv***	The ID of the gateway.
PageNumber	Integer	No	1	The number of the page to return. - Pages start from page 1. - Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. - Maximum value: 50 - Default value: 10

Response parameters

Parameter	Type	Example	Description
RequestId	String	4C02F482-FC2E-45BF-88C5-8D40134D11**	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
TotalCount	Integer	1	The total number of SMB users.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
Users	Array		The list of SMB users.

Parameter	Type	Example	Description
Username	String	alex***	The name of the SMB user.

5.4.4.7. DeleteGatewaySMBUser

Deletes an SMB user from a gateway.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteGatewaySMBUser	The operation that you want to perform. Set the value to DeleteGatewaySMBUser.
GatewayId	String	Yes	gw-000c6hshwukbrypj***	The ID of the gateway.
Username	String	Yes	username	The name of the SMB user. The name must be 1 to 32 characters in length and can contain letters, digits, hyphens (-), and underscores (_).

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
RequestId	String	89398CFB-4EB6-4C7E-BB3C-EF213AC8FA50	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
TaskId	String	t-000c6hshwukcfn3k**	The ID of the task.

5.4.4.8. CacheConfig parameter description

Request parameters

Parameter	Type	Required	Example	Description
Name	String	Yes	myshare	The name of the share.
Protocol	String	Yes	NFS	The protocol of the share. Valid values: - NFS - SMB
Enabled	Boolean	Yes	true	Specifies whether to enable the share. Valid values: - true - false
Squash	String	No	root_squash	The parameter takes effect only when the Protocol parameter is set to NFS. Valid values: - none - root_squash - all_squash - all_anonymous

Parameter	Type	Required	Example	Description
LagPeriod	Integer	Yes	5	The synchronization latency. Unit: seconds. Note - Valid values: 5 to 120. - Recommended value: 5.
RemoteSync	Boolean	No	true	Specifies whether to enable the reverse synchronization feature. Valid values: - true - false
PollingInterval	Integer	No	60	The interval between two consecutive reverse synchronization tasks. Unit: seconds. Note Valid values: 60 to 3600.
CacheMode	String	Yes	Cache	The data synchronization mode of the data volume.
BucketEndpoint	String	Yes	oss.aliyun.com	The domain name of the Object Storage Service (OSS) bucket.
BucketName	String	Yes	mybucket	The name of the OSS bucket.
IgnoreDelete	Boolean	No	false	Specifies whether to enable the ignore deletions feature. Valid values: - true - false
PathPrefix	String	No	/sub/folder	This parameter is required if you use a subdirectory of the OSS bucket.
LocalFilePath	String	No	/dev/sdb	The path of the cache disk used by the share.

5.4.5. Cache disks

5.4.5.1. DescribeGatewayCaches

Queries the information of a cache disk, including the capacity and type.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeGatewayCaches	The operation that you want to perform. Set the value to DescribeGatewayCaches.
GatewayId	String	Yes	gw-000eg44nmxbfwbvq***	The ID of the gateway.

Response parameters

Parameter	Type	Example	Description
RequestId	String	40797B4D-09D9-482E-B163-CCB568BADE35	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.

Parameter	Type	Example	Description
Caches	Array		The list of cache disks.
SizeInGB	Long	101	The capacity of the cache disk. Unit: GB.
Iops	Long	2605	The IOPS.
CacheId	String	d-bp1dcnq8zldree6ok***	The ID of the cache disk.
CacheType	String	cloud_efficiency	The type of the cache disk. Valid values: - cloud_efficiency: ultra disk - cloud_ssd: standard SSD
IsUsed	Boolean	false	Indicates whether the cache disk was used.
LocalFilePath	String	/dev/vdb	The path of the cache disk.

5.4.5.2. ExpandCacheDisk

Scales up a cache disk for a gateway.

When you call the operation, take note of the following information:

- Each cache disk of a file gateway can be scaled up from 40 GB to 4 TB.
- You cannot scale down cache disks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ExpandCacheDisk	The operation that you want to perform. Set the value to ExpandCacheDisk.
GatewayId	String	Yes	gw-000eg44nmxbfwbvq***	The ID of the gateway.
LocalFilePath	String	Yes	/dev/vdb	The path of the cache disk. You can call the DescribeGatewayCaches operation to query the path of the cache disk.
NewSizeInGB	Integer	Yes	101	The new capacity of the cache disk. Unit: GB. Take note of the following information: - The new capacity must be greater than the original capacity. This is because you cannot scale down a cache disk. - The cache disk can be scaled up from 40 GB to 4 TB.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
RequestId	String	F8B59F29-453D-49BF-8673-EEB8F9F2D5C6	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
TaskId	String	t-000eg44nmxbsh3qk3***	The ID of the task.

5.4.5.3. DeleteGatewayCacheDisk

Deletes a cache disk from a gateway.

When you call the operation, make sure that no share is associated with the cache disk.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteGatewayCacheDisk	The operation that you want to perform. Set the value to DeleteGatewayCacheDisk.
CacheId	String	Yes	d-bp1dcnq8zldree6ok***	The ID of the cache disk. You can call the DescribeGatewayCaches operation to query the ID of the cache disk.
GatewayId	String	Yes	gw-0001xv7je3580031y***	The ID of the gateway.
LocalFilePath	String	Yes	/dev/vdb	The path of the cache disk. You can call the DescribeGatewayCaches operation to query the path of the cache disk.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. If the request is successful, 200 is returned.
Message	String	successful	The description of the status code.
RequestId	String	EEFC9927-B097-446D-8FDA-F848C8B2C9E3	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.
TaskId	String	t-0001xv7je357xv4s***	The ID of the task.