ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Agility SE

Developer Guide

Product Version: 2006, Internal: V3.3.0

Document Version: 20201019

Alibaba Cloud

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**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.ASCM console 31
1.1. ASAPI Developer Guide 31
1.1.1. Overview 31
1.1.2. Log on to the API Tool console 32
1.1.3. View the information about API operations 32
1.1.4. Obtain the endpoint of Asapi 33
1.1.5. Obtain the SDK package for Asapi 35
1.1.6. Obtain an AccessKey pair 35
1.1.7. Query the region ID 36
1.1.8. SDK for Java 36
1.1.9. SDK for Node.js 38
1.1.10. Use the POP SDK to call ASAPI 40
1.2. API reference 42
1.2.1. Common parameters 42
1.2.2. Make API requests 42
1.2.3. ASCM authentication 43
1.2.4. Enterprise 44
1.2.4.1. Organizations 44
1.2.4.1.1. UpdateOrganization 44
1.2.4.1.2. UpdateSupportRegionsForOrg 45
1.2.4.1.3. GetOrganization 46
1.2.4.1.4. GetOrganizationList 48
1.2.4.1.5. RemoveOrganization 50
1.2.4.1.6. ListResourceGroupByOrgId 51
1.2.4.1.7. CreateOrganization 54
1.2.4.2. Resource sets 55

1.2.4.2.1. BindAscmUserAndResourceGroup 56
1.2.4.2.2. CreateResourceGroup 56
1.2.4.2.3. UnbindAscmUserAndResourceGroup 58
1.2.4.2.4. UpdateResourceGroup 59
1.2.4.2.5. RemoveResourceGroup 60
1.2.4.2.6. ListResourceGroup 61
1.2.4.2.7. ListAscmUsersInsideResourceGroup 69
1.2.4.3. Roles 79
1.2.4.3.1. SetDefaultRole 79
1.2.4.3.2. ListRoles 80
1.2.4.3.3. ListActiveRoles 88
1.2.4.4. Users 97
1.2.4.4.1. AddUser 97
1.2.4.4.2. AddRoleToUser 105
1.2.4.4.3. UpdatePassword 106
1.2.4.4.4. ResetPasswordByLoginName 107
1.2.4.4.5. GetLoginPolicyByLoginName 107
1.2.4.4.6. RemoveUserByLoginName 110
1.2.4.4.7. GetUserInfo 111
1.2.4.4.8. ListUsers 119
1.2.4.4.9. ListDeletedUsers 128
1.2.4.4.10. ListResourceGroupByAscmUserId 130
1.2.4.4.11. ModifyUserInformation 138
1.2.4.4.12. RecoverUserByLoginName 147
1.2.4.4.13. RemoveRoleFromUser 151
1.2.4.5. Logon policies 152
1.2.4.5.1. AddLoginPolicy 153
1.2.4.5.2. CheckLoginPolicy 155

1.2.4.5.3. ListLoginPolicies 157
1.2.4.5.4. ModifyLoginPolicy 161
1.2.4.5.5. GetLoginPolicyById 164
1.2.5. Cloud services 167
1.2.5.1. DescribeModifiableInstanceType 167
1.2.5.2. DescribeSeriesIdFamilies 168
1.2.5.3. GetEnvProducts 170
1.2.5.4. GroupCommonSpec 175
1.2.5.5. GetRegionsByProduct 176
1.2.5.6. DescribeInstanceTypeFamilies 177
1.2.5.7. GetClustersByProduct 178
1.2.5.8. GetJumpUrlByProduct 180
1.2.6. Configurations 181
1.2.6.1. RemoveLoginPolicyById 181
1.2.6.2. GetPasswordPolicy 182
1.2.6.3. ResetPasswordPolicy 184
1.2.6.4. SetPasswordPolicy 186
1.2.7. Monitoring 189
1.2.7.1. Major metrics of Apsara Stack services 189
1.2.7.2. Monitoring data on time series metrics of cloud s... 192
1.2.7.2.1. DescribeProjectMeta 192
1.2.7.2.2. DescribeMetricLast 197
1.2.7.2.3. DescribeMetricList 205
1.2.7.2.4. DescribeMetricMetaList 214
1.2.7.3. Alert rules for time series metrics 229
1.2.7.3.1. DescribeMetricRuleList 229
1.2.7.3.2. PutResourceMetricRule 242
1.2.7.3.3. DescribeMetricRuleCount 249

1.2.7.3.4. DeleteMetricRules 251
1.2.7.3.5. EnableMetricRules 253
1.2.7.3.6. DisableMetricRules 254
1.2.7.3.7. DescribeAlertingMetricRuleResources 256
1.2.7.3.8. DeleteMetricRuleResources 260
2.Object Storage Service (OSS) 263
2.1. Java SDK 263
2.1.1. Preface 263
2.1.2. Installation 264
2.1.3. Initialization 265
2.1.4. Quick start 271
2.1.5. Buckets 275
2.1.5.1. Create buckets 275
2.1.5.2. List buckets 276
2.1.5.3. Determine whether buckets exist 278
2.1.5.4. Manage bucket ACLs 279
2.1.5.5. Delete buckets 280
2.1.5.6. Manage lifecycle rules 281
2.1.5.7. Configure hotlink protection 285
2.1.5.8. Configure CORS 287
2.1.5.9. Configure logging 290
2.1.5.10. Configure static website hosting 292
2.1.6. Upload objects 294
2.1.6.1. Overview 294
2.1.6.2. Simple upload 294
2.1.6.3. Form upload 297
2.1.6.4. Append upload 302
2.1.6.5. Resumable upload 304

2.1.6.6. Multipart upload 305
2.1.6.7. Use a progress bar 317
2.1.6.8. Upload callback 319
2.1.7. Download objects 320
2.1.7.1. Overview 320
2.1.7.2. Streaming download 320
2.1.7.3. Download to local files 321
2.1.7.4. Range download 322
2.1.7.5. Resumable download 326
2.1.7.6. Conditional download 328
2.1.7.7. Use a progress bar 329
2.1.8. Manage objects 331
2.1.8.1. Overview 331
2.1.8.2. Determine whether an object exists 332
2.1.8.3. Manage object ACLs 332
2.1.8.4. Manage object metadata 334
2.1.8.5. List objects 338
2.1.8.6. Query objects 355
2.1.8.7. Delete objects 358
2.1.8.8. Copy objects 362
2.1.9. Authorized access 368
2.1.10. IMG 375
2.1.11. Handle exceptions 379
2.1.12. FAQ 383
2.2. Python-SDK 392
2.2.1. Preface 393
2.2.2. Installation 394
2.2.3. Quick start 396

2.2.4. Initialization 399
2.2.5. Buckets 400
2.2.5.1. Create buckets 400
2.2.5.2. List buckets 401
2.2.5.3. Determine whether buckets exist 401
2.2.5.4. Manage bucket ACLs 402
2.2.5.5. Delete buckets 403
2.2.5.6. Manage lifecycle rules 404
2.2.5.7. Static website hosting 408
2.2.5.8. Configure CORS 409
2.2.5.9. Configure hotlink protection 411
2.2.5.10. Configure logging 412
2.2.6. Upload objects 414
2.2.6.1. Overview 414
2.2.6.2. Simple upload 414
2.2.6.3. Append upload 418
2.2.6.4. Resumable upload 418
2.2.6.5. Multipart upload 420
2.2.6.6. Use progress bars 424
2.2.6.7. Upload callback 425
2.2.7. Download objects 426
2.2.7.1. Overview 426
2.2.7.2. Streaming download 427
2.2.7.3. Download to local files 428
2.2.7.4. Range download 428
2.2.7.5. Resumable download 430
2.2.7.6. Use progress bars 432
2.2.8. Manage objects 433

2.2.8.1. Overview 433
2.2.8.2. Determine whether an object exists 433
2.2.8.3. Manage object ACLs 434
2.2.8.4. Manage object metadata 435
2.2.8.5. List objects 438
2.2.8.6. Query objects 441
2.2.8.7. Delete objects 447
2.2.8.8. Copy an object 449
2.2.9. Authorized access 450
2.2.10. Chinese characters and time 453
2.2.11. IMG 455
2.2.12. Handle exceptions 461
2.3. PHP-SDK 464
2.3.1. Preface 464
2.3.2. Installation 465
2.3.3. Initialization 466
2.3.4. Quick start 470
2.3.5. Buckets 475
2.3.5.1. Create buckets 475
2.3.5.2. List buckets 476
2.3.5.3. Determine whether buckets exist 477
2.3.5.4. Manage bucket ACLs 478
2.3.5.5. Delete buckets 481
2.3.5.6. Manage lifecycle rules 482
2.3.5.7. Configure hotlink protection 486
2.3.5.8. Configure logging 489
2.3.5.9. Configure static website hosting 492
2.3.5.10. Configure CORS 495

2.3.6. Upload objects 498
2.3.6.1. Overview 498
2.3.6.2. Simple upload 499
2.3.6.3. Append upload 501
2.3.6.4. Multipart upload 504
2.3.6.5. Upload callback 513
2.3.7. Download objects 517
2.3.7.1. Overview 517
2.3.7.2. Download to local files 517
2.3.7.3. Download objects to local memory 519
2.3.7.4. Range download 519
2.3.7.5. Conditional download 520
2.3.8. Manage objects 522
2.3.8.1. Overview 522
2.3.8.2. Determine whether objects exist 523
2.3.8.3. Manage object ACLs 524
2.3.8.4. Manage object metadata 527
2.3.8.5. List objects 530
2.3.8.6. Delete objects 534
2.3.8.7. Copy an object 536
2.3.9. Authorized access 539
2.3.10. IMG 544
2.3.11. Error handling 553
2.4. C-SDK 556
2.4.1. Preface 556
2.4.2. Installation 557
2.4.3. Initialization 561
2.4.4. Quick start 565

2.4.5. Buckets 575
2.4.5.1. Create buckets 575
2.4.5.2. List buckets 576
2.4.5.3. Manage bucket ACLs 578
2.4.5.4. Delete buckets 582
2.4.5.5. Manage lifecycle rules 583
2.4.5.6. CORS 590
2.4.5.7. Configure logging 595
2.4.5.8. Configure hotlink protection 600
2.4.5.9. Configure static website hosting 605
2.4.6. Upload objects 610
2.4.6.1. Overview 610
2.4.6.2. Simple upload 610
2.4.6.3. Append upload 616
2.4.6.4. Resumable upload 620
2.4.6.5. Multipart upload 623
2.4.6.6. Use progress bars 629
2.4.6.7. Upload callback 631
2.4.7. Download objects 633
2.4.7.1. Overview 633
2.4.7.2. Download to local files 633
2.4.7.3. Download to local memory 635
2.4.7.4. Range download 637
2.4.7.5. Resumable download 640
2.4.7.6. Use progress bars 642
2.4.8. Manage objects 644
2.4.8.1. Overview 644
2.4.8.2. Manage object ACLs 644

2.4.8.3. Manage object metadata 647
2.4.8.4. List objects 649
2.4.8.5. Delete objects 669
2.4.8.6. Copy objects 675
2.4.9. Authorized access 681
2.4.10. IMG 687
2.4.11. Error handling 698
2.5. Go-SDK 701
2.5.1. Preface 701
2.5.2. Installation 702
2.5.3. Initialization 703
2.5.4. Quick start 706
2.5.5. Buckets 713
2.5.5.1. Create buckets 713
2.5.5.2. List buckets 713
2.5.5.3. Manage bucket ACLs 717
2.5.5.4. Delete buckets 719
2.5.5.5. Manage lifecycle rules 720
2.5.5.6. Hotlink protection 724
2.5.5.7. Configure logging 727
2.5.5.8. Static website hosting 730
2.5.5.9. CORS 733
2.5.6. Upload objects 736
2.5.6.1. Overview 736
2.5.6.2. Simple upload 737
2.5.6.3. Append upload 741
2.5.6.4. Resumable upload 743
2.5.6.5. Multipart upload 745

2.5.6.6. Use a progress bar 754
2.5.7. Download objects 756
2.5.7.1. Overview 756
2.5.7.2. Streaming download 756
2.5.7.3. Range download 760
2.5.7.4. Download to local files 763
2.5.7.5. Resumable download 764
2.5.7.6. Conditional download 766
2.5.7.7. Use a progress bar 768
2.5.8. Manage objects 770
2.5.8.1. Overview 770
2.5.8.2. Determine whether objects exist 770
2.5.8.3. Manage object ACLs 771
2.5.8.4. Manage object metadata 774
2.5.8.5. List objects 778
2.5.8.6. Delete objects 792
2.5.8.7. Copy objects 796
2.5.9. Authorized access 802
2.5.10. IMG 806
2.5.11. Error handling 816
2.6. Overview 822
2.7. List of operations by function 823
2.8. Access control 825
2.8.1. Authenticate users by using signatures 825
2.8.2. Add signatures to the Authorization header 826
2.8.3. Generate a signed URL 833
2.8.4. Authorize temporary access 835
2.8.5. Bucket ACL 836

2.9. Common HTTP headers 837
2.10. Service-relevant operations 838
2.10.1. GetService(ListBuckets) 838
2.11. Bucket-relevant operations 844
2.11.1. PutBucket 844
2.11.2. PutBucketACL 845
2.11.3. PutBucketLogging 847
2.11.4. PutBucketWebsite 853
2.11.5. PutBucketReferer 855
2.11.6. PutBucketLifecycle 858
2.11.7. GetBucket 863
2.11.8. GetBucketAcl 871
2.11.9. GetBucketLocation 873
2.11.10. GetBucketInfo 874
2.11.11. GetBucketLogging 878
2.11.12. GetBucketWebsite 880
2.11.13. GetBucketReferer 883
2.11.14. GetBucketLifecycle 885
2.11.15. DeleteBucket 887
2.11.16. DeleteBucketLogging 888
2.11.17. DeleteBucketWebsite 889
2.11.18. DeleteBucketLifecycle 890
2.12. Object-relevant operations 891
2.12.1. PutObject 891
2.12.2. GetObject 894
2.12.3. CopyObject 899
2.12.4. AppendObject 902
2.12.5. DeleteObject 906

2.12.6. DeleteMultipleObjects 907
2.12.7. HeadObject 913
2.12.8. GetObjectMeta 915
2.12.9. PutObjectACL 916
2.12.10. GetObjectACL 918
2.12.11. PostObject 920
2.12.12. Callback 930
2.12.13. SelectObject 943
2.13. Multipart upload-relevant operations 974
2.13.1. Overview 974
2.13.2. InitiateMultipartUpload 974
2.13.3. UploadPart 977
2.13.4. UploadPartCopy 979
2.13.5. CompleteMultipartUpload 982
2.13.6. AbortMultipartUpload 986
2.13.7. ListMultipartUploads 987
2.13.8. ListParts 992
2.14. CORS 997
2.14.1. Overview 997
2.14.2. PutBucketcors 998
2.14.3. GetBucketcors 1001
2.14.4. DeleteBucketcors 1004
2.14.5. OptionObject 1005
2.15. Error responses 1007
2.16. Obtain an endpoint 1013
2.17. Obtain an AccessKey pair 1014
3.ApsaraDB for RDS 1016
3.1. Use ASAPI to call Apsara Stack SDKs 1016

3.1.1. Overview 1016
3.1.2. Log on to the API Tool console 1017
3.1.3. View the information about API operations 1017
3.1.4. Obtain the endpoint of ASAPI 1018
3.1.5. Obtain the core package for ASAPI 1020
3.1.6. Obtain an AccessKey pair 1020
3.1.7. Query the region ID 1021
3.1.8. Call SDK for Java 1021
3.1.9. Call SDK for Node.js 1023
3.1.10. (Optional) Use POP SDK to call ASAPI 1025
3.2. API reference 1026
3.2.1. List of operations by function 1026
3.2.2. Request method 1029
3.2.2.1. Request structure 1029
3.2.2.2. Common parameters 1030
3.2.2.2.1. Common request parameters 1030
3.2.2.2.2. Common response parameters 1031
3.2.2.3. Response 1031
3.2.2.3.1. Success responses 1031
3.2.2.3.2. Sample error responses 1032
3.2.2.4. Request signatures 1032
3.2.3. Instance management 1035
3.2.3.1. CreateDBInstance 1035
3.2.3.2. DeleteDBInstance 1037
3.2.3.3. RestartDBInstance 1038
3.2.3.4. DescribeDBInstanceAttribute 1039
3.2.3.5. DescribeDBInstances 1044
3.2.3.6. ModifyDBInstanceSpec 1052

3.2.3.7. DescribeDBInstanceNetInfo 1055
3.2.3.8. DescribeDBInstanceIPArrayList 1056
3.2.3.9. ModifySecurityIps 1057
3.2.3.10. DescribeRegions 1058
3.2.3.11. DescribeDBInstanceHAConfig 1061
3.2.3.12. PurgeDBInstanceLog 1062
3.2.3.13. ModifyDBInstanceDescription 1063
3.2.3.14. ModifyDBInstanceMaintainTime 1064
3.2.3.15. ModifyDBInstanceHAConfig 1066
3.2.3.16. SwitchDBInstanceHA 1066
3.2.3.17. CreateReadOnlyDBInstance 1067
3.2.3.18. DescribeDBInstanceSSL 1069
3.2.3.19. ModifyDBInstanceSSL 1070
3.2.4. Database management 1071
3.2.4.1. CreateDatabase 1071
3.2.4.2. DeleteDatabase 1072
3.2.4.3. DescribeDatabases 1073
3.2.4.4. ModifyDBDescription 1077
3.2.5. Account management 1078
3.2.5.1. CreateAccount 1078
3.2.5.2. DeleteAccount 1080
3.2.5.3. DescribeAccounts 1081
3.2.5.4. GrantAccountPrivilege 1083
3.2.5.5. RevokeAccountPrivilege 1085
3.2.5.6. ModifyAccountDescription 1086
3.2.5.7. ResetAccountPassword 1087
3.2.5.8. ResetAccount 1089
3.2.6. Backup and restoration 1089

3.2.6.1. CreateBackup 1089
3.2.6.2. CloneDBInstance 1091
3.2.6.3. DescribeBackups 1093
3.2.6.4. DescribeBackupPolicy 1094
3.2.6.5. ModifyBackupPolicy 1095
3.2.6.6. RestoreDBInstance 1097
3.2.6.7. DeleteBackup 1098
3.2.7. Monitoring management 1099
3.2.7.1. DescribeResourceUsage 1099
3.2.7.2. DescribeDBInstancePerformance 1100
3.2.7.3. DescribeDBInstanceMonitor 1103
3.2.7.4. ModifyDBInstanceMonitor 1103
3.2.8. Parameter management 1104
3.2.8.1. DescribeParameterTemplates 1104
3.2.8.2. DescribeParameters 1106
3.2.8.3. ModifyParameter 1109
3.2.9. Log management 1111
3.2.9.1. DescribeSlowLogs 1111
3.2.9.2. DescribeSlowLogRecords 1115
3.2.9.3. DescribeErrorLogs 1119
3.2.9.4. DescribeBinlogFiles 1121
3.2.9.5. DescribeSQLCollectorPolicy 1125
3.2.9.6. ModifySQLCollectorPolicy 1125
3.2.9.7. DescribeSQLLogRecords 1126
3.2.9.8. DescribeSQLLogFiles 1129
3.2.10. Data migration 1130
3.2.10.1. ImportDatabaseBetweenInstances 1130
3.2.10.2. CancelImport 1132

3.2.11. Performance optimization 1133
3.2.11.1. DescribeSQLLogReports 1133
3.2.12. RAM authorization 1135
3.2.13. Appendix 1139
3.2.13.1. Client error codes 1139
3.2.13.1.1. Common errors 1139
3.2.13.1.2. Specific errors 1140
3.2.13.2. Error codes on the server 1146
3.2.13.3. Instance states 1146
3.2.13.4. Performance parameters 1147
3.2.13.5. Forbidden keywords 1148
3.2.13.6. Instance types 1149
4.Data Transmission Service (DTS) 1154
4.1. API reference 1154
4.1.1. Call OpenAPI 1154
4.1.1.1. List of operations by function 1154
4.1.2. Data migration APIs 1155
4.1.2.1. ConfigureMigrationJob 1155
4.1.2.1.1. Overview 1155
4.1.2.1.2. Request parameters 1155
4.1.2.1.3. MigrationObject parameter definitions 1159
4.1.2.1.4. Response parameters 1161
4.1.2.1.5. Example 1161
4.1.2.2. CreateMigrationJob 1162
4.1.2.2.1. Overview 1162
4.1.2.2.2. Request parameters 1162
4.1.2.2.3. Response parameters 1163
4.1.2.2.4. Examples 1163

4.1.2.3. DeletetMigrationJob 1164
4.1.2.3.1. Overview 1164
4.1.2.3.2. Request parameters 1164
4.1.2.3.3. Response parameters 1164
4.1.2.3.4. Examples 1164
4.1.2.4. DescribeMigrationJobDetail 1165
4.1.2.4.1. Overview 1165
4.1.2.4.2. Request parameters 1165
4.1.2.4.3. Response parameters 1166
4.1.2.4.4. StructureInitializationDetail parameter definitio... 1167
4.1.2.4.5. DataInitializationDetail parameter definitions 1167
4.1.2.4.6. DataSynchronizationDetail parameter definitio... 1168
4.1.2.4.7. Examples 1169
4.1.2.5. DescribeMigrationJobStatus 1170
4.1.2.5.1. Overview 1170
4.1.2.5.2. Request parameters 1170
4.1.2.5.3. Response parameters 1170
4.1.2.5.4. Endpoint parameter definitions 1172
4.1.2.5.5. MigrationMode parameter definitions 1172
4.1.2.5.6. PrecheckStatus parameter definitions 1173
4.1.2.5.7. CheckItem parameter definitions 1173
4.1.2.5.8. StructureInitializationStatus parameter definitio... 1173
4.1.2.5.9. DataInitializationStatus parameter definitions 1174
4.1.2.5.10. DataSynchronizationStatus parameter definiti... 1174
4.1.2.5.11. Examples 1175
4.1.2.6. StartMigrationJob 1178
4.1.2.6.1. Overview 1178
4.1.2.6.2. Request parameters 1178

4.1.2.6.3. Response parameters 1178
4.1.2.6.4. Examples 1179
4.1.2.7. StopMigrationJob 1179
4.1.2.7.1. Overview 1179
4.1.2.7.2. Request parameters 1179
4.1.2.7.3. Response parameters 1179
4.1.2.7.4. Examples 1180
4.1.2.8. SuspendMigrationJob 1180
4.1.2.8.1. Overview 1180
4.1.2.8.2. Request parameters 1180
4.1.2.8.3. Response parameters 1180
4.1.2.8.4. Examples 1181
4.1.3. Data synchronization APIs 1181
4.1.3.1. ConfigureSynchronizationJob 1181
4.1.3.1.1. Overview 1181
4.1.3.1.2. Request parameters 1181
4.1.3.1.3. Response parameters 1182
4.1.3.1.4. Example 1183
4.1.3.2. CreateSynchronizationJob 1183
4.1.3.2.1. Overview 1183
4.1.3.2.2. Request parameters 1183
4.1.3.2.3. Response parameters 1186
4.1.3.2.4. Examples 1186
4.1.3.3. DeleteSynchronizationJob 1187
4.1.3.3.1. Overview 1187
4.1.3.3.2. Request parameters 1187
4.1.3.3.3. Response parameters 1188
4.1.3.3.4. Examples 1188

4.1.3.4. DescribeDataLoadStatus 1188
4.1.3.4.1. Overview 1188
4.1.3.4.2. Request parameters 1188
4.1.3.4.3. Response parameters 1189
4.1.3.4.4. Examples 1189
4.1.3.5. DescribeSynchronizationJobStatus 1191
4.1.3.5.1. Overview 1191
4.1.3.5.2. Request parameters 1191
4.1.3.5.3. Response parameters 1191
4.1.3.5.4. PrecheckStatus parameter definitions 1193
4.1.3.5.5. CheckItem parameter definitions 1193
4.1.3.5.6. StructureInitializationStatus parameter definitio... 1193
4.1.3.5.7. DataInitializationStatus parameter definitions 1194
4.1.3.5.8. DataSynchronizationStatus parameter definitio... 1194
4.1.3.5.9. Performance parameter definitions 1195
4.1.3.5.10. Examples 1195
4.1.3.6. DescribeSynchronizationJobs 1198
4.1.3.6.1. Overview 1198
4.1.3.6.2. Request parameters 1198
4.1.3.6.3. Response parameters 1198
4.1.3.6.4. SynchronizationInstance parameter definitions 1199
4.1.3.6.5. PrecheckStatus parameter definitions 1200
4.1.3.6.6. CheckItem parameter definitions 1201
4.1.3.6.7. StructureInitializationStatus parameter definitio... 1201
4.1.3.6.8. DataInitializationStatus parameter definitions 1202
4.1.3.6.9. DataSynchronizationStatus parameter definitio... 1202
4.1.3.6.10. Performance parameter definitions 1203
4.1.3.6.11. Examples 1203

4.1.3.7. DescribeObjectModifyStatus 1205
4.1.3.7.1. Overview 1205
4.1.3.7.2. Request parameters 1205
4.1.3.7.3. Response parameters 1206
4.1.3.7.4. PrecheckStatus parameter definitions 1206
4.1.3.7.5. CheckItem parameter definitions 1207
4.1.3.7.6. StructureInitializationStatus parameter definitio... 1207
4.1.3.7.7. DataInitializationStatus parameter definitions 1208
4.1.3.7.8. DataSynchronizationStatus parameter definitio... 1208
4.1.3.7.9. Examples 1208
4.1.3.8. ModifySynchronizationObject 1210
4.1.3.8.1. Overview 1210
4.1.3.8.2. Request parameters 1210
4.1.3.8.3. Response parameters 1211
4.1.3.8.4. Examples 1211
4.1.3.9. StartSynchronizationJob 1211
4.1.3.9.1. Overview 1211
4.1.3.9.2. Request parameters 1212
4.1.3.9.3. Response parameters 1212
4.1.3.9.4. Examples 1212
4.1.3.10. SuspendSynchronizationJob 1212
4.1.3.10.1. Overview 1212
4.1.3.10.2. Request parameters 1213
4.1.3.10.3. Response parameters 1213
4.1.3.10.4. Examples 1213
4.1.4. Data subscription APIs 1213
4.1.4.1. ConfigureSubscriptionInstance 1214
4.1.4.1.1. Overview 1214

4.1.4.1.2. Request parameters 1214
4.1.4.1.3. SubscriptionObject parameter definitions 1215
4.1.4.1.4. Response parameters 1215
4.1.4.1.5. Example 1216
4.1.4.2. CreateSubscriptionInstance 1217
4.1.4.2.1. Overview 1217
4.1.4.2.2. Request parameters 1217
4.1.4.2.3. Response parameters 1217
4.1.4.2.4. Examples 1218
4.1.4.3. DeleteSubscriptionInstance 1218
4.1.4.3.1. Overview 1218
4.1.4.3.2. Request parameters 1218
4.1.4.3.3. Response parameters 1219
4.1.4.3.4. Examples 1219
4.1.4.4. DescribeSubscriptionInstances 1219
4.1.4.4.1. Overview 1219
4.1.4.4.2. Request parameters 1219
4.1.4.4.3. Response parameters 1220
4.1.4.4.4. Examples 1220
4.1.4.5. DescribeSubscriptionInstanceStatus 1221
4.1.4.5.1. Overview 1221
4.1.4.5.2. Request parameters 1221
4.1.4.5.3. Response parameters 1221
4.1.4.5.4. Endpoint parameter definitions 1223
4.1.4.5.5. SubscriptionObject parameter definitions 1223
4.1.4.5.6. Examples 1223
4.1.4.6. ModifyComsumptionTimestamp 1224
4.1.4.6.1. Overview 1224

4.1.4.6.2. Request parameters 1224
4.1.4.6.3. Response parameters 1224
4.1.4.6.4. Examples 1225
4.1.4.7. ModifySubscriptionObject 1225
4.1.4.7.1. Overview 1225
4.1.4.7.2. Request parameters 1225
4.1.4.7.3. SubscriptionObject parameter definitions 1226
4.1.4.7.4. Response parameters 1226
4.1.4.7.5. Examples 1227
4.1.4.8. StartSubscriptionInstance 1227
4.1.4.8.1. Overview 1227
4.1.4.8.2. Request parameters 1227
4.1.4.8.3. Response parameters 1228
4.1.4.8.4. Examples 1228
4.2. Call the Java SDK 1228
4.2.1. Quick start 1229
4.2.2. User manual 1230
4.2.2.1. Install the Java SDK 1230
4.2.2.2. Configure identity verification credentials 1230
4.2.2.3. Initiate a call 1231
4.2.2.4. Troubleshooting 1232
4.2.2.5. Errors and troubleshooting 1233
5.Cloud Native Distributed Database PolarDB-X 1235
5.1. API reference 1235
5.1.1. List of operations by function 1235
5.1.2. Make API requests 1240
5.1.3. Common parameters 1241
5.1.4. Request signature 1243

5.1.5. Database management 1247
5.1.5.1. ChangeAccountPassword 1247
5.1.5.2. CheckDrdsDbName 1248
5.1.5.3. CheckExpandStatus 1250
5.1.5.4. CheckSqlAuditEnableStatus 1252
5.1.5.5. CreateDrdsDB 1253
5.1.5.6. CreateShardTask 1256
5.1.5.7. DatalinkReplicationPrecheck 1258
5.1.5.8. DeleteShardTasks 1260
5.1.5.9. DescribeBackMenu 1261
5.1.5.10. DescribeBackupDbs 1264
5.1.5.11. DescribeBackupTimes 1265
5.1.5.12. DescribeBroadcastTables 1267
5.1.5.13. DescribeDbInstanceDbs 1271
5.1.5.14. DescribeDbInstances 1274
5.1.5.15. DescribeDrdsDB 1278
5.1.5.16. DescribeDrdsDbInstance 1281
5.1.5.17. DescribeDrdsDbInstances 1286
5.1.5.18. DescribeDrdsDbRdsNameList 1293
5.1.5.19. DescribeDrdsDbRdsRelationInfo 1295
5.1.5.20. DescribeDrdsDbTasks 1298
5.1.5.21. DescribeDrdsInstanceDbMonitor 1302
5.1.5.22. DescribeDrdsShardingDbs 1305
5.1.5.23. DescribeDrdsSlowSqls 1311
5.1.5.24. DescribeDrdsSqlAuditStatus 1314
5.1.5.25. DescribeExpandLogicTableInfoList 1316
5.1.5.26. DescribeGlobalBroadcastType 1317
5.1.5.27. DescribeHotDbList 1319

5.1.5.28. DescribeInstDbLogInfo 1320
5.1.5.29. DescribeRDSPerformance 1322
5.1.5.30. DescribeRdsPerformanceSummary 1324
5.1.5.31. DescribeRdsSuperAccountInstances 1327
5.1.5.32. DescribeRecycleBinStatus 1329
5.1.5.33. DescribeRecycleBinTables 1331
5.1.5.34. DescribeShardTaskInfo 1333
5.1.5.35. DescribeShardTaskList 1339
5.1.5.36. DescribeSqlFlashbakTask 1342
5.1.5.37. DescribeTable 1346
5.1.5.38. DescribeTableListByType 1351
5.1.5.39. DescribeTables 1354
5.1.5.40. DisableSqlAudit 1357
5.1.5.41. ModifyAccountPrivilege 1359
5.1.5.42. ModifyDrdsIpWhiteList 1361
5.1.5.43. ModifyRdsReadWeight 1363
5.1.5.44. DescribeDbInstances 1365
5.1.5.45. ReleaseHiStoreInstance 1369
5.1.5.46. RemoveBackupsSet 1370
5.1.5.47. RemoveDrdsDbFailedRecord 1372
5.1.5.48. RemoveRecycleBinTable 1373
5.1.5.49. ResetDrdsToRdsConnections 1375
5.1.5.50. SetupBroadcastTables 1377
5.1.5.51. SetupRecycleBinStatus 1378
5.1.5.52. SetupTable 1380
5.1.5.53. StartRestore 1381
5.1.5.54. SubmitCleanTask 1384
5.1.5.55. SubmitSmoothExpandPreCheck 1386

5.1.5.56. SubmitSmoothExpandPreCheckTask 1387
5.1.5.57. SubmitSmoothExpandTask 1389
5.1.5.58. SubmitSwitchTask 1391
5.1.5.59. ValidateShardTask 1393
5.1.5.60. RemoveDrdsDb 1397
5.1.5.61. RefreshDrdsAtomUrl 1399
5.1.6. Instance management 1400
5.1.6.1. CreateInstanceAccount 1400
5.1.6.2. DescribeDrdsDBs 1402
5.1.6.3. DescribeDrdsInstance 1405
5.1.6.4. DescribeDrdsInstanceLevelTasks 1411
5.1.6.5. DescribeDrdsInstanceMonitor 1414
5.1.6.6. DescribeDrdsInstances 1418
5.1.6.7. DescribeDrdsInstanceVersion 1425
5.1.6.8. DescribeDrdsParams 1427
5.1.6.9. DescribeDrdsTasks 1432
5.1.6.10. DescribeHiStoreInstanceInfo 1434
5.1.6.11. DescribeInstanceAccounts 1435
5.1.6.12. DescribeInstanceMenuSwitch 1437
5.1.6.13. DescribeInstDbSlsInfo 1438
5.1.6.14. DescribePreCheckResult 1440
5.1.6.15. DescribeStorageInstanceSubDbInfo 1441
5.1.6.16. EnableSqlAudit 1447
5.1.6.17. EnableSqlFlashbackMatchSwitch 1448
5.1.6.18. FlashbackRecycleBinTable 1450
5.1.6.19. ModifyAccountDescription 1452
5.1.6.20. ModifyDrdsInstanceDescription 1453
5.1.6.21. RemoveDrdsInstance 1455

5.1.6.22. RemoveInstanceAccount 1456
5.1.6.23. RestartDrdsInstance 1457
5.1.6.24. RollbackInstanceVersion 1459
5.1.6.25. SetBackupPolicy 1460
5.1.6.26. SetupDrdsParams 1463
5.1.6.27. SubmitRollbackTask 1465
5.1.6.28. SubmitSqlFlashbackTask 1467
5.1.6.29. SwitchGlobalBroadcastType 1470
5.1.6.30. UpgradeInstanceVersion 1471
5.2. Call SDKs for cloud services by using ASAPI 1473
5.2.1. Overview 1473
5.2.2. Log on to the API Tool console 1474
5.2.3. View the information about API operations 1475
5.2.4. Obtain the endpoint of ASAPI 1475
5.2.5. Obtain the SDK package for Asapi 1477
5.2.6. Obtain an AccessKey pair 1478
5.2.7. Query the region ID 1478
5.2.8. Call PolarDB-X SDK for Java 1479
5.2.9. Call PolarDB-X SDK for Node.js 1481
5.2.10. (Optional) Use POP SDK to call ASAPI API operations 1483

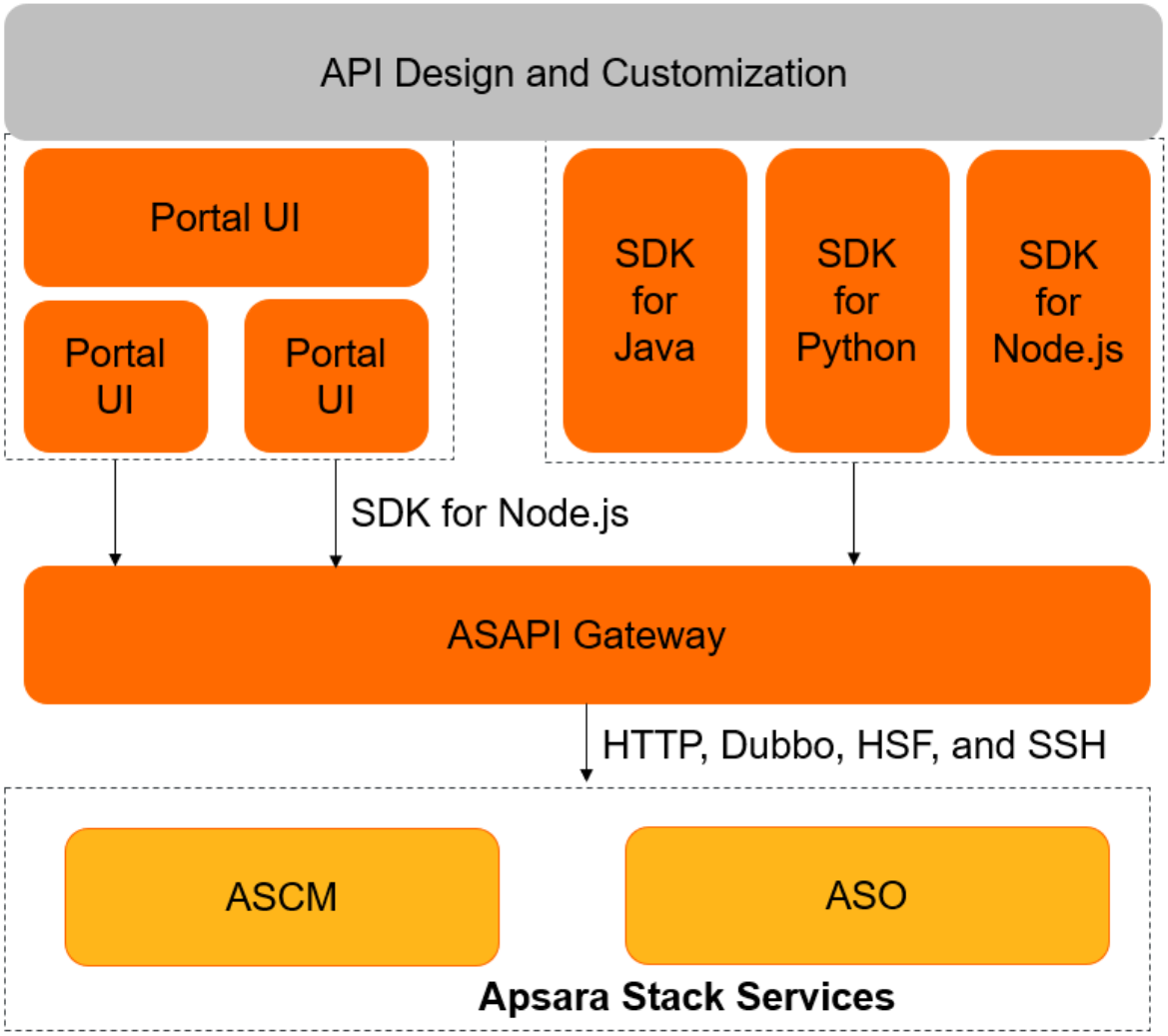
1.ASCM console

1.1. ASAPI Developer Guide

1.1.1. Overview

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

ASAPI is a platform that integrates all features and is the only portal to connect to Apsara Stack. ASAPI can categorize operations and audit and manage their calls. It can also assemble and customize existing operations and adapt to your operations. Each cloud service can call ASAPI for required data and features, avoiding inconsistent operations caused by different call methods and separate encapsulation. This can ensure the operation program integrity and improve development efficiency and code portability.



ASAPI allows you to manage and call operations in a unified manner and customize development. ASAPI supports access of most cloud services deployed in Apsara Stack and passthrough or encapsulation of different backend APIs. This way, operations can be called with the same method. ASAPI allows you to use SDKs and API Portal to call APIs registered in ASAPI, customize SDKs, and manage versions of Apsara Stack, services, and APIs.

1.1.2. Log on to the API Tool console

You can log on to the API Tool console through the Apsara Stack Cloud Management (ASCM) console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password used to log on to the console from the operations administrator.

   > **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum 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. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%).

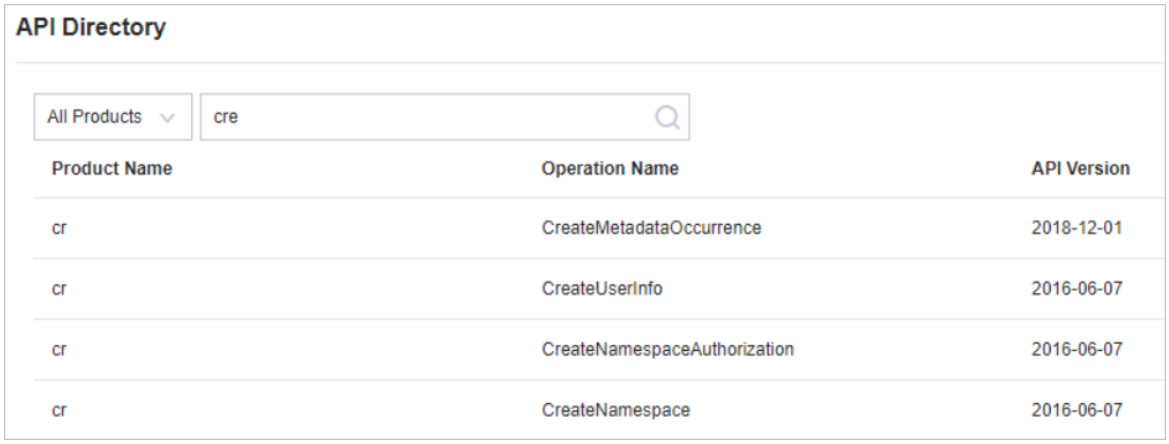
3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products** > **Others** > **API Tool**.

1.1.3. View the information about API operations

You must obtain a service name, API operation name, and API version. All the information is required when you use SDKs.

Procedure

1. Log on to the API Tool console.
2. In the left-side navigation pane, click **API fold**.
3. On the **API Directory** page, select the target service from the drop-down list, enter the API operation name in the search box, and then click the search icon.



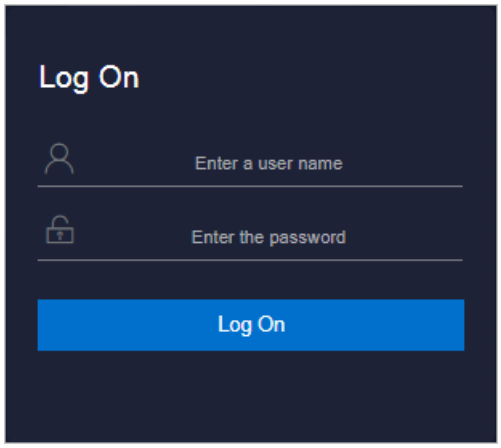
4. View and remember the service name, API operation name, and API version.

1.1.4. Obtain the endpoint of Asapi

You can obtain the endpoint of Apsara Stack API (ASAPI) in the Apsara Stack Operations (ASO) 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.

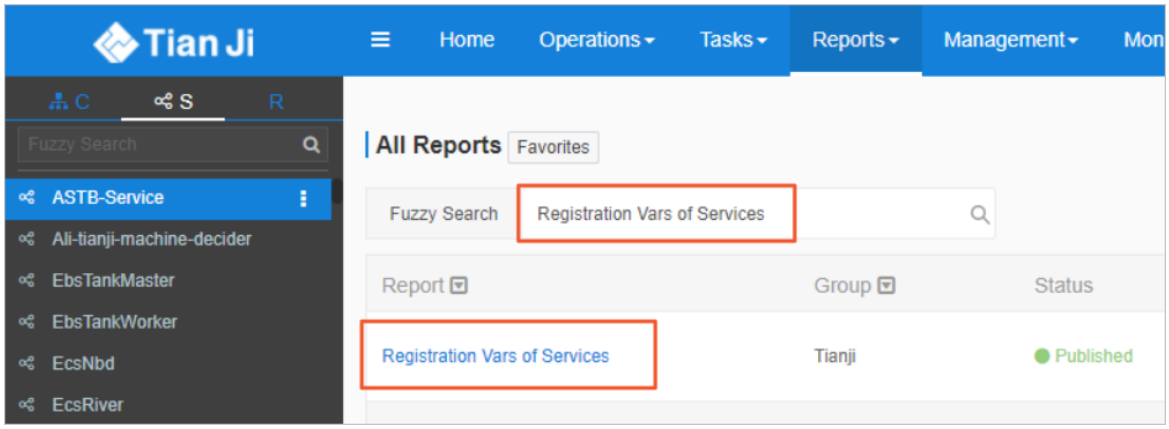
When you log on to the ASO console for the first time, you must change the password of your username as prompted.

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

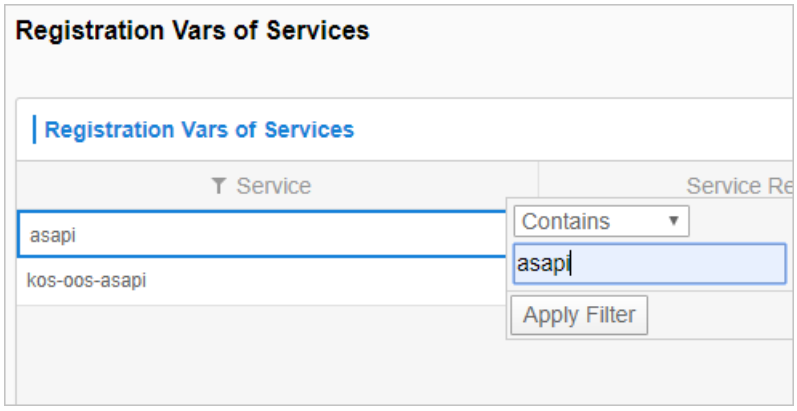
- It must contain uppercase and lowercase letters.

- 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.

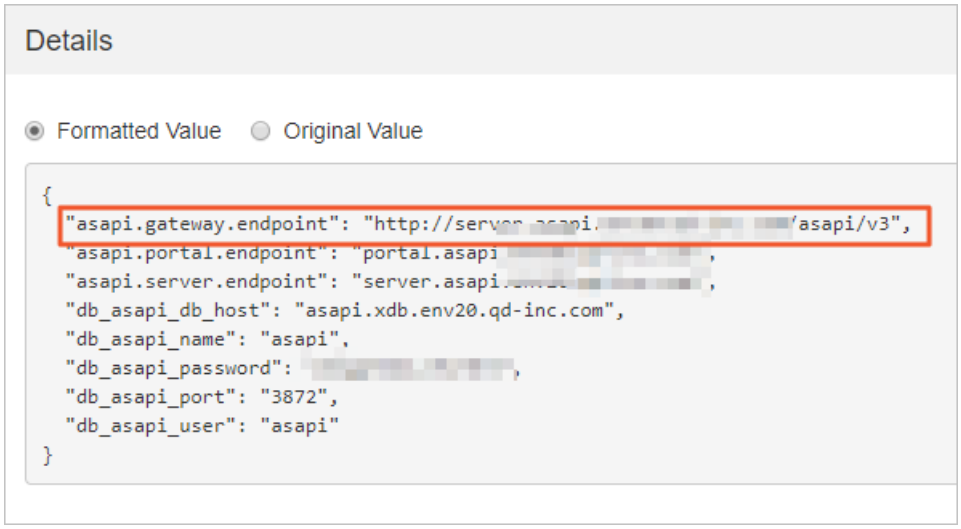
3. Click **Log On** to go to the **ASO** console.
4. In the left-side navigation pane, choose **Products** > **Product List**. On the Product List page, click **Apsara Infrastructure Management Framework**.
5. In the left-side navigation pane of the Apsara Infrastructure Management Framework console, click **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 corresponding to asapi and select **Show More**.
9. In the **Details** dialog box, view the value of **asapi.gateway.endpoint**. The value is the logon URL of the Apsara ASAPI Management platform.



1.1.5. Obtain the SDK package for Asapi

Apsara Stack uses Asapi as the unified portal for all Apsara Stack products that are connected to Asapi. Asapi also provides SDKs to call product APIs.

To obtain the SDK package for Asapi, see Apsara Stack documentation.

1.1.6. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only operations administrators and 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:

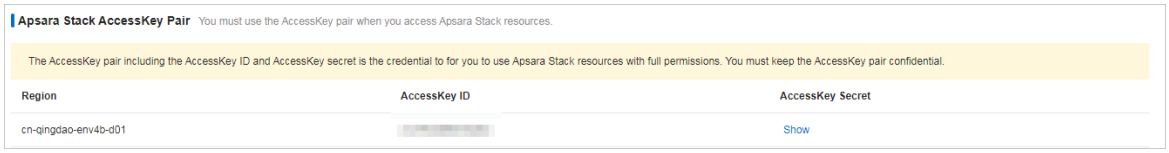
1. Log on to the Apsara Stack Cloud Management (ASCM) console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, click the icon on the right of the organization to be added.
5. Select **AccessKey**.
6. In the message that appears, view the AccessKey pair of the organization.

> **Note** An AccessKey pair is automatically allocated to a level-1 organization. Subordinate organizations use the same AccessKey pair as their level-1 organization.

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 ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section of the User Information page, view your AccessKey pair.



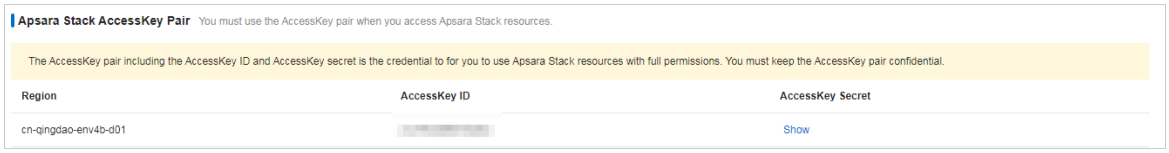
> **Note** The AccessKey pair consists of an AccessKey ID and an AccessKey secret. They allow you to access Apsara Stack resources with full permissions for your account. You must keep them confidential.

1.1.7. Query the region ID

You must specify the region ID when you call Apsara Stack SDK for Node.js. You must query the region ID in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section, you can view the region ID.



1.1.8. SDK for Java

This topic describes how to obtain and use Cloud Native Distributed Database PolarDB-X (PolarDB-X) SDK for Java.

Prerequisites

- You must have an Apsara Stack tenant account and an AccessKey pair (AccessKey ID and AccessKey secret) to use PolarDB-X SDK for Java. For more information about how to obtain and view your AccessKey pair in the Apsara Stack Cloud Management (ASCM) console, see Obtain an AccessKey pair.
- You have obtained the endpoint of Apsara Stack API (ASAPI). For more information, see Obtain the endpoint of Asapi.
- The version of PolarDB-X SDK for Java is 1.6 or later.

Add the dependency

Add the following dependency to pom.xml:

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>asapi</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/lib/aliyun-java-sdk-asapi.jar</systemPath>
</dependency>
```

Use SDK for Java

You can call ASAPI API operations by sending GET and POST requests.

- GET request

```
 ASClient client = new ASClient();

// Form 1
String result = client.doGet(endpoint, parameters);

// Form 2
String result = client.doRequest(endpoint, parameters);
```

- POST request

```
// Set the endpoint of ASAPI.
String endpoint = "http://xxx/asapi/v3";

// Set request parameters.
Map<String, String> parameters = new HashMap<String, String>();

// Set product parameters.
parameters.put("Product", "ascm");          // The product name, which must be the same as that disp
layed on the APIs page.
parameters.put("Action", "AddUser");        // The API operation, which is case-sensitive and must b
e the same as that on the APIs page.
parameters.put("Version", "2019-05-10");    // The API version.

// Set authorization parameters.
parameters.put("AccessKeyId", "<your-access-key-id>");        // The AccessKey ID of the authorized
account.
parameters.put("AccessKeySecret", "<your-access-key-secret>"); // The AccessKey secret of the a
uthorized account.

// Set service parameters.
parameters.put("username", "<your-username>");              // The username used to log on to the
ASCM console
parameters.put("password", "<your-password>");              // The password used to log on to the A
SCM console
JSONObject jsonObject = new JSONObject();
jsonObject.put("param1", "json parameter 1");
jsonObject.put("param2", "json parameter 2");
parameters.put("json", jsonObject.toJSONString());

// Create an ASClient instance for initiating requests.
ASClient client = new ASClient();
// Add the request header.
client.addHeader("x-acs-resourceGroupId", "112233");
// Initiate the request and obtain the response.
String result = client.doPost(endpoint, parameters);
```

1.1.9. SDK for Node.js

This topic describes how to obtain and use Apsara Stack SDK for Node.js.

Prerequisites

- You must have an authorized account and an AccessKey pair (AccessKey ID and AccessKey secret) to use Apsara Stack SDK for Node.js. For more information, see Obtain an AccessKey pair.
- You have obtained the endpoint of Asapi. For more information about how to query the endpoint of Asapi, see Obtain the endpoint of Asapi.
- You have obtained the region ID. For more information about how to query the region ID, see Query the region ID.

Add dependencies

1. Install Apsara Stack SDK for Node.js. `tnpm install --save @ali/asapi-nodejs-sdk`
2. If *mocha.opts* depends on eggjs, you must configure *mocha.opts*.

```
--recursive
--colors
--timeout 60000
--reporter mochawesome
--require co-mocha
--require egg-ts-helper/register
--require espower-typescript/guess
-- require intelli-espower-loader
```

Use SDK for Node.js

The following example shows how to use Apsara Stack SDK for Node.js:

1. Create a client instance. To create a client instance, you must set the region ID, endpoint, AccessKey ID, and AccessKey secret in the *config.ts* file. Example:

```
local: {
  regionId: 'obtained region ID',
  API_GATEWAY: 'obtained endpoint',
  accessKeyId: 'your AccessKey ID',
  accessKeySecret: 'your AccessKey secret',
},
```

2. Create an API request and configure parameters.
3. Initiate the request and handle any responses or exceptions.

```
// Speciofy the Asapi endpoint.
const API_GATEWAY = "http://xxxx/asapi/v3"

// Create the rpc client.
const asapiClient = ctx.service.sdk.rpcClient

// Set request parameters.
let params = {
   Action: 'TestAPI',
   Product: 'test',
   regionId: 'cn-qingdao-env12-d01',
   accessKeyId: 'xxxxxxx',
   accessKeySecret: 'xxxxxxx',
   Version: '2018-01-22',
};

// Set parameters in the request header.
asapiClient.addHeader("x-acs-regionId", "cn-hangzhou");

// Initiate the request and obtain the response.
const result = await asapiClient.doRequest(API_GATEWAY, params, {});
```

1.1.10. Use the POP SDK to call ASAPI

If an application used the POP SDK to call cloud services, you must switch the gateway to ASAPI by performing the following operations:

Procedure

Reference the following sample code and add required common parameters and the **Product Name** field to the request header. Use the code to switch the gateway to ASAPI.

> **Note** Different services use different endpoints in the POP SDK. However, ASAPI provides only one endpoint. You cannot identify the service to which an API operation belongs based on the API operation name and the version number. For example, both ECS and KVStore for Redis provide the DescribeInstances API operation of the 2015-01-01 version.

Sample code

The following sample code shows how to use POP SDK to call ECS ASAPI.

```
try {
        DefaultProfile clientProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
        // Set the ASAPI gateway. Format: server.asapi.${intranet-domain}.
        clientProfile.addEndpoint(region, region, "ecs", "<ASAPI gateway>");
        DefaultAcsClient acsClient = new DefaultAcsClient(clientProfile);

        DescribeImagesRequest request = new DescribeImagesRequest();
        request.setSysRegionId(regionId); // Or setRegionId(regionId)
        // Set common parameters based on the information about the current user.
        rrequest.putHeaderParameter("x-acs-organizationid", "<Organization ID>");
        request.putHeaderParameter("x-acs-resourcegroupid", "<Resource group ID>");
        request.putHeaderParameter("x-ascm-product-name", "<Service name>");
        request.putHeaderParameter("x-ascm-product-version", "<API version");
        request.putHeaderParameter("x-ascm-product-ak", "<AccessKey ID>");
        request.putHeaderParameter("x-ascm-product-sk", "<AccessKey Secret>");

        DescribeImagesResponse result = (DescribeImagesResponse) acsClient.getAcsResponse(reque
st);
        List<DescribeImagesResponse.Image> images = result.getImages();
        for (DescribeImagesResponse.Image image: images) {
           System.out.println(image.getImageId());
           System.out.println(image.getImageName());
        }

     } catch (Exception e) {
        e.printStackTrace();
     }
```

Header parameters

Parameter	Description
x-ascm-product-name	The name of the service to which the API operation belongs.
x-ascm-product-version	The API version.
x-ascm-product-ak	The AccessKey ID of the organization in the ASCM console.
x-ascm-product-sk	The AccessKey secret of the organization in the ASCM console.

1.2. API reference

1.2.1. Common parameters

Common request parameters

Common request parameters must be included in all API requests.

Common request parameters

Parameter	Type	Required	Description
Version	String	Yes	The version number of the API, in YYYY-MM-DD format. Set the value to 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.

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 JSON format. Every response returns a unique RequestID regardless of whether the call is successful.

1.2.2. Make API requests

ASCM operations are available in ASAPI. You must use the ASAPI SDK to make API requests.

You must add corresponding request parameters in the request based on the operation description. After the operation is called, the system returns the result. All requests and responses are encoded in UTF-8. You can call ASAPI API operations by using GET and POST.

The following example uses the Java SDK:

Add the dependency

Add the dependency to pom.xml:

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>asapi</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</dependency>
```

Sample requests

- GET request:

```
ASClient client = new ASClient();
String result = client.doGet(endpoint, parameters);
// or
String result = client.doRequest(endpoint, parameters);
```

- POST request:

```
// Configure the endpoint.
String endpoint = "http://xxx/asapi/v3";
// Configure request parameters.
Map<String, String> parameters = new HashMap<String, String>();
parameters.put("Product", "ascm");
parameters.put("Action", "AddMockUser");
parameters.put("Version", "2019-05-10");
parameters.put("AccessKeyId", "xxx");
parameters.put("username", "xxx");
parameters.put("password", "xxx");
parameters.put("alias", "xxx");

// Add JSON objects.
JSONObject jsonObject = new JSONObject();
parameters.put("json", jsonObject.toJSONString());

// Configure the client.
ASClient client = new ASClient();

// Add the request header.
client.addHeader("x-acs-resourceGroupId", "112233");

// Initiate the request and obtain the response.
String result = client.doPost(endpoint, parameters);
```

1.2.3. ASCM authentication

Before you can make ASAPI requests, you must obtain the AccessKey pair (AccessKey ID and AccessKey secret) of the department account in the Apsara Stack Cloud Management (ASCM) console.

The department account grants access permissions on available resources. The resources can then be accessed by using the AccessKey pair. For more information about how to obtain the AccessKey pair, see Obtain an AccessKey pair.

1.2.4. Enterprise

1.2.4.1. Organizations

1.2.4.1.1. UpdateOrganization

You can call this operation to update an organization.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateOrganization	The operation that you want to perform. Set the value to UpdateOrganization.
Id	String	Yes	1	The ID of the organization.
Name	String	Yes	helloworld	The name of the organization.
RequestId	String	No	sdfsadfsdfsdfsdf	The ID of the request.
PersonNum	String	No	10	A reserved parameter.
ResourceGroupNum	String	No	10	A reserved parameter.
Alias	String	No	test	A reserved parameter.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=UpdateOrganization
&Id=1
&Name=helloworld
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
"code":"200",
"cost":3,
"data":{
"alias":"testNew_b82d1a37-2e33-4b67-9510-bf0c4082f125",
"cuserId":"",
"id":93,
"internal":false,
"level":"0.1",
"mtime":1574762643696,
"muserId":"",
"name":"testNew_b82d1a37-2e33-4b67-9510-bf0c4082f125",
"parentId":1,
"supportRegions":"cn-qingdao-env8d-d01"
},
"message":"success",
"redirect":false,
"requestId":"6d5be854-0394-49a3-8609-7c216424faa4",
"success":true,
"asapiRequestId":"72ac550d-83f4-4841-b989-860a566a7637"
}
```


1.2.4.1.2. UpdateSupportRegionsForOrg

You can call this operation to update the supported regions for an organization.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateSupportRegionsForOrg	The operation that you want to perform. Set the value to UpdateSupportRegionsForOrg.
OrganizationId	Integer	Yes	5	The ID of the organization.
SupportRegions	String	Yes	RegionId1,RegionId2	The list of region IDs.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=UpdateSupportRegionsForOrg
&OrganizationId=5
&SupportRegions=RegionId1,RegionId2
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 4,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "9e2c88fc-c86e-4c73-bb63-838f23fe68b1"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>4</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>9e2c88fc-c86e-4c73-bb63-838f23fe68b1</asapiRequestId>
```

1.2.4.1.3. GetOrganization

You can call this operation to query an organization.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetOrganization	The operation that you want to perform. Set the value to GetOrganization.
Id	String	Yes	1	The ID of the organization.
RequestId	String	No	dsfsdfsd	

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetOrganization
&Id=1
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
"code":"200",
"cost":14,
"data":{
"alias":"nana01",
"cuserId":"",
"id":7,
"internal":false,
"level":"0.1.5",
"muserId":"",
"name":"nana01",
"parentId":5
},
"message":"success",
"redirect":false,
"requestId":"42a15796-b6d7-4131-aaba-a2bf8b8c3fce",
"success":true,
"asapiRequestId":"01f351c5-5878-4300-99c9-a703e8e00a04"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>14</cost>
<data>
   <alias>nana01</alias>
   <cuserId/>
   <id>7</id>
   <internal>false</internal>
   <level>0.1.5</level>
   <muserId/>
   <name>nana01</name>
   <parentId>5</parentId>
</data>
<message>success</message>
<redirect>false</redirect>
<requestId>42a15796-b6d7-4131-aaba-a2bf8b8c3fce</requestId>
<success>true</success>
<asapiRequestId>01f351c5-5878-4300-99c9-a703e8e00a04</asapiRequestId>
```

1.2.4.1.4. GetOrganizationList

You can call this operation to query a list of organizations.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetOrganizationList	The operation that you want to perform. Set the value to GetOrganizationList.
RequestId	String	No	sdfasdfafsdf	The ID of the request.
Id	String	No	1	The ID of the organization.
Name	String	No	test	A reserved parameter.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetOrganizationList
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
"code":"200",
"cost":14,
"data":[
{
"alias":"nana01",
"cuserId":"",
"id":7,
"internal":false,
"level":"0.1.5",
"muserId":"",
"name":"nana01",
"parentId":5
}],
"message":"success",
"redirect":false,
"requestId":"42a15796-b6d7-4131-aaba-a2bf8b8c3fce",
"success":true,
"asapiRequestId":"01f351c5-5878-4300-99c9-a703e8e00a04"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>14</cost>
<data>
   <alias>nana01</alias>
   <cuserId/>
   <id>7</id>
   <internal>false</internal>
   <level>0.1.5</level>
   <muserId/>
   <name>nana01</name>
   <parentId>5</parentId>
</data>
<message>success</message>
<redirect>false</redirect>
<requestId>42a15796-b6d7-4131-aaba-a2bf8b8c3fce</requestId>
<success>true</success>
<asapiRequestId>01f351c5-5878-4300-99c9-a703e8e00a04</asapiRequestId>
```

1.2.4.1.5. RemoveOrganization

You can call this operation to remove an organization.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveOrganization	The operation that you want to perform. Set the value to RemoveOrganization.
Id	String	Yes	10	The ID of the organization.
RequestId	String	No	retertertet	The ID of the request.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=RemoveOrganization
&Id=10
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 13,
  "message": "success",
  "redirect": false,
  "requestId": "269b4d46-9647-47b2-83c9-80f1a6b5077b",
  "success": true,
  "asapiRequestId": "38e8cc13-e5b1-4f5e-a1f7-e6a092309f32"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>13</cost>
<message>success</message>
<redirect>false</redirect>
<requestId>269b4d46-9647-47b2-83c9-80f1a6b5077b</requestId>
<success>true</success>
<asapiRequestId>38e8cc13-e5b1-4f5e-a1f7-e6a092309f32</asapiRequestId>
```

1.2.4.1.6. ListResourceGroupByOrgId

You can call this operation to query the list of resource sets in an organization.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListResourceGroupByOrgId	The operation that you want to perform. Set the value to ListResourceGroupByOrgId.
Id	String	Yes	3	The ID of the organization.
resource_group_name	String	No	testrs	The name of the resource set.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListResourceGroupByOrgId
&Id=3
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 9,
  "data": [
    {
      "creator": "admin",
      "gmtCreated": 1574652263873,
      "gmtModified": 1574652263873,
      "id": 2,
      "organizationID": 3
```

```
        "organizationID": 3,
        "organizationName": "guyi-test",
        "resourceGroupName": "DefaultResourceSet(aaa)",
        "resourceGroupType": 1
      },
      {
        "creator": "admin",
        "gmtCreated": 1575177273979,
        "id": 8,
        "organizationID": 3,
        "organizationName": "guyi-test",
        "resourceGroupName": "DefaultResourceSet(drds-test)"
      },
      {
        "creator": "admin",
        "gmtCreated": 1575692481707,
        "id": 17,
        "organizationID": 3,
        "organizationName": "guyi-test",
        "resourceGroupName": "testrs"
      }
    ],
    "message": "success",
    "pageInfo": {
      "currentPage": 0,
      "pageSize": 0,
      "total": 0,
      "totalPage": 0
    },
    "redirect": false,
    "success": true,
    "asapiRequestId": "35fab935-ccdc-4b5f-a508-3d765b86b8f7"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>9</cost>
<data>
  <creator>admin</creator>
  <gmtCreated>1574652263873</gmtCreated>
  <gmtModified>1574652263873</gmtModified>
  <id>2</id>
  <organizationID>3</organizationID>
  <organizationName>guyi-test</organizationName>
  <resourceGroupName>DefaultResourceSet(aaa)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1575177273979</gmtCreated>
  <id>8</id>
  <organizationID>3</organizationID>
  <organizationName>guyi-test</organizationName>
  <resourceGroupName>DefaultResourceSet(drds-test)</resourceGroupName>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1575692481707</gmtCreated>
  <id>17</id>
  <organizationID>3</organizationID>
  <organizationName>guyi-test</organizationName>
  <resourceGroupName>testrs</resourceGroupName>
</data>
<message>success</message>
<pageInfo>
  <currentPage>0</currentPage>
  <pageSize>0</pageSize>
  <total>0</total>
  <totalPage>0</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>35fab935-ccdc-4b5f-a508-3d765b86b8f7</asapiRequestId>
```

1.2.4.1.7. CreateOrganization

You can call this operation to create an organization.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateOrganization	The operation that you want to perform. Set the value to CreateOrganization.
Name	String	Yes	hellowolrd	The name of the organization.
ParentId	String	Yes	1	The ID of the parent organization.
RequestId	String	No	100000000	None
PersonNum	String	No	1	A reserved parameter.
ResourceGroupNum	String	No	1	A reserved parameter.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=CreateOrganization
&Name=hellowolrd
&ParentId=1
&<Common request parameters>
```

- Sample success responses in JSON format

```
{ "
code":"200",
"cost":8,
"data":{
"alias":"SubOrg20191126-070129",
"cuserId":"",
"id":95,
"internal":false,
"level":"0.1.52",
"muserId":"",
"name":"SubOrg20191126-070129",
"parentId":52
},
"message":"success",
"redirect":false,
"requestId":"957f04f3-428e-4865-a33d-4a5acc990855",
"success":true,
"asapiRequestId":"c7a1bae9-cd46-4a53-8737-76b114af1f2f"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>8</cost>
<data>
  <alias>SubOrg20191126-070129</alias>
  <cuserId/>
  <id>95</id>
  <internal>false</internal>
  <level>0.1.52</level>
  <muserId/>
  <name>SubOrg20191126-070129</name>
  <parentId>52</parentId>
</data>
<message>success</message>
<redirect>false</redirect>
<requestId>957f04f3-428e-4865-a33d-4a5acc990855</requestId>
<success>true</success>
<asapiRequestId>c7a1bae9-cd46-4a53-8737-76b114af1f2f</asapiRequestId>
```


1.2.4.2. Resource sets

1.2.4.2.1. BindAscmUserAndResourceGroup

You can call this operation to bind a resource set to a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	BindAscmUserAndResourceGroup	The operation that you want to perform. Set the value to BindAscmUserAndResourceGroup.
ascm_user_ids	String	Yes	"[\"5017\",\"5023\"]"	The list of user IDs.
resource_group_id	Integer	Yes	7	The ID of the resource set.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=BindAscmUserAndResourceGroup
&ascm_user_ids="[\"5017\",\"5023\"]"
&resource_group_id=7
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 11,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "be356a4f-265e-4380-9a6f-c09668796ed8"
}
```

1.2.4.2.2. CreateResourceGroup

You can call this operation to create a resource set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateResourceGroup	The operation that you want to perform. Set the value to CreateResourceGroup.
organization_id	Integer	Yes	3	The ID of the organization.
resource_group_name	String	Yes	testrs	The name of the resource set.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=CreateResourceGroup
&organization_id=3
&resource_group_name=testrs
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 13,
  "data": {
    "creator": "admin",
    "gmtCreated": 1575692481707,
    "id": 17,
    "organizationID": 3,
    "resourceGroupName": "testrs"
  },
  "message": "success",
  "redirect": false,
  "requestId": "699b5aa9-e154-40a5-b26d-908966a58632",
  "success": true,
  "asapiRequestId": "3f4f5a56-1e5f-40f4-b477-7b0a62ac36f1"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>13</cost>
<data>
  <creator>admin</creator>
  <gmtCreated>1575692481707</gmtCreated>
  <id>17</id>
  <organizationID>3</organizationID>
  <resourceGroupName>testrs</resourceGroupName>
</data>
<message>success</message>
<redirect>false</redirect>
<requestId>699b5aa9-e154-40a5-b26d-908966a58632</requestId>
<success>true</success>
<asapiRequestId>3f4f5a56-1e5f-40f4-b477-7b0a62ac36f1</asapiRequestId>
```

1.2.4.2.3. UnbindAscmUserAndResourceGroup

You can call this operation to unbind a user from a resource set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UnbindAscmUserAndResourceGroup	The operation that you want to perform. Set the value to UnbindAscmUserAndResourceGroup.
ascm_user_ids	String	Yes	"[\"5023\",\"5017\"]"	The list of user IDs.
resource_group_id	Integer	Yes	7	The ID of the resource set.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=UnbindAscmUserAndResourceGroup
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 12,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "b963358b-4f9f-4c2e-a7f2-46191fc1e1f5"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>12</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>b963358b-4f9f-4c2e-a7f2-46191fc1e1f5</asapiRequestId>
```

1.2.4.2.4. UpdateResourceGroup

You can call this operation to modify the information of a resource set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateResourceGroup	The operation that you want to perform. Set the value to UpdateResourceGroup.
id	String	Yes	16	The ID of the resource set.
resource_group_name	String	Yes	DefaultRewwww	The new name of the resource set.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=UpdateResourceGroup
&id=16
&resource_group_name=DefaultRewwww
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 3,
  "data": {
    "gmtModified": 1575698412237,
    "id": 16,
    "resourceGroupName": "DefaultRewwww"
  },
  "message": "success",
  "redirect": false,
  "requestId": "8befdaa3-7529-4ad7-a2e2-a95a2b00d45a",
  "success": true,
  "asapiRequestId": "7e1680da-544d-4e92-831c-9eb15e8b3ab2"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>3</cost>
<data>
  <gmtModified>1575698412237</gmtModified>
  <id>16</id>
  <resourceGroupName>DefaultRewwww</resourceGroupName>
</data>
<message>success</message>
<redirect>false</redirect>
<requestId>8befdaa3-7529-4ad7-a2e2-a95a2b00d45a</requestId>
<success>true</success>
<asapiRequestId>7e1680da-544d-4e92-831c-9eb15e8b3ab2</asapiRequestId>
```


1.2.4.2.5. RemoveResourceGroup

You can call this operation to remove a resource set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveResourceGroup	The operation that you want to perform. Set the value to RemoveResourceGroup.

Parameter	Type	Required	Example	Description
resource_group_id	Integer	Yes	5	The ID of the resource set.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=RemoveResourceGroup
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 9,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "4cb8e83a-2ee1-4191-97d7-65bc90938b51"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>9</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>4cb8e83a-2ee1-4191-97d7-65bc90938b51</asapiRequestId>
```

1.2.4.2.6. ListResourceGroup

You can call this operation to query the list of available resource sets of a user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 35,
```

```
"data": [
  {
    "creator": "ASCM-SYSTEM",
    "gmtCreated": 1574415325387,
    "gmtModified": 1574415325387,
    "id": 1,
    "organizationID": 2,
    "organizationName": "SystemDefaultOrganization",
    "resourceGroupName": "DefaultResourceSet(SystemDefaultOrganization)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1574652263873,
    "gmtModified": 1574652263873,
    "id": 2,
    "organizationID": 3,
    "organizationName": "guyi-test",
    "resourceGroupName": "DefaultResourceSet(aaa)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1574652289959,
    "gmtModified": 1574652289959,
    "id": 3,
    "organizationID": 4,
    "organizationName": "guyi-quota1",
    "resourceGroupName": "DefaultResourceSet(guyi-quota1)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1574674717192,
    "gmtModified": 1574674717192,
    "id": 4,
    "organizationID": 5,
    "organizationName": "dataworks_test",
    "resourceGroupName": "DefaultResourceSet(dataworks_test)",
    "resourceGroupType": 1
  },
```

```
{
    "creator": "admin",
    "gmtCreated": 1574909023628,
    "gmtModified": 1574909023628,
    "id": 6,
    "organizationID": 7,
    "organizationName": "ecs_test",
    "resourceGroupName": "DefaultResourceSet(ecs_test)",
    "resourceGroupType": 1
},
{
    "creator": "admin",
    "gmtCreated": 1574920237061,
    "gmtModified": 1574920237061,
    "id": 7,
    "organizationID": 8,
    "organizationName": "lqx-test",
    "resourceGroupName": "DefaultResourceSet(lqx-test)",
    "resourceGroupType": 1
},
{
    "creator": "admin",
    "gmtCreated": 1575177273979,
    "id": 8,
    "organizationID": 3,
    "organizationName": "guyi-test",
    "resourceGroupName": "DefaultResourceSet(drds-test)"
},
{
    "creator": "admin",
    "gmtCreated": 1575178007522,
    "gmtModified": 1575178007522,
    "id": 9,
    "organizationID": 9,
    "organizationName": "drds_test",
    "resourceGroupName": "DefaultResourceSet(drds_test)",
    "resourceGroupType": 1
},
{
    "creator": "admin",
    "gmtCreated": 1575178015019
```

```
    "gmtCreated": 1575178019019,
    "id": 10,
    "organizationID": 9,
    "organizationName": "drds_test",
    "resourceGroupName": "drds_test"
  },
  {
    "creator": "admin",
    "gmtCreated": 1575447221761,
    "gmtModified": 1575447221761,
    "id": 12,
    "organizationID": 11,
    "organizationName": "dataworks_new",
    "resourceGroupName": "DefaultResourceSet(dataworks_new)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1575452086085,
    "gmtModified": 1575452086085,
    "id": 13,
    "organizationID": 12,
    "organizationName": "kenny-test",
    "resourceGroupName": "DefaultResourceSet(kenny-test)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1575514294110,
    "gmtModified": 1575514294110,
    "id": 14,
    "organizationID": 13,
    "organizationName": "AJC311",
    "resourceGroupName": "DefaultResourceSet(AJC311)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1575514325218,
    "id": 15,
    "organizationID": 13,
```

```
            "organizationName": "AJC311",
            "resourceGroupName": "AJC311"
        },
        {
            "creator": "admin",
            "gmtCreated": 1575598055480,
            "gmtModified": 1575598055480,
            "id": 16,
            "organizationID": 14,
            "organizationName": "holotest",
            "resourceGroupName": "DefaultResourceSet(holotest)",
            "resourceGroupType": 1
        }
    ],
    "message": "success",
    "pageInfo": {
        "currentPage": 0,
        "pageSize": 0,
        "total": 0,
        "totalPage": 0
    },
    "redirect": false,
    "success": true,
    "asapiRequestId": "34f61c00-952d-4139-8779-9880747b1b0d"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>35</cost>
<data>
    <creator>ASCM-SYSTEM</creator>
    <gmtCreated>1574415325387</gmtCreated>
    <gmtModified>1574415325387</gmtModified>
    <id>1</id>
    <organizationID>2</organizationID>
    <organizationName>SystemDefaultOrganization</organizationName>
    <resourceGroupName>DefaultResourceSet(SystemDefaultOrganization)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
```

```
    <gmtCreated>1574652263873</gmtCreated>
    <gmtModified>1574652263873</gmtModified>
    <id>2</id>
    <organizationID>3</organizationID>
    <organizationName>guyi-test</organizationName>
    <resourceGroupName>DefaultResourceSet(aaa)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1574652289959</gmtCreated>
    <gmtModified>1574652289959</gmtModified>
    <id>3</id>
    <organizationID>4</organizationID>
    <organizationName>guyi-quota1</organizationName>
    <resourceGroupName>DefaultResourceSet(guyi-quota1)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1574674717192</gmtCreated>
    <gmtModified>1574674717192</gmtModified>
    <id>4</id>
    <organizationID>5</organizationID>
    <organizationName>dataworks_test</organizationName>
    <resourceGroupName>DefaultResourceSet(dataworks_test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1574909023628</gmtCreated>
    <gmtModified>1574909023628</gmtModified>
    <id>6</id>
    <organizationID>7</organizationID>
    <organizationName>ecs_test</organizationName>
    <resourceGroupName>DefaultResourceSet(ecs_test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1574920237061</gmtCreated>
```

```
    <gmtCreated>1574920259061</gmtCreated>
    <gmtModified>1574920237061</gmtModified>
    <id>7</id>
    <organizationID>8</organizationID>
    <organizationName>lqx-test</organizationName>
    <resourceGroupName>DefaultResourceSet(lqx-test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575177273979</gmtCreated>
    <id>8</id>
    <organizationID>3</organizationID>
    <organizationName>guyi-test</organizationName>
    <resourceGroupName>DefaultResourceSet(drds-test)</resourceGroupName>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575178007522</gmtCreated>
    <gmtModified>1575178007522</gmtModified>
    <id>9</id>
    <organizationID>9</organizationID>
    <organizationName>drds_test</organizationName>
    <resourceGroupName>DefaultResourceSet(drds_test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575178015019</gmtCreated>
    <id>10</id>
    <organizationID>9</organizationID>
    <organizationName>drds_test</organizationName>
    <resourceGroupName>drds_test</resourceGroupName>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575447221761</gmtCreated>
    <gmtModified>1575447221761</gmtModified>
    <id>12</id>
    <organizationID>11</organizationID>
    <organizationName>dataworks_new</organizationName>
```

```
    <resourceGroupName>DefaultResourceSet(dataworks_new)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575452086085</gmtCreated>
    <gmtModified>1575452086085</gmtModified>
    <id>13</id>
    <organizationID>12</organizationID>
    <organizationName>kenny-test</organizationName>
    <resourceGroupName>DefaultResourceSet(kenny-test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575514294110</gmtCreated>
    <gmtModified>1575514294110</gmtModified>
    <id>14</id>
    <organizationID>13</organizationID>
    <organizationName>AJC311</organizationName>
    <resourceGroupName>DefaultResourceSet(AJC311)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575514325218</gmtCreated>
    <id>15</id>
    <organizationID>13</organizationID>
    <organizationName>AJC311</organizationName>
    <resourceGroupName>AJC311</resourceGroupName>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575598055480</gmtCreated>
    <gmtModified>1575598055480</gmtModified>
    <id>16</id>
    <organizationID>14</organizationID>
    <organizationName>holotest</organizationName>
    <resourceGroupName>DefaultResourceSet(holotest)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
```

```
<message>success</message>
<pageInfo>
  <currentPage>0</currentPage>
  <pageSize>0</pageSize>
  <total>0</total>
  <totalPage>0</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>34f61c00-952d-4139-8779-9880747b1b0d</asapiRequestId>
```

1.2.4.2.7. ListAscmUsersInsideResourceGroup

You can call this operation to query the list of users within a resource set.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListAscmUsersInsideResourceGroup	The operation that you want to perform. Set the value to ListAscmUsersInsideResourceGroup.
ResourceGroupId	Integer	Yes	7	The ID of the resource set.
CurrentPage	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListAscmUsersInsideResourceGroup
&ResourceGroupId=7
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
```

```
"cost": 22,
"data": [
  {
    "cellphoneNum": "13552062486",
    "default": false,
    "defaultRole": {
      "active": true,
      "code": "ROLE_ORGANIZATION_ADMIN",
      "cuserId": "",
      "default": true,
      "description": "defaultRole.organizationAdmin.description",
      "id": 4,
      "muserId": "",
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.organizationAdmin.name",
      "roleRange": "roleRange.orgAndSubOrgs",
      "roleType": "ROLETYPE_ASCM"
    },
    "deleted": false,
    "displayName": "<name>",
    "email": "<emailAddress>",
    "enableDingTalk": true,
    "enableEmail": true,
    "enableShortMessage": true,
    "id": 5017,
    "lastLoginTime": 1575601343930,
    "loginName": "saizhang",
    "loginPolicy": {
      "cuserId": "",
      "default": true,
      "id": 1,
      "ipRanges": [
        {
          "ipRange": "0.0.0.0/0",
          "loginPolicyId": 1,
          "protocol": "IPV4"
        }
      ],
      "muserId": "",
      "name": "defaultLoginPolicy.name",
```

```
      "rule": "ALLOW",
      "timeRanges": [
        {
          "endTime": "24:00",
          "loginPolicyId": 1,
          "startTime": "00:00"
        }
      ]
    },
    "mobileNationCode": "86",
    "organization": {
      "alias": "lqx-test",
      "cuserId": "",
      "id": 8,
      "internal": false,
      "level": "0.1",
      "muserId": "",
      "name": "lqx-test",
      "parentId": 1,
      "supportRegions": "cn-qingdao-env25-d01"
    },
    "parentPk": "<yourParentPK>",
    "primaryKey": "<yourPrimaryKey>",
    "roles": [
      {
        "active": true,
        "code": "ROLE_ORGANIZATION_ADMIN",
        "default": true,
        "description": "defaultRole.organizationAdmin.description",
        "id": 4,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.organizationAdmin.name",
        "roleRange": "roleRange.orgAndSubOrgs",
        "roleType": "ROLETYPE_ASCM"
      }
    ],
    "status": "ACTIVE",
    "userGroupRoles": [],
    "userRoles": [
      {
```

```
        {
          "active": true,
          "code": "ROLE_ORGANIZATION_ADMIN",
          "cuserId": "",
          "default": true,
          "description": "defaultRole.organizationAdmin.description",
          "id": 4,
          "muserId": "",
          "rAMRole": false,
          "roleLevel": 2147483647,
          "roleName": "defaultRole.organizationAdmin.name",
          "roleRange": "roleRange.orgAndSubOrgs",
          "roleType": "ROLETYPE_ASCM"
        }
      ]
    },
    {
      "cellphoneNum": "13552062486",
      "default": false,
      "defaultRole": {
        "active": true,
        "code": "ROLE_ORGANIZATION_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.organizationAdmin.description",
        "id": 4,
        "muserId": "",
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.organizationAdmin.name",
        "roleRange": "roleRange.orgAndSubOrgs",
        "roleType": "ROLETYPE_ASCM"
      },
      "deleted": false,
      "displayName": "<name>",
      "email": "<emailAddress>",
      "enableDingTalk": true,
      "enableEmail": true,
      "enableShortMessage": true,
      "id": 5023,
      "lastLoginTime": 1575461152779,
```

```
"loginName": "saizhang1",
"loginPolicy": {
  "cuserId": "",
  "default": true,
  "id": 1,
  "ipRanges": [
    {
      "ipRange": "0.0.0.0/0",
      "loginPolicyId": 1,
      "protocol": "IPV4"
    }
  ],
  "muserId": "",
  "name": "defaultLoginPolicy.name",
  "rule": "ALLOW",
  "timeRanges": [
    {
      "endTime": "24:00",
      "loginPolicyId": 1,
      "startTime": "00:00"
    }
  ]
},
"mobileNationCode": "86",
"organization": {
  "alias": "lqx-test",
  "cuserId": "",
  "id": 8,
  "internal": false,
  "level": "0.1",
  "muserId": "",
  "name": "lqx-test",
  "parentId": 1,
  "supportRegions": "cn-qingdao-env25-d01"
},
"parentPk": "1864274920236591",
"primaryKey": "267494175449476273",
"roles": [
  {
    "active": true,
    "code": "ROLE_ORGANIZATION_ADMIN",
```

```
            "default": true,
            "description": "defaultRole.organizationAdmin.description",
            "id": 4,
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.organizationAdmin.name",
            "roleRange": "roleRange.orgAndSubOrgs",
            "roleType": "ROLETYPE_ASCM"
          }
        ],
        "status": "ACTIVE",
        "userGroupRoles": [],
        "userRoles": [
          {
            "active": true,
            "code": "ROLE_ORGANIZATION_ADMIN",
            "cuserId": "",
            "default": true,
            "description": "defaultRole.organizationAdmin.description",
            "id": 4,
            "muserId": "",
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.organizationAdmin.name",
            "roleRange": "roleRange.orgAndSubOrgs",
            "roleType": "ROLETYPE_ASCM"
          }
        ]
      }
    ],
    "message": "success",
    "pageInfo": {
      "currentPage": 1,
      "pageSize": 10,
      "total": 2,
      "totalPage": 1
    },
    "redirect": false,
    "success": true,
    "asapiRequestId": "cce9bda7-0737-43bd-9f6a-9de9ccdbc260"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>22</cost>
<data>
  <cellphoneNum>1355206****</cellphoneNum>
  <default>false</default>
  <defaultRole>
    <active>true</active>
    <code>ROLE_ORGANIZATION_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.organizationAdmin.description</description>
    <id>4</id>
    <muserId/>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.organizationAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </defaultRole>
  <deleted>false</deleted>
  <displayName><name></displayName>
  <email><emailAddress></email>
  <enableDingTalk>true</enableDingTalk>
  <enableEmail>true</enableEmail>
  <enableShortMessage>true</enableShortMessage>
  <id>5017</id>
  <lastLoginTime>1575601343930</lastLoginTime>
  <loginName>saizhang</loginName>
  <loginPolicy>
    <cuserId/>
    <default>true</default>
    <id>1</id>
    <ipRanges>
      <ipRange>0.0.0.0/0</ipRange>
      <loginPolicyId>1</loginPolicyId>
      <protocol>IPV4</protocol>
    </ipRanges>
    <muserId/>
```

```
    <name>defaultLoginPolicy.name</name>
    <rule>ALLOW</rule>
    <timeRanges>
      <endTime>24:00</endTime>
      <loginPolicyId>1</loginPolicyId>
      <startTime>00:00</startTime>
    </timeRanges>
  </loginPolicy>
  <mobileNationCode>86</mobileNationCode>
  <organization>
    <alias>lqx-test</alias>
    <cuserId/>
    <id>8</id>
    <internal>false</internal>
    <level>0.1</level>
    <muserId/>
    <name>lqx-test</name>
    <parentId>1</parentId>
    <supportRegions>cn-qingdao-env25-d01</supportRegions>
  </organization>
  <parentPk><yourParentPK></parentPk>
  <primaryKey><yourPrimaryKey></primaryKey>
  <roles>
    <active>true</active>
    <code>ROLE_ORGANIZATION_ADMIN</code>
    <default>true</default>
    <description>defaultRole.organizationAdmin.description</description>
    <id>4</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.organizationAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </roles>
  <status>ACTIVE</status>
  <userRoles>
    <active>true</active>
    <code>ROLE_ORGANIZATION_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.organizationAdmin.description</description>
```

```
      <id>4</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.organizationAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
</data>
<data>
    <cellphoneNum>1355206****</cellphoneNum>
    <default>false</default>
    <defaultRole>
      <active>true</active>
      <code>ROLE_ORGANIZATION_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.organizationAdmin.description</description>
      <id>4</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.organizationAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </defaultRole>
    <deleted>false</deleted>
    <displayName><name></displayName>
    <email><yourEmailAddress></email>
    <enableDingTalk>true</enableDingTalk>
    <enableEmail>true</enableEmail>
    <enableShortMessage>true</enableShortMessage>
    <id>5023</id>
    <lastLoginTime>1575461152779</lastLoginTime>
    <loginName>saizhang1</loginName>
    <loginPolicy>
      <cuserId/>
      <default>true</default>
      <id>1</id>
      <ipRanges>
        <ipRange>0.0.0.0/0</ipRange>
```

```
        <loginPolicyId>1</loginPolicyId>
        <protocol>IPV4</protocol>
      </ipRanges>
      <muserId/>
      <name>defaultLoginPolicy.name</name>
      <rule>ALLOW</rule>
      <timeRanges>
        <endTime>24:00</endTime>
        <loginPolicyId>1</loginPolicyId>
        <startTime>00:00</startTime>
      </timeRanges>
    </loginPolicy>
    <mobileNationCode>86</mobileNationCode>
    <organization>
      <alias>lqx-test</alias>
      <cuserId/>
      <id>8</id>
      <internal>false</internal>
      <level>0.1</level>
      <muserId/>
      <name>lqx-test</name>
      <parentId>1</parentId>
      <supportRegions>cn-qingdao-env25-d01</supportRegions>
    </organization>
    <parentPk><yourParentPK></parentPk>
    <primaryKey><yourPrimaryKey></primaryKey>
    <roles>
      <active>true</active>
      <code>ROLE_ORGANIZATION_ADMIN</code>
      <default>true</default>
      <description>defaultRole.organizationAdmin.description</description>
      <id>4</id>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.organizationAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </roles>
    <status>ACTIVE</status>
    <userRoles>
```

```
      <active>true</active>
      <code>ROLE_ORGANIZATION_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.organizationAdmin.description</description>
      <id>4</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.organizationAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
</data>
<message>success</message>
<pageInfo>
    <currentPage>1</currentPage>
    <pageSize>10</pageSize>
    <total>2</total>
    <totalPage>1</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>cce9bda7-0737-43bd-9f6a-9de9ccdbc260</asapiRequestId>
```


1.2.4.3. Roles

1.2.4.3.1. SetDefaultRole

You can call this operation to set a default role.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetDefaultRole	The operation that you want to perform. Set the value to SetDefaultRole.
LoginName	String	No	testu	The logon name.

Parameter	Type	Required	Example	Description
RoleId	String	No	5	The ID of the role.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=SetDefaultRole
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
"code": "200",
"cost": 174,
"message": "success",
"redirect": false,
"success": true,
"asapiRequestId": "82b5ab6e-c5a2-4c67-b694-efbdddd968f6"
}
```

1.2.4.3.2. ListRoles

You can call this operation to query the list of available roles of a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListRoles	The operation that you want to perform. Set the value to ListRoles.
CurrentPage	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListRoles
&<Common request parameters>
```

- Sample success responses in JSON format


```
{
  "code": "200",
  "cost": 42,
  "data": [
    {
      "active": true,
      "code": "ROLE_OPERATION_ADMIN",
      "default": true,
      "description": "defaultRole.operationAdmin.description",
      "id": 1,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.operationAdmin.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_RESOURCE_USER",
      "default": true,
      "description": "defaultRole.resourceUser.description",
      "id": 2,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.resourceUser.name",
      "roleRange": "roleRange.resourceSet",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_PLATFORM_ADMIN",
      "default": true,
      "description": "defaultRole.platformAdmin.description",
      "id": 3,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.platformAdmin.name",
```

```
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_ORGANIZATION_ADMIN",
      "default": true,
      "description": "defaultRole.organizationAdmin.description",
      "id": 4,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.organizationAdmin.name",
      "roleRange": "roleRange.orgAndSubOrgs",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 5
    },
    {
      "active": true,
      "code": "ROLE_RESOURCE_SET_ADMIN",
      "default": true,
      "description": "defaultRole.resourceSetAdmin.description",
      "id": 5,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.resourceSetAdmin.name",
      "roleRange": "roleRange.resourceSet",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_GLOBAL_RESOURCE_AUDITOR",
      "default": true,
      "description": "defaultRole.globalResourceAuditor.description",
      "id": 6,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.globalResourceAuditor.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
```

```
    "userCount": 1
  },
  {
    "active": true,
    "code": "ROLE_YUNDUN_ORG_SEC_ADMIN",
    "default": true,
    "description": "defaultRole.yundunOrgSecAdmin.description",
    "id": 8,
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.yundunOrgSecAdmin.name",
    "roleRange": "roleRange.orgAndSubOrgs",
    "roleType": "ROLETYPE_ASCM",
    "userCount": 1
  },
  {
    "active": true,
    "code": "ROLE_YUNDUN_SEC_AUDITOR",
    "default": true,
    "description": "defaultRole.yundunSecAuditor.description",
    "id": 9,
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.yundunSecAuditor.name",
    "roleRange": "roleRange.allOrganizations",
    "roleType": "ROLETYPE_ASCM",
    "userCount": 0
  },
  {
    "active": true,
    "code": "ROLE_YUNDUN_SECSYSCONF_ADMIN",
    "default": true,
    "description": "defaultRole.yundunSecSysConfAdmin.description",
    "id": 10,
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.yundunSecSysConfAdmin.name",
    "roleRange": "roleRange.allOrganizations",
    "roleType": "ROLETYPE_ASCM",
    "userCount": 1
  }
```

```
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_GLOBALORG_ADMIN",
      "default": true,
      "description": "defaultRole.yundunGlobalOrgAdmin.description",
      "id": 11,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunGlobalOrgAdmin.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    }
  ],
  "message": "success",
  "pageInfo": {
    "currentPage": 1,
    "pageSize": 10,
    "total": 14,
    "totalPage": 2
  },
  "redirect": false,
  "success": true,
  "asapiRequestId": "561a3c30-b1f9-4dcf-a4fa-f92218db6260"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>42</cost>
<data>
  <active>true</active>
  <code>ROLE_OPERATION_ADMIN</code>
  <default>true</default>
  <description>defaultRole.operationAdmin.description</description>
  <id>1</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.operationAdmin.name</roleName>
  <roleRange>roleRange.allOrganizations</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
```

```
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_RESOURCE_USER</code>
    <default>true</default>
    <description>defaultRole.resourceUser.description</description>
    <id>2</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.resourceUser.name</roleName>
    <roleRange>roleRange.resourceSet</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_PLATFORM_ADMIN</code>
    <default>true</default>
    <description>defaultRole.platformAdmin.description</description>
    <id>3</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.platformAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_ORGANIZATION_ADMIN</code>
    <default>true</default>
    <description>defaultRole.organizationAdmin.description</description>
    <id>4</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.organizationAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>5</userCount>
</data>
```

```
<data>
  <active>true</active>
  <code>ROLE_RESOURCE_SET_ADMIN</code>
  <default>true</default>
  <description>defaultRole.resourceSetAdmin.description</description>
  <id>5</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.resourceSetAdmin.name</roleName>
  <roleRange>roleRange.resourceSet</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
  <userCount>1</userCount>
</data>
<data>
  <active>true</active>
  <code>ROLE_GLOBAL_RESOURCE_AUDITOR</code>
  <default>true</default>
  <description>defaultRole.globalResourceAuditor.description</description>
  <id>6</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.globalResourceAuditor.name</roleName>
  <roleRange>roleRange.allOrganizations</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
  <userCount>1</userCount>
</data>
<data>
  <active>true</active>
  <code>ROLE_YUNDUN_ORG_SEC_ADMIN</code>
  <default>true</default>
  <description>defaultRole.yundunOrgSecAdmin.description</description>
  <id>8</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.yundunOrgSecAdmin.name</roleName>
  <roleRange>roleRange.orgAndSubOrgs</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
  <userCount>1</userCount>
</data>
<data>
  <active>true</active>
```

```
    <active>true</active>
    <code>ROLE_YUNDUN_SEC_AUDITOR</code>
    <default>true</default>
    <description>defaultRole.yundunSecAuditor.description</description>
    <id>9</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunSecAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>0</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_SECSYSCONF_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunSecSysConfAdmin.description</description>
    <id>10</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunSecSysConfAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_GLOBALORG_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunGlobalOrgAdmin.description</description>
    <id>11</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunGlobalOrgAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<message>success</message>
<pageInfo>
    <currentPage>1</currentPage>
```

```
    <pageSize>10</pageSize>
    <total>14</total>
    <totalPage>2</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>561a3c30-b1f9-4dcf-a4fa-f92218db6260</asapiRequestId>
```

1.2.4.3.3. ListActiveRoles

You can call this operation to query the list of roles that can be bound.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListActiveRoles	The operation that you want to perform. Set the value to ListActiveRoles.
IncludeRAMRoles	Boolean	No	false	Specifies whether to include RAM roles.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListActiveRoles
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 42,
  "data": [
    {
      "active": true,
      "code": "ROLE_RESOURCE_USER",
      "default": true,
      "description": "defaultRole.resourceUser.description",
      "id": 2,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.resourceUser.name"
```

```
    "roleName": "defaultRole.resourceUser.name",
    "roleRange": "roleRange.resourceSet",
    "roleType": "ROLETYPE_ASCM",
    "userCount": 1
  },
  {
    "active": true,
    "code": "ROLE_ORGANIZATION_ADMIN",
    "default": true,
    "description": "defaultRole.organizationAdmin.description",
    "id": 4,
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.organizationAdmin.name",
    "roleRange": "roleRange.orgAndSubOrgs",
    "roleType": "ROLETYPE_ASCM",
    "userCount": 5
  },
  {
    "active": true,
    "code": "ROLE_RESOURCE_SET_ADMIN",
    "default": true,
    "description": "defaultRole.resourceSetAdmin.description",
    "id": 5,
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.resourceSetAdmin.name",
    "roleRange": "roleRange.resourceSet",
    "roleType": "ROLETYPE_ASCM",
    "userCount": 1
  },
  {
    "active": true,
    "code": "ROLE_GLOBAL_RESOURCE_AUDITOR",
    "default": true,
    "description": "defaultRole.globalResourceAuditor.description",
    "id": 6,
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.globalResourceAuditor.name",
    "roleRange": "roleRange.allOrganizations",
```

```
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_ORG_SEC_ADMIN",
      "default": true,
      "description": "defaultRole.yundunOrgSecAdmin.description",
      "id": 8,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunOrgSecAdmin.name",
      "roleRange": "roleRange.orgAndSubOrgs",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_SEC_AUDITOR",
      "default": true,
      "description": "defaultRole.yundunSecAuditor.description",
      "id": 9,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunSecAuditor.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 0
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_SECSYSCONF_ADMIN",
      "default": true,
      "description": "defaultRole.yundunSecSysConfAdmin.description",
      "id": 10,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunSecSysConfAdmin.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
```

```
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_GLOBALORG_ADMIN",
      "default": true,
      "description": "defaultRole.yundunGlobalOrgAdmin.description",
      "id": 11,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunGlobalOrgAdmin.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 1
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_PLATSEC_ADMIN",
      "default": true,
      "description": "defaultRole.yundunPlatSecAdmin.description",
      "id": 12,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunPlatSecAdmin.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 2
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_GLOBALORGSEC_AUDITOR",
      "default": true,
      "description": "defaultRole.yundunGlobalOrgSecAuditor.description",
      "id": 13,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunGlobalOrgSecAuditor.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM",
      "userCount": 0
    },
```

```
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSEC_AUDITOR",
        "default": true,
        "description": "defaultRole.yundunPlatSecAuditor.description",
        "id": 14,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecAuditor.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM",
        "userCount": 0
      },
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSECCONF_ADMIN",
        "default": true,
        "description": "defaultRole.yundunPlatSecConfAdmin.description",
        "id": 15,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecConfAdmin.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM",
        "userCount": 1
      }
    ],
    "message": "success",
    "pageInfo": {
      "currentPage": 1,
      "pageSize": 12,
      "total": 12,
      "totalPage": 1
    },
    "redirect": false,
    "success": true,
    "asapiRequestId": "88029f4d-30b1-4862-8dfd-3185c499d88b"
}
```


- Sample success responses in XML format


```
<code>200</code>
```

```
<cost>42</cost>
<data>
  <active>true</active>
  <code>ROLE_RESOURCE_USER</code>
  <default>true</default>
  <description>defaultRole.resourceUser.description</description>
  <id>2</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.resourceUser.name</roleName>
  <roleRange>roleRange.resourceSet</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
  <userCount>1</userCount>
</data>
<data>
  <active>true</active>
  <code>ROLE_ORGANIZATION_ADMIN</code>
  <default>true</default>
  <description>defaultRole.organizationAdmin.description</description>
  <id>4</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.organizationAdmin.name</roleName>
  <roleRange>roleRange.orgAndSubOrgs</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
  <userCount>5</userCount>
</data>
<data>
  <active>true</active>
  <code>ROLE_RESOURCE_SET_ADMIN</code>
  <default>true</default>
  <description>defaultRole.resourceSetAdmin.description</description>
  <id>5</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.resourceSetAdmin.name</roleName>
  <roleRange>roleRange.resourceSet</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
  <userCount>1</userCount>
</data>
<data>
```

```
    <active>true</active>
    <code>ROLE_GLOBAL_RESOURCE_AUDITOR</code>
    <default>true</default>
    <description>defaultRole.globalResourceAuditor.description</description>
    <id>6</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.globalResourceAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_ORG_SEC_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunOrgSecAdmin.description</description>
    <id>8</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunOrgSecAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_SEC_AUDITOR</code>
    <default>true</default>
    <description>defaultRole.yundunSecAuditor.description</description>
    <id>9</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunSecAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>0</userCount>
</data>
<data>
    <active>true</active>
```

```
    <code>ROLE_YUNDUN_SECSYSCONF_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunSecSysConfAdmin.description</description>
    <id>10</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunSecSysConfAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_GLOBALORG_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunGlobalOrgAdmin.description</description>
    <id>11</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunGlobalOrgAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_PLATSEC_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunPlatSecAdmin.description</description>
    <id>12</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunPlatSecAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>2</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_GLOBALORGSEC_AUDITOR</code>
    <default>true</default>
```

```
    <description>defaultRole.yundunGlobalOrgSecAuditor.description</description>
    <id>13</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunGlobalOrgSecAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>0</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_PLATSEC_AUDITOR</code>
    <default>true</default>
    <description>defaultRole.yundunPlatSecAuditor.description</description>
    <id>14</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunPlatSecAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>0</userCount>
</data>
<data>
    <active>true</active>
    <code>ROLE_YUNDUN_PLATSECCONF_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunPlatSecConfAdmin.description</description>
    <id>15</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunPlatSecConfAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
    <userCount>1</userCount>
</data>
<message>success</message>
<pageInfo>
    <currentPage>1</currentPage>
    <pageSize>12</pageSize>
    <total>12</total>
    <totalPage>1</totalPage>
```

```
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>88029f4d-30b1-4862-8dfd-3185c499d88b</asapiRequestId>
```

1.2.4.4. Users

1.2.4.4.1. AddUser

You can call this operation to create a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AddUser	The operation that you want to perform. Set the value to AddUser.
CellphoneNum	String	Yes	10000000000	The mobile number of the user.
DisplayName	String	Yes	testuser	The display name of the user.
Email	String	Yes	a@a.com	The email address of the user.
LoginName	String	Yes	testu	The logon name of the user.
MobileNationCode	String	Yes	86	The country code of the user.
OrganizationId	Integer	Yes	5	The ID of the organization.
roleIdList	JSON	No	{"RoleIdList": ["5", "6", "8"]}	The list of role IDs.
LoginPolicyId	Integer	No	1	The ID of the logon policy.
TelephoneNum	String	No	00000000	The telephone number of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=AddUser
&CellphoneNum=10000000000
&DisplayName=testuser
&Email=a@a.com
&LoginName=testu
&MobileNationCode=86
&OrganizationId=5
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 1370,
  "data": {
    "cellphoneNum": "1870000****",
    "default": false,
    "deleted": false,
    "displayName": "testuser",
    "email": "a@a.com",
    "enableDingTalk": true,
    "enableEmail": true,
    "enableShortMessage": true,
    "id": 5031,
    "loginName": "testu",
    "loginPolicy": {
      "cuserId": "",
      "default": true,
      "id": 1,
      "ipRanges": [
        {
          "ipRange": "0.0.0.0/0",
          "loginPolicyId": 1,
          "protocol": "IPV4"
        }
      ],
      "muserId": "",
      "name": "defaultLoginPolicy.name",
      "rule": "ALLOW",
      "timeRanges": [
        {
          "endTime": "24:00"
```

```
            "endTime": "24:00",
            "loginPolicyId": 1,
            "startTime": "00:00"
          }
        ]
      },
      "mobileNationCode": "86",
      "organization": {
        "alias": "lqx-test",
        "cuserId": "",
        "id": 8,
        "internal": false,
        "level": "0.1",
        "muserId": "",
        "name": "lqx-test",
        "parentId": 1,
        "supportRegions": "cn-qingdao-env25-d01"
      },
"parentPk": "<yourParenetPK>",
      "primaryKey": "<yourPrimaryKey>",
      "roles": [
        {
          "active": true,
          "code": "ROLE_RESOURCE_SET_ADMIN",
          "default": true,
          "description": "defaultRole.resourceSetAdmin.description",
          "id": 5,
          "rAMRole": false,
          "roleLevel": 2147483647,
          "roleName": "defaultRole.resourceSetAdmin.name",
          "roleRange": "roleRange.resourceSet",
          "roleType": "ROLETYPE_ASCM"
        },
        {
          "active": true,
          "code": "ROLE_YUNDUN_ORG_SEC_ADMIN",
          "default": true,
          "description": "defaultRole.yundunOrgSecAdmin.description",
          "id": 8,
          "rAMRole": false,
          "roleLevel": 2147483647,
```

```
            "roleName": "defaultRole.yundunOrgSecAdmin.name",
            "roleRange": "roleRange.orgAndSubOrgs",
            "roleType": "ROLETYPE_ASCM"
        },
        {
            "active": true,
            "code": "ROLE_GLOBAL_RESOURCE_AUDITOR",
            "default": true,
            "description": "defaultRole.globalResourceAuditor.description",
            "id": 6,
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.globalResourceAuditor.name",
            "roleRange": "roleRange.allOrganizations",
            "roleType": "ROLETYPE_ASCM"
        }
    ],
    "status": "ACTIVE",
    "telephoneNum": "00000000",
    "userGroupRoles": [],
    "userRoles": [
        {
            "active": true,
            "code": "ROLE_RESOURCE_SET_ADMIN",
            "cuserId": "",
            "default": true,
            "description": "defaultRole.resourceSetAdmin.description",
            "id": 5,
            "muserId": "",
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.resourceSetAdmin.name",
            "roleRange": "roleRange.resourceSet",
            "roleType": "ROLETYPE_ASCM"
        },
        {
            "active": true,
            "code": "ROLE_GLOBAL_RESOURCE_AUDITOR",
            "cuserId": "",
            "default": true,
            "description": "defaultRole.globalResourceAuditor.description",
```

```
            "id": 6,
            "muserId": "",
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.globalResourceAuditor.name",
            "roleRange": "roleRange.allOrganizations",
            "roleType": "ROLETYPE_ASCM"
          },
          {
            "active": true,
            "code": "ROLE_YUNDUN_ORG_SEC_ADMIN",
            "cuserId": "",
            "default": true,
            "description": "defaultRole.yundunOrgSecAdmin.description",
            "id": 8,
            "muserId": "",
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.yundunOrgSecAdmin.name",
            "roleRange": "roleRange.orgAndSubOrgs",
            "roleType": "ROLETYPE_ASCM"
          }
        ]
    },
    "message": "success",
    "redirect": false,
    "success": true,
    "asapiRequestId": "8e1ae1fe-a603-4698-bc33-deb16629efde"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>1370</cost>
<data>
    <cellphoneNum>18700000000</cellphoneNum>
    <default>false</default>
    <deleted>false</deleted>
    <displayName>testuser</displayName>
    <email>a@a.com</email>
    <enableDingTalk>true</enableDingTalk>
```

```
<enableEmail>true</enableEmail>
<enableShortMessage>true</enableShortMessage>
<id>5031</id>
<loginName>testu</loginName>
<loginPolicy>
  <cuserId/>
  <default>true</default>
  <id>1</id>
  <ipRanges>
    <ipRange>0.0.0.0/0</ipRange>
    <loginPolicyId>1</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <muserId/>
  <name>defaultLoginPolicy.name</name>
  <rule>ALLOW</rule>
  <timeRanges>
    <endTime>24:00</endTime>
    <loginPolicyId>1</loginPolicyId>
    <startTime>00:00</startTime>
  </timeRanges>
</loginPolicy>
<mobileNationCode>86</mobileNationCode>
<organization>
  <alias>lqx-test</alias>
  <cuserId/>
  <id>8</id>
  <internal>false</internal>
  <level>0.1</level>
  <muserId/>
  <name>lqx-test</name>
  <parentId>1</parentId>
  <supportRegions>cn-qingdao-env25-d01</supportRegions>
</organization>
<parentPk>1864274920236591</parentPk>
<primaryKey>218016875699716713</primaryKey>
<roles>
  <active>true</active>
  <code>ROLE_RESOURCE_SET_ADMIN</code>
  <default>true</default>
  <description>defaultRole.resourceSetAdmin.description</description>
```

```
    <id>5</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.resourceSetAdmin.name</roleName>
    <roleRange>roleRange.resourceSet</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </roles>
  <roles>
    <active>true</active>
    <code>ROLE_YUNDUN_ORG_SEC_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunOrgSecAdmin.description</description>
    <id>8</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunOrgSecAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </roles>
  <roles>
    <active>true</active>
    <code>ROLE_GLOBAL_RESOURCE_AUDITOR</code>
    <default>true</default>
    <description>defaultRole.globalResourceAuditor.description</description>
    <id>6</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.globalResourceAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </roles>
  <status>ACTIVE</status>
  <telephoneNum>00000000</telephoneNum>
  <userRoles>
    <active>true</active>
    <code>ROLE_RESOURCE_SET_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.resourceSetAdmin.description</description>
    <id>5</id>
    <muserId/>
```

```
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.resourceSetAdmin.name</roleName>
      <roleRange>roleRange.resourceSet</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
    <userRoles>
      <active>true</active>
      <code>ROLE_GLOBAL_RESOURCE_AUDITOR</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.globalResourceAuditor.description</description>
      <id>6</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.globalResourceAuditor.name</roleName>
      <roleRange>roleRange.allOrganizations</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
    <userRoles>
      <active>true</active>
      <code>ROLE_YUNDUN_ORG_SEC_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.yundunOrgSecAdmin.description</description>
      <id>8</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.yundunOrgSecAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
  </data>
  <message>success</message>
  <redirect>false</redirect>
  <success>true</success>
  <asapiRequestId>8e1ae1fe-a603-4698-bc33-deb16629efde</asapiRequestId>
```

1.2.4.4.2. AddRoleToUser

You can call this operation to bind a role to a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AddRoleToUser	The operation that you want to perform. Set the value to AddRoleToUser.
LoginName	String	Yes	test	The logon name.
RoleId	Integer	Yes	3	The ID of the role.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=AddRoleToUser
&LoginName=testu
&RoleId=3
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 174,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "82b5ab6e-c5a2-4c67-b694-efbdddd968f6"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>174</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>82b5ab6e-c5a2-4c67-b694-efbdddd968f6</asapiRequestId>
```

1.2.4.4.3. UpdatePassword

You can call this operation to change a password.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdatePassword	The operation that you want to perform. Set the value to UpdatePassword.
NewPassword	String	Yes	!QAZ2wsx	The new password.
OldPassword	String	Yes	#EDC4rfv	The original password.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=UpdatePassword
&NewPassword=!QAZ2wsx
&OldPassword=#EDC4rfv
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 174,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "82b5ab6e-c5a2-4c67-b694-efbdddd968f6"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>174</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>82b5ab6e-c5a2-4c67-b694-efbdddd968f6</asapiRequestId>
```

1.2.4.4.4. ResetPasswordByLoginName

You can call this operation to reset a user password.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ResetPasswordByLoginName	The operation that you want to perform. Set the value to ResetPasswordByLoginName.
LoginName	String	Yes	testu	The logon name of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ResetPasswordByLoginName
&LoginName=testu
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 174,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "82b5ab6e-c5a2-4c67-b694-efbdddd968f6"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>174</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>82b5ab6e-c5a2-4c67-b694-efbdddd968f6</asapiRequestId>
```

1.2.4.4.5. GetLoginPolicyByLoginName

You can call this operation to query the logon policy that is bound to a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetLoginPolicyByLoginName	The operation that you want to perform. Set the value to GetLoginPolicyByLoginName.
LoginName	String	Yes	testu	The logon name of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetLoginPolicyByLoginName
&LoginName=testu
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 7,
  "data": {
    "cuserId": "",
    "default": false,
    "description": "just for test",
    "id": 1001,
    "ipRanges": [
      {
        "ipRange": "1.1.1.0/24",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      },
      {
        "ipRange": "2.2.2.5/24",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      }
    ],
    "muserId": "",
    "name": "testIp",
    "rule": "DENY",
```

```
      "timeRanges": [
        {
          "endTime": "13:35",
          "loginPolicyId": 1001,
          "startTime": "11:00"
        },
        {
          "endTime": "12:30",
          "loginPolicyId": 1001,
          "startTime": "12:00"
        }
      ]
    },
    "message": "success",
    "redirect": false,
    "success": true,
    "asapiRequestId": "c21470ee-7f4d-4282-8be7-959755fe2015"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>7</cost>
<data>
  <cuserId/>
  <default>false</default>
  <description>just for test</description>
  <id>1001</id>
  <ipRanges>
    <ipRange>1.1.1.0/24</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <ipRanges>
    <ipRange>2.2.2.5/24</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <muserId/>
  <name>testIp</name>
  <rule>DENY</rule>
  <timeRanges>
    <endTime>13:35</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>11:00</startTime>
  </timeRanges>
  <timeRanges>
    <endTime>12:30</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>12:00</startTime>
  </timeRanges>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>c21470ee-7f4d-4282-8be7-959755fe2015</asapiRequestId>
```

1.2.4.4.6. RemoveUserByLoginName

You can call this operation to remove a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveUserByLoginName	The operation that you want to perform. Set the value to RemoveUserByLoginName.
LoginName	String	Yes	testu	The logon name of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=RemoveUserByLoginName
&LoginName=testu
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 626,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "cb109b93-ccfa-43b2-9114-4177c346bb7b"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>626</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>cb109b93-ccfa-43b2-9114-4177c346bb7b</asapiRequestId>
```

1.2.4.4.7. GetUserInfo

You can call this operation to query the detailed information of a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetUserInfo	The operation that you want to perform. Set the value to GetUserInfo.
LoginName	String	No	testu	The logon name of the user.
UserAccessKeyId	String	No	test	The AccessKey ID of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetUserInfo
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 1370,
  "data": {
    "accessKeys": [
      {
        "accesskeyId": "**************",
        "accesskeySecret": "********************",
        "ctime": 1575670917000,
        "cuserId": "admin",
        "id": 34,
        "region": "cn-qingdao-env25-d01",
        "status": "ACTIVE"
      }
    ],
    "cellphoneNum": "18700000000",
    "default": false,
    "deleted": false,
    "displayName": "testuser",
    "email": "a@a.com",
    "enableDingTalk": true,
    "enableEmail": true,
    "enableShortMessage": true,
```

```
"id": 5031,
"loginName": "testu",
"loginPolicy": {
  "cuserId": "",
  "default": true,
  "id": 1,
  "ipRanges": [
    {
      "ipRange": "0.0.0.0/0",
      "loginPolicyId": 1,
      "protocol": "IPV4"
    }
  ],
  "muserId": "",
  "name": "defaultLoginPolicy.name",
  "rule": "ALLOW",
  "timeRanges": [
    {
      "endTime": "24:00",
      "loginPolicyId": 1,
      "startTime": "00:00"
    }
  ]
},
"mobileNationCode": "86",
"organization": {
  "alias": "lqx-test",
  "cuserId": "",
  "id": 8,
  "internal": false,
  "level": "0.1",
  "muserId": "",
  "name": "lqx-test",
  "parentId": 1,
  "supportRegions": "cn-qingdao-env25-d01"
},
"parentPk": "1864274920236591",
"primaryKey": "218016875699716713",
"roles": [
  {
    "active": true,
```

```
            "code": "ROLE_RESOURCE_SET_ADMIN",
            "default": true,
            "description": "defaultRole.resourceSetAdmin.description",
            "id": 5,
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.resourceSetAdmin.name",
            "roleRange": "roleRange.resourceSet",
            "roleType": "ROLETYPE_ASCM"
        },
        {
            "active": true,
            "code": "ROLE_YUNDUN_ORG_SEC_ADMIN",
            "default": true,
            "description": "defaultRole.yundunOrgSecAdmin.description",
            "id": 8,
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.yundunOrgSecAdmin.name",
            "roleRange": "roleRange.orgAndSubOrgs",
            "roleType": "ROLETYPE_ASCM"
        },
        {
            "active": true,
            "code": "ROLE_GLOBAL_RESOURCE_AUDITOR",
            "default": true,
            "description": "defaultRole.globalResourceAuditor.description",
            "id": 6,
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.globalResourceAuditor.name",
            "roleRange": "roleRange.allOrganizations",
            "roleType": "ROLETYPE_ASCM"
        }
    ],
    "status": "ACTIVE",
    "telephoneNum": "00000000",
    "userGroupRoles": [],
    "userRoles": [
        {
            "active": true,
```

```
    "active": true,
    "code": "ROLE_RESOURCE_SET_ADMIN",
    "cuserId": "",
    "default": true,
    "description": "defaultRole.resourceSetAdmin.description",
    "id": 5,
    "muserId": "",
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.resourceSetAdmin.name",
    "roleRange": "roleRange.resourceSet",
    "roleType": "ROLETYPE_ASCM"
  },
  {
    "active": true,
    "code": "ROLE_GLOBAL_RESOURCE_AUDITOR",
    "cuserId": "",
    "default": true,
    "description": "defaultRole.globalResourceAuditor.description",
    "id": 6,
    "muserId": "",
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.globalResourceAuditor.name",
    "roleRange": "roleRange.allOrganizations",
    "roleType": "ROLETYPE_ASCM"
  },
  {
    "active": true,
    "code": "ROLE_YUNDUN_ORG_SEC_ADMIN",
    "cuserId": "",
    "default": true,
    "description": "defaultRole.yundunOrgSecAdmin.description",
    "id": 8,
    "muserId": "",
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.yundunOrgSecAdmin.name",
    "roleRange": "roleRange.orgAndSubOrgs",
    "roleType": "ROLETYPE_ASCM"
  }
```

```
    ]
  },
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "8e1ae1fe-a603-4698-bc33-deb16629efde"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>1370</cost>
<data>
  <accessKeys>
    <accesskeyId>*************</accesskeyId>
    <accesskeySecret>******************</accesskeySecret>
    <ctime>1575670917000</ctime>
    <cuserId>admin</cuserId>
    <id>34</id>
    <region>cn-qingdao-env25-d01</region>
    <status>ACTIVE</status>
  </accessKeys>
  <cellphoneNum>18700000000</cellphoneNum>
  <default>false</default>
  <deleted>false</deleted>
  <displayName>testuser</displayName>
  <email>a@a.com</email>
  <enableDingTalk>true</enableDingTalk>
  <enableEmail>true</enableEmail>
  <enableShortMessage>true</enableShortMessage>
  <id>5031</id>
  <loginName>testu</loginName>
  <loginPolicy>
    <cuserId/>
    <default>true</default>
    <id>1</id>
    <ipRanges>
      <ipRange>0.0.0.0/0</ipRange>
      <loginPolicyId>1</loginPolicyId>
      <protocol>IPV4</protocol>
    </ipRanges>
    <muserId/>
```

```
    <name>defaultLoginPolicy.name</name>
    <rule>ALLOW</rule>
    <timeRanges>
      <endTime>24:00</endTime>
      <loginPolicyId>1</loginPolicyId>
      <startTime>00:00</startTime>
    </timeRanges>
</loginPolicy>
<mobileNationCode>86</mobileNationCode>
<organization>
    <alias>lqx-test</alias>
    <cuserId/>
    <id>8</id>
    <internal>false</internal>
    <level>0.1</level>
    <muserId/>
    <name>lqx-test</name>
    <parentId>1</parentId>
    <supportRegions>cn-qingdao-env25-d01</supportRegions>
</organization>
<parentPk>1864274920236591</parentPk>
<primaryKey>218016875699716713</primaryKey>
<roles>
    <active>true</active>
    <code>ROLE_RESOURCE_SET_ADMIN</code>
    <default>true</default>
    <description>defaultRole.resourceSetAdmin.description</description>
    <id>5</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.resourceSetAdmin.name</roleName>
    <roleRange>roleRange.resourceSet</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
</roles>
<roles>
    <active>true</active>
    <code>ROLE_YUNDUN_ORG_SEC_ADMIN</code>
    <default>true</default>
    <description>defaultRole.yundunOrgSecAdmin.description</description>
    <id>8</id>
    <rAMRole>false</rAMRole>
```

```
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunOrgSecAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </roles>
  <roles>
    <active>true</active>
    <code>ROLE_GLOBAL_RESOURCE_AUDITOR</code>
    <default>true</default>
    <description>defaultRole.globalResourceAuditor.description</description>
    <id>6</id>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.globalResourceAuditor.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </roles>
  <status>ACTIVE</status>
  <telephoneNum>00000000</telephoneNum>
  <userRoles>
    <active>true</active>
    <code>ROLE_RESOURCE_SET_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.resourceSetAdmin.description</description>
    <id>5</id>
    <muserId/>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.resourceSetAdmin.name</roleName>
    <roleRange>roleRange.resourceSet</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </userRoles>
  <userRoles>
    <active>true</active>
    <code>ROLE_GLOBAL_RESOURCE_AUDITOR</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.globalResourceAuditor.description</description>
    <id>6</id>
```

```
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.globalResourceAuditor.name</roleName>
      <roleRange>roleRange.allOrganizations</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
    <userRoles>
      <active>true</active>
      <code>ROLE_YUNDUN_ORG_SEC_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.yundunOrgSecAdmin.description</description>
      <id>8</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.yundunOrgSecAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>8e1ae1fe-a603-4698-bc33-deb16629efde</asapiRequestId>
```


1.2.4.4.8. ListUsers

You can call this operation to query the list of available users.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListUsers	The operation that you want to perform. Set the value to ListUsers.
LoginName	String	No	test	The logon name of the user.

Parameter	Type	Required	Example	Description
RoleId	Integer	No	2	The ID of the role.
OrganizationId	Integer	No	3	The ID of the organization.
CurrentPage	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListUsers
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 23,
  "data": [
    {
      "cellphoneNum": "18700993233",
      "default": false,
      "defaultRole": {
        "active": true,
        "code": "ROLE_ORGANIZATION_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.organizationAdmin.description",
        "id": 4,
        "muserId": "",
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.organizationAdmin.name",
        "roleRange": "roleRange.orgAndSubOrgs",
        "roleType": "ROLETYPE_ASCM"
      },
      "deleted": false,
```

```
"displayName": "holouser",
"email": "holouser@ascm.com",
"enableDingTalk": true,
"enableEmail": true,
"enableShortMessage": true,
"id": 5030,
"lastLoginTime": 1575599944661,
"loginName": "holouser",
"loginPolicy": {
  "cuserId": "",
  "default": true,
  "id": 1,
  "ipRanges": [
    {
      "ipRange": "0.0.0.0/0",
      "loginPolicyId": 1,
      "protocol": "IPV4"
    }
  ],
  "muserId": "",
  "name": "defaultLoginPolicy.name",
  "rule": "ALLOW",
  "timeRanges": [
    {
      "endTime": "24:00",
      "loginPolicyId": 1,
      "startTime": "00:00"
    }
  ]
},
"mobileNationCode": "86",
"organization": {
  "alias": "holotest",
  "cuserId": "",
  "id": 14,
  "internal": false,
  "level": "0.1",
  "muserId": "",
  "name": "holotest",
  "parentId": 1,
```

```
      "supportRegions": "cn-qingdao-env25-d01"
    },
    "parentPk": "1035075598054324",
    "primaryKey": "233913275599630622",
    "roles": [
      {
        "active": true,
        "code": "ROLE_ORGANIZATION_ADMIN",
        "default": true,
        "description": "defaultRole.organizationAdmin.description",
        "id": 4,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.organizationAdmin.name",
        "roleRange": "roleRange.orgAndSubOrgs",
        "roleType": "ROLETYPE_ASCM"
      },
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSECCONF_ADMIN",
        "default": true,
        "description": "defaultRole.yundunPlatSecConfAdmin.description",
        "id": 15,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecConfAdmin.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM"
      },
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSEC_ADMIN",
        "default": true,
        "description": "defaultRole.yundunPlatSecAdmin.description",
        "id": 12,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecAdmin.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM"
      }
```

```
    ],
    "status": "ACTIVE",
    "userGroupRoles": [],
    "userRoles": [
      {
        "active": true,
        "code": "ROLE_ORGANIZATION_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.organizationAdmin.description",
        "id": 4,
        "muserId": "",
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.organizationAdmin.name",
        "roleRange": "roleRange.orgAndSubOrgs",
        "roleType": "ROLETYPE_ASCM"
      },
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSEC_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.yundunPlatSecAdmin.description",
        "id": 12,
        "muserId": "",
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecAdmin.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM"
      },
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSECCONF_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.yundunPlatSecConfAdmin.description",
        "id": 15,
        "muserId": "",
        "rAMRole": false,
```

```
            "roleLevel": 2147483647,
            "roleName": "defaultRole.yundunPlatSecConfAdmin.name",
            "roleRange": "roleRange.allOrganizations",
            "roleType": "ROLETYPE_ASCM"
          }
        ]
      }
  ],
  "message": "success",
  "pageInfo": {
    "currentPage": 1,
    "pageSize": 10,
    "total": 1,
    "totalPage": 1
  },
  "redirect": false,
  "success": true,
  "asapiRequestId": "2d3aaa3d-b6c6-4700-8b19-2fae40a86a46"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>23</cost>
<data>
  <cellphoneNum>18700993233</cellphoneNum>
  <default>false</default>
  <defaultRole>
    <active>true</active>
    <code>ROLE_ORGANIZATION_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.organizationAdmin.description</description>
    <id>4</id>
    <muserId/>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.organizationAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </defaultRole>
```

```
<deleted>false</deleted>
<displayName>holouser</displayName>
<email>holouser@ascm.com</email>
<enableDingTalk>true</enableDingTalk>
<enableEmail>true</enableEmail>
<enableShortMessage>true</enableShortMessage>
<id>5030</id>
<lastLoginTime>1575599944661</lastLoginTime>
<loginName>holouser</loginName>
<loginPolicy>
  <cuserId/>
  <default>true</default>
  <id>1</id>
  <ipRanges>
    <ipRange>0.0.0.0/0</ipRange>
    <loginPolicyId>1</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <muserId/>
  <name>defaultLoginPolicy.name</name>
  <rule>ALLOW</rule>
  <timeRanges>
    <endTime>24:00</endTime>
    <loginPolicyId>1</loginPolicyId>
    <startTime>00:00</startTime>
  </timeRanges>
</loginPolicy>
<mobileNationCode>86</mobileNationCode>
<organization>
  <alias>holotest</alias>
  <cuserId/>
  <id>14</id>
  <internal>false</internal>
  <level>0.1</level>
  <muserId/>
  <name>holotest</name>
  <parentId>1</parentId>
  <supportRegions>cn-qingdao-env25-d01</supportRegions>
</organization>
<parentPk>1035075598054324</parentPk>
<primaryKey>233913275599630622</primaryKey>
```

```
<roles>
  <active>true</active>
  <code>ROLE_ORGANIZATION_ADMIN</code>
  <default>true</default>
  <description>defaultRole.organizationAdmin.description</description>
  <id>4</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.organizationAdmin.name</roleName>
  <roleRange>roleRange.orgAndSubOrgs</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
</roles>
<roles>
  <active>true</active>
  <code>ROLE_YUNDUN_PLATSECCONF_ADMIN</code>
  <default>true</default>
  <description>defaultRole.yundunPlatSecConfAdmin.description</description>
  <id>15</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.yundunPlatSecConfAdmin.name</roleName>
  <roleRange>roleRange.allOrganizations</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
</roles>
<roles>
  <active>true</active>
  <code>ROLE_YUNDUN_PLATSEC_ADMIN</code>
  <default>true</default>
  <description>defaultRole.yundunPlatSecAdmin.description</description>
  <id>12</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.yundunPlatSecAdmin.name</roleName>
  <roleRange>roleRange.allOrganizations</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
</roles>
<status>ACTIVE</status>
<userRoles>
  <active>true</active>
  <code>ROLE_ORGANIZATION_ADMIN</code>
  <cuserId/>
```

```
      <default>true</default>
      <description>defaultRole.organizationAdmin.description</description>
      <id>4</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.organizationAdmin.name</roleName>
      <roleRange>roleRange.orgAndSubOrgs</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
    <userRoles>
      <active>true</active>
      <code>ROLE_YUNDUN_PLATSEC_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.yundunPlatSecAdmin.description</description>
      <id>12</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.yundunPlatSecAdmin.name</roleName>
      <roleRange>roleRange.allOrganizations</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
    <userRoles>
      <active>true</active>
      <code>ROLE_YUNDUN_PLATSECCONF_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.yundunPlatSecConfAdmin.description</description>
      <id>15</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.yundunPlatSecConfAdmin.name</roleName>
      <roleRange>roleRange.allOrganizations</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
  </data>
  <message>success</message>
```

```
<pageInfo>
   <currentPage>1</currentPage>
   <pageSize>10</pageSize>
   <total>1</total>
   <totalPage>1</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>2d3aaa3d-b6c6-4700-8b19-2fae40a86a46</asapiRequestId>
```

1.2.4.4.9. ListDeletedUsers

You can call this operation to query the list of deleted users.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDeletedUsers	The operation that you want to perform. Set the value to ListDeletedUsers.
LoginName	String	No	testu	The logon name of the user.
CurrentPage	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListDeletedUsers
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 2,
  "data": [
    {
      "cellphoneNum": "18700000000",
      "default": false,
      "deleted": true,
      "displayName": "testuser",
      "email": "a@a.com",
      "enableDingTalk": true,
      "enableEmail": true,
      "enableShortMessage": true,
      "id": 5031,
      "loginName": "testu",
      "mobileNationCode": "86",
      "status": "ACTIVE",
      "telephoneNum": "00000000"
    }
  ],
  "message": "success",
  "pageInfo": {
    "currentPage": 1,
    "pageSize": 10,
    "total": 1,
    "totalPage": 1
  },
  "redirect": false,
  "success": true,
  "asapiRequestId": "668ee4c1-5307-46a7-ab67-7f347930be00"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>2</cost>
<data>
   <cellphoneNum>18700000000</cellphoneNum>
   <default>false</default>
   <deleted>true</deleted>
   <displayName>testuser</displayName>
   <email>a@a.com</email>
   <enableDingTalk>true</enableDingTalk>
   <enableEmail>true</enableEmail>
   <enableShortMessage>true</enableShortMessage>
   <id>5031</id>
   <loginName>testu</loginName>
   <mobileNationCode>86</mobileNationCode>
   <status>ACTIVE</status>
   <telephoneNum>00000000</telephoneNum>
</data>
<message>success</message>
<pageInfo>
   <currentPage>1</currentPage>
   <pageSize>10</pageSize>
   <total>1</total>
   <totalPage>1</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>668ee4c1-5307-46a7-ab67-7f347930be00</asapiRequestId>
```

1.2.4.4.10. ListResourceGroupByAscmUserId

You can call this operation to query the list of resource sets of a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListResourceGroupByAscmUserId	The operation that you want to perform. Set the value to ListResourceGroupByAscmUserId.

Parameter	Type	Required	Example	Description
ascmuser_id	Integer	Yes	5	The ID of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListResourceGroupByAscmUserId
&ascmuser_id=5
&<Common request parameters>
```

- Sample success responses in JSON format


```
{
  "code": "200",
  "cost": 35,
  "data": [
    {
      "creator": "ASCM-SYSTEM",
      "gmtCreated": 1574415325387,
      "gmtModified": 1574415325387,
      "id": 1,
      "organizationID": 2,
      "organizationName": "SystemDefaultOrganization",
      "resourceGroupName": "DefaultResourceSet(SystemDefaultOrganization)",
      "resourceGroupType": 1
    },
    {
      "creator": "admin",
      "gmtCreated": 1574652263873,
      "gmtModified": 1574652263873,
      "id": 2,
      "organizationID": 3,
      "organizationName": "guyi-test",
      "resourceGroupName": "DefaultResourceSet(aaa)",
      "resourceGroupType": 1
    },
    {
      "creator": "admin",
      "gmtCreated": 1574652289959,
      "gmtModified": 1574652289959,
```

```
        "id": 3,
        "organizationID": 4,
        "organizationName": "guyi-quota1",
        "resourceGroupName": "DefaultResourceSet(guyi-quota1)",
        "resourceGroupType": 1
    },
    {
        "creator": "admin",
        "gmtCreated": 1574674717192,
        "gmtModified": 1574674717192,
        "id": 4,
        "organizationID": 5,
        "organizationName": "dataworks_test",
        "resourceGroupName": "DefaultResourceSet(dataworks_test)",
        "resourceGroupType": 1
    },
    {
        "creator": "admin",
        "gmtCreated": 1574909023628,
        "gmtModified": 1574909023628,
        "id": 6,
        "organizationID": 7,
        "organizationName": "ecs_test",
        "resourceGroupName": "DefaultResourceSet(ecs_test)",
        "resourceGroupType": 1
    },
    {
        "creator": "admin",
        "gmtCreated": 1574920237061,
        "gmtModified": 1574920237061,
        "id": 7,
        "organizationID": 8,
        "organizationName": "lqx-test",
        "resourceGroupName": "DefaultResourceSet(lqx-test)",
        "resourceGroupType": 1
    },
    {
        "creator": "admin",
        "gmtCreated": 1575177273979,
        "id": 8,
        "organizationID": 3,
```

```
    "organizationName": "guyi-test",
    "resourceGroupName": "DefaultResourceSet(drds-test)"
  },
  {
    "creator": "admin",
    "gmtCreated": 1575178007522,
    "gmtModified": 1575178007522,
    "id": 9,
    "organizationID": 9,
    "organizationName": "drds_test",
    "resourceGroupName": "DefaultResourceSet(drds_test)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1575178015019,
    "id": 10,
    "organizationID": 9,
    "organizationName": "drds_test",
    "resourceGroupName": "drds_test"
  },
  {
    "creator": "admin",
    "gmtCreated": 1575447221761,
    "gmtModified": 1575447221761,
    "id": 12,
    "organizationID": 11,
    "organizationName": "dataworks_new",
    "resourceGroupName": "DefaultResourceSet(dataworks_new)",
    "resourceGroupType": 1
  },
  {
    "creator": "admin",
    "gmtCreated": 1575452086085,
    "gmtModified": 1575452086085,
    "id": 13,
    "organizationID": 12,
    "organizationName": "kenny-test",
    "resourceGroupName": "DefaultResourceSet(kenny-test)",
    "resourceGroupType": 1
```

```
    },
    {
      "creator": "admin",
      "gmtCreated": 1575514294110,
      "gmtModified": 1575514294110,
      "id": 14,
      "organizationID": 13,
      "organizationName": "AJC311",
      "resourceGroupName": "DefaultResourceSet(AJC311)",
      "resourceGroupType": 1
    },
    {
      "creator": "admin",
      "gmtCreated": 1575514325218,
      "id": 15,
      "organizationID": 13,
      "organizationName": "AJC311",
      "resourceGroupName": "AJC311"
    },
    {
      "creator": "admin",
      "gmtCreated": 1575598055480,
      "gmtModified": 1575598055480,
      "id": 16,
      "organizationID": 14,
      "organizationName": "holotest",
      "resourceGroupName": "DefaultResourceSet(holotest)",
      "resourceGroupType": 1
    }
  ],
  "message": "success",
  "pageInfo": {
    "currentPage": 0,
    "pageSize": 0,
    "total": 0,
    "totalPage": 0
  },
  "redirect": false,
  "success": true,
  "asapiRequestId": "34f61c00-952d-4139-8779-9880747b1b0d"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>35</cost>
<data>
  <creator>ASCM-SYSTEM</creator>
  <gmtCreated>1574415325387</gmtCreated>
  <gmtModified>1574415325387</gmtModified>
  <id>1</id>
  <organizationID>2</organizationID>
  <organizationName>SystemDefaultOrganization</organizationName>
  <resourceGroupName>DefaultResourceSet(SystemDefaultOrganization)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1574652263873</gmtCreated>
  <gmtModified>1574652263873</gmtModified>
  <id>2</id>
  <organizationID>3</organizationID>
  <organizationName>guyi-test</organizationName>
  <resourceGroupName>DefaultResourceSet(aaa)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1574652289959</gmtCreated>
  <gmtModified>1574652289959</gmtModified>
  <id>3</id>
  <organizationID>4</organizationID>
  <organizationName>guyi-quota1</organizationName>
  <resourceGroupName>DefaultResourceSet(guyi-quota1)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1574674717192</gmtCreated>
  <gmtModified>1574674717192</gmtModified>
  <id>4</id>
  <organizationID>5</organizationID>
  <organizationName>dataworks_test</organizationName>
```

```
    <resourceGroupName>DefaultResourceSet(dataworks_test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1574909023628</gmtCreated>
    <gmtModified>1574909023628</gmtModified>
    <id>6</id>
    <organizationID>7</organizationID>
    <organizationName>ecs_test</organizationName>
    <resourceGroupName>DefaultResourceSet(ecs_test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1574920237061</gmtCreated>
    <gmtModified>1574920237061</gmtModified>
    <id>7</id>
    <organizationID>8</organizationID>
    <organizationName>lqx-test</organizationName>
    <resourceGroupName>DefaultResourceSet(lqx-test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575177273979</gmtCreated>
    <id>8</id>
    <organizationID>3</organizationID>
    <organizationName>guyi-test</organizationName>
    <resourceGroupName>DefaultResourceSet(drds-test)</resourceGroupName>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575178007522</gmtCreated>
    <gmtModified>1575178007522</gmtModified>
    <id>9</id>
    <organizationID>9</organizationID>
    <organizationName>drds_test</organizationName>
    <resourceGroupName>DefaultResourceSet(drds_test)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
```

```
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1575178015019</gmtCreated>
  <id>10</id>
  <organizationID>9</organizationID>
  <organizationName>drds_test</organizationName>
  <resourceGroupName>drds_test</resourceGroupName>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1575447221761</gmtCreated>
  <gmtModified>1575447221761</gmtModified>
  <id>12</id>
  <organizationID>11</organizationID>
  <organizationName>dataworks_new</organizationName>
  <resourceGroupName>DefaultResourceSet(dataworks_new)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1575452086085</gmtCreated>
  <gmtModified>1575452086085</gmtModified>
  <id>13</id>
  <organizationID>12</organizationID>
  <organizationName>kenny-test</organizationName>
  <resourceGroupName>DefaultResourceSet(kenny-test)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
  <gmtCreated>1575514294110</gmtCreated>
  <gmtModified>1575514294110</gmtModified>
  <id>14</id>
  <organizationID>13</organizationID>
  <organizationName>AJC311</organizationName>
  <resourceGroupName>DefaultResourceSet(AJC311)</resourceGroupName>
  <resourceGroupType>1</resourceGroupType>
</data>
<data>
  <creator>admin</creator>
```

```
    <gmtCreated>1575514325218</gmtCreated>
    <id>15</id>
    <organizationID>13</organizationID>
    <organizationName>AJC311</organizationName>
    <resourceGroupName>AJC311</resourceGroupName>
</data>
<data>
    <creator>admin</creator>
    <gmtCreated>1575598055480</gmtCreated>
    <gmtModified>1575598055480</gmtModified>
    <id>16</id>
    <organizationID>14</organizationID>
    <organizationName>holotest</organizationName>
    <resourceGroupName>DefaultResourceSet(holotest)</resourceGroupName>
    <resourceGroupType>1</resourceGroupType>
</data>
<message>success</message>
<pageInfo>
    <currentPage>0</currentPage>
    <pageSize>0</pageSize>
    <total>0</total>
    <totalPage>0</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>34f61c00-952d-4139-8779-9880747b1b0d</asapiRequestId>
```


1.2.4.4.11. ModifyUserInformation

You can call this operation to modify user information.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyUserInformation	The operation that you want to perform. Set the value to ModifyUserInformation.
Id	Integer	No	100	The ID of the user.

Parameter	Type	Required	Example	Description
LoginName	String	No	test	The logon name of the user.
DisplayName	String	No	test	The new display name of the user.
LoginPolicyId	Integer	No	100	The new ID of the logon policy.
MobileNationCode	String	No	test	The new country code.
CellphoneNum	String	No	12345678	The new mobile number.
TelephoneNum	String	No	12345678	The new telephone number.
Email	String	No	est@test.com	The new email address.
Description	String	No	test	The new description of the user.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ModifyUserInformation
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 32,
  "data": {
    "cellphoneNum": "18700993235",
    "default": false,
    "defaultRole": {
      "active": true,
      "code": "ROLE_ORGANIZATION_ADMIN",
      "cuserId": "",
      "default": true,
      "description": "defaultRole.organizationAdmin.description",
      "id": 4,
      "muserId": "",
```

```
    "rAMRole": false,
    "roleLevel": 2147483647,
    "roleName": "defaultRole.organizationAdmin.name",
    "roleRange": "roleRange.orgAndSubOrgs",
    "roleType": "ROLETYPE_ASCM"
  },
  "deleted": false,
  "displayName": "holo",
  "email": "holouser@ascm.com",
  "enableDingTalk": true,
  "enableEmail": true,
  "enableShortMessage": true,
  "id": 5030,
  "lastLoginTime": 1575599944661,
  "loginName": "holouser",
  "loginPolicy": {
    "cuserId": "",
    "default": true,
    "id": 1,
    "ipRanges": [
      {
        "ipRange": "0.0.0.0/0",
        "loginPolicyId": 1,
        "protocol": "IPV4"
      }
    ],
    "muserId": "",
    "name": "defaultLoginPolicy.name",
    "rule": "ALLOW",
    "timeRanges": [
      {
        "endTime": "24:00",
        "loginPolicyId": 1,
        "startTime": "00:00"
      }
    ]
  },
  "mobileNationCode": "86",
  "organization": {
    "alias": "holotest",
    "cuserId": ""
```

```
    "cuserId": "",
    "id": 14,
    "internal": false,
    "level": "0.1",
    "muserId": "",
    "name": "holotest",
    "parentId": 1,
    "supportRegions": "cn-qingdao-env25-d01"
  },
  "parentPk": "1035075598054324",
  "primaryKey": "233913275599630622",
  "roles": [
    {
      "active": true,
      "code": "ROLE_ORGANIZATION_ADMIN",
      "default": true,
      "description": "defaultRole.organizationAdmin.description",
      "id": 4,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.organizationAdmin.name",
      "roleRange": "roleRange.orgAndSubOrgs",
      "roleType": "ROLETYPE_ASCM"
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_PLATSECCONF_ADMIN",
      "default": true,
      "description": "defaultRole.yundunPlatSecConfAdmin.description",
      "id": 15,
      "rAMRole": false,
      "roleLevel": 2147483647,
      "roleName": "defaultRole.yundunPlatSecConfAdmin.name",
      "roleRange": "roleRange.allOrganizations",
      "roleType": "ROLETYPE_ASCM"
    },
    {
      "active": true,
      "code": "ROLE_YUNDUN_PLATSEC_ADMIN",
      "default": true,
      "description": "defaultRole.yundunPlatSecAdmin.description",
```

```
        "id": 12,
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecAdmin.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM"
      }
    ],
    "status": "ACTIVE",
    "userGroupRoles": [],
    "userRoles": [
      {
        "active": true,
        "code": "ROLE_ORGANIZATION_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.organizationAdmin.description",
        "id": 4,
        "muserId": "",
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.organizationAdmin.name",
        "roleRange": "roleRange.orgAndSubOrgs",
        "roleType": "ROLETYPE_ASCM"
      },
      {
        "active": true,
        "code": "ROLE_YUNDUN_PLATSEC_ADMIN",
        "cuserId": "",
        "default": true,
        "description": "defaultRole.yundunPlatSecAdmin.description",
        "id": 12,
        "muserId": "",
        "rAMRole": false,
        "roleLevel": 2147483647,
        "roleName": "defaultRole.yundunPlatSecAdmin.name",
        "roleRange": "roleRange.allOrganizations",
        "roleType": "ROLETYPE_ASCM"
      },
      {
        "active": true,
```

```
            "code": "ROLE_YUNDUN_PLATSECCONF_ADMIN",
            "cuserId": "",
            "default": true,
            "description": "defaultRole.yundunPlatSecConfAdmin.description",
            "id": 15,
            "muserId": "",
            "rAMRole": false,
            "roleLevel": 2147483647,
            "roleName": "defaultRole.yundunPlatSecConfAdmin.name",
            "roleRange": "roleRange.allOrganizations",
            "roleType": "ROLETYPE_ASCM"
        }
      ]
    },
    "message": "success",
    "redirect": false,
    "success": true,
    "asapiRequestId": "c2e9fdc7-6de8-44d3-9bc3-7f18b4229df3"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>32</cost>
<data>
    <cellphoneNum>18700993235</cellphoneNum>
    <default>false</default>
    <defaultRole>
        <active>true</active>
        <code>ROLE_ORGANIZATION_ADMIN</code>
        <cuserId/>
        <default>true</default>
        <description>defaultRole.organizationAdmin.description</description>
        <id>4</id>
        <muserId/>
        <rAMRole>false</rAMRole>
        <roleLevel>2147483647</roleLevel>
        <roleName>defaultRole.organizationAdmin.name</roleName>
        <roleRange>roleRange.orgAndSubOrgs</roleRange>
        <roleType>ROLETYPE_ASCM</roleType>
    </defaultRole>
```

```
<deleted>false</deleted>
<displayName>holo</displayName>
<email>holouser@ascm.com</email>
<enableDingTalk>true</enableDingTalk>
<enableEmail>true</enableEmail>
<enableShortMessage>true</enableShortMessage>
<id>5030</id>
<lastLoginTime>1575599944661</lastLoginTime>
<loginName>holouser</loginName>
<loginPolicy>
  <cuserId/>
  <default>true</default>
  <id>1</id>
  <ipRanges>
    <ipRange>0.0.0.0/0</ipRange>
    <loginPolicyId>1</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <muserId/>
  <name>defaultLoginPolicy.name</name>
  <rule>ALLOW</rule>
  <timeRanges>
    <endTime>24:00</endTime>
    <loginPolicyId>1</loginPolicyId>
    <startTime>00:00</startTime>
  </timeRanges>
</loginPolicy>
<mobileNationCode>86</mobileNationCode>
<organization>
  <alias>holotest</alias>
  <cuserId/>
  <id>14</id>
  <internal>false</internal>
  <level>0.1</level>
  <muserId/>
  <name>holotest</name>
  <parentId>1</parentId>
  <supportRegions>cn-qingdao-env25-d01</supportRegions>
</organization>
<parentPk>1035075598054324</parentPk>
<primaryKey>233913275599630622</primaryKey>
```

```
<roles>
  <active>true</active>
  <code>ROLE_ORGANIZATION_ADMIN</code>
  <default>true</default>
  <description>defaultRole.organizationAdmin.description</description>
  <id>4</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.organizationAdmin.name</roleName>
  <roleRange>roleRange.orgAndSubOrgs</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
</roles>
<roles>
  <active>true</active>
  <code>ROLE_YUNDUN_PLATSECCONF_ADMIN</code>
  <default>true</default>
  <description>defaultRole.yundunPlatSecConfAdmin.description</description>
  <id>15</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.yundunPlatSecConfAdmin.name</roleName>
  <roleRange>roleRange.allOrganizations</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
</roles>
<roles>
  <active>true</active>
  <code>ROLE_YUNDUN_PLATSEC_ADMIN</code>
  <default>true</default>
  <description>defaultRole.yundunPlatSecAdmin.description</description>
  <id>12</id>
  <rAMRole>false</rAMRole>
  <roleLevel>2147483647</roleLevel>
  <roleName>defaultRole.yundunPlatSecAdmin.name</roleName>
  <roleRange>roleRange.allOrganizations</roleRange>
  <roleType>ROLETYPE_ASCM</roleType>
</roles>
<status>ACTIVE</status>
<userRoles>
  <active>true</active>
  <code>ROLE_ORGANIZATION_ADMIN</code>
  <cuserId/>
```

```
    <default>true</default>
    <description>defaultRole.organizationAdmin.description</description>
    <id>4</id>
    <muserId/>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.organizationAdmin.name</roleName>
    <roleRange>roleRange.orgAndSubOrgs</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </userRoles>
  <userRoles>
    <active>true</active>
    <code>ROLE_YUNDUN_PLATSEC_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.yundunPlatSecAdmin.description</description>
    <id>12</id>
    <muserId/>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunPlatSecAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </userRoles>
  <userRoles>
    <active>true</active>
    <code>ROLE_YUNDUN_PLATSECCONF_ADMIN</code>
    <cuserId/>
    <default>true</default>
    <description>defaultRole.yundunPlatSecConfAdmin.description</description>
    <id>15</id>
    <muserId/>
    <rAMRole>false</rAMRole>
    <roleLevel>2147483647</roleLevel>
    <roleName>defaultRole.yundunPlatSecConfAdmin.name</roleName>
    <roleRange>roleRange.allOrganizations</roleRange>
    <roleType>ROLETYPE_ASCM</roleType>
  </userRoles>
</data>
<message>success</message>
```

```
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>c2e9fdc7-6de8-44d3-9bc3-7f18b4229df3</asapiRequestId>
```

1.2.4.4.12. RecoverUserByLoginName

You can call this operation to restore a deleted user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RecoverUserByLoginName	The operation that you want to perform. Set the value to RecoverUserByLoginName.
LoginName	String	Yes	testtest	The logon name of the user.
OrganizationId	Integer	Yes	5	The ID of the organization.
roleIdList	Json	No	{"RoleIdList": ["5", "6", "8"]}	The list of roles.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=RecoverUserByLoginName
&LoginName=testtest
&OrganizationId=5
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 678,
  "data": {
    "cellphoneNum": "18700000000",
    "default": false,
    "deleted": false,
    "displayName": "testuser",
    "email": "a@a.com",
    "enableDingTalk": true,
    "enableEmail": true,
```

```
"enableShortMessage": true,
"id": 5031,
"loginName": "testu",
"loginPolicy": {
  "cuserId": "",
  "default": true,
  "id": 1,
  "ipRanges": [
    {
      "ipRange": "0.0.0.0/0",
      "loginPolicyId": 1,
      "protocol": "IPV4"
    }
  ],
  "muserId": "",
  "name": "defaultLoginPolicy.name",
  "rule": "ALLOW",
  "timeRanges": [
    {
      "endTime": "24:00",
      "loginPolicyId": 1,
      "startTime": "00:00"
    }
  ]
},
"mobileNationCode": "86",
"organization": {
  "alias": "ecs_test",
  "cuserId": "",
  "id": 7,
  "internal": false,
  "level": "0.1",
  "muserId": "",
  "name": "ecs_test",
  "parentId": 1,
  "supportRegions": "cn-qingdao-env25-d01"
},
"parentPk": "1979974909022324",
"primaryKey": "245466875700598631",
"roles": [
  {
```

```
        {
          "active": true,
          "code": "ROLE_RESOURCE_SET_ADMIN",
          "default": true,
          "description": "defaultRole.resourceSetAdmin.description",
          "id": 5,
          "rAMRole": false,
          "roleLevel": 2147483647,
          "roleName": "defaultRole.resourceSetAdmin.name",
          "roleRange": "roleRange.resourceSet",
          "roleType": "ROLETYPE_ASCM"
        }
      ],
      "status": "ACTIVE",
      "telephoneNum": "00000000",
      "userGroupRoles": [],
      "userRoles": [
        {
          "active": true,
          "code": "ROLE_RESOURCE_SET_ADMIN",
          "cuserId": "",
          "default": true,
          "description": "defaultRole.resourceSetAdmin.description",
          "id": 5,
          "muserId": "",
          "rAMRole": false,
          "roleLevel": 2147483647,
          "roleName": "defaultRole.resourceSetAdmin.name",
          "roleRange": "roleRange.resourceSet",
          "roleType": "ROLETYPE_ASCM"
        }
      ]
    },
    "message": "success",
    "redirect": false,
    "success": true,
    "asapiRequestId": "90e3b86e-3039-4e42-930c-1a8e0032deff"
}
```


- Sample success responses in XML format


```
<code>200</code>
```

```
<cost>678</cost>
<data>
  <cellphoneNum>18700000000</cellphoneNum>
  <default>false</default>
  <deleted>false</deleted>
  <displayName>testuser</displayName>
  <email>a@a.com</email>
  <enableDingTalk>true</enableDingTalk>
  <enableEmail>true</enableEmail>
  <enableShortMessage>true</enableShortMessage>
  <id>5031</id>
  <loginName>testu</loginName>
  <loginPolicy>
    <cuserId/>
    <default>true</default>
    <id>1</id>
    <ipRanges>
      <ipRange>0.0.0.0/0</ipRange>
      <loginPolicyId>1</loginPolicyId>
      <protocol>IPV4</protocol>
    </ipRanges>
    <muserId/>
    <name>defaultLoginPolicy.name</name>
    <rule>ALLOW</rule>
    <timeRanges>
      <endTime>24:00</endTime>
      <loginPolicyId>1</loginPolicyId>
      <startTime>00:00</startTime>
    </timeRanges>
  </loginPolicy>
  <mobileNationCode>86</mobileNationCode>
  <organization>
    <alias>ecs_test</alias>
    <cuserId/>
    <id>7</id>
    <internal>false</internal>
    <level>0.1</level>
    <muserId/>
    <name>ecs_test</name>
    <parentId>1</parentId>
    <supportRegions>cn-qingdao-env25-d01</supportRegions>
```

```
    </organization>
    <parentPk>1979974909022324</parentPk>
    <primaryKey>245466875700598631</primaryKey>
    <roles>
      <active>true</active>
      <code>ROLE_RESOURCE_SET_ADMIN</code>
      <default>true</default>
      <description>defaultRole.resourceSetAdmin.description</description>
      <id>5</id>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.resourceSetAdmin.name</roleName>
      <roleRange>roleRange.resourceSet</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </roles>
    <status>ACTIVE</status>
    <telephoneNum>00000000</telephoneNum>
    <userRoles>
      <active>true</active>
      <code>ROLE_RESOURCE_SET_ADMIN</code>
      <cuserId/>
      <default>true</default>
      <description>defaultRole.resourceSetAdmin.description</description>
      <id>5</id>
      <muserId/>
      <rAMRole>false</rAMRole>
      <roleLevel>2147483647</roleLevel>
      <roleName>defaultRole.resourceSetAdmin.name</roleName>
      <roleRange>roleRange.resourceSet</roleRange>
      <roleType>ROLETYPE_ASCM</roleType>
    </userRoles>
  </data>
  <message>success</message>
  <redirect>false</redirect>
  <success>true</success>
  <asapiRequestId>90e3b86e-3039-4e42-930c-1a8e0032deff</asapiRequestId>
```


1.2.4.4.13. RemoveRoleFromUser

You can call this operation to unbind a role from a user.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveRoleFromUser	The operation that you want to perform. Set the value to RemoveRoleFromUser.
LoginName	String	Yes	testu	The logon name of the user.
RoleId	Integer	Yes	5	The ID of the role.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=RemoveRoleFromUser
&LoginName=testu
&RoleId=5
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 174,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "82b5ab6e-c5a2-4c67-b694-efbdddd968f6"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>174</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>82b5ab6e-c5a2-4c67-b694-efbdddd968f6</asapiRequestId>
```

1.2.4.5. Logon policies

1.2.4.5.1. AddLoginPolicy

You can call this operation to create a logon policy.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AddLoginPolicy	The operation that you want to perform. Set the value to AddLoginPolicy.
Name	String	Yes	testlp	The name of the logon policy.
Description	String	No	just for test	The description of the logon policy.
Rule	String	No	deny	The rules of the logon policy. Valid values: Allow and Deny.
ipRanges	JSON	No	{"IpRanges": [{ " ipRange": "1. 1.1.0/24" }, { " ipRange": "2.2.2 .5/24" }]}	The CIDR blocks.
timeRanges	JSON	No	{"TimeRanges ": [{ "startTime ": "11:00", " endTime": " 13:35" }, { " startTime": "12: 00", "endTime": "12:30" }]}	The time ranges.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=AddLoginPolicy
&Name=testlp
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
   "code": 200,
```

```
  "cost": 7,
  "data": {
    "cuserId": "",
    "default": false,
    "description": "just for test",
    "id": 1001,
    "ipRanges": [
      {
        "ipRange": "1.1.1.0/24",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      },
      {
        "ipRange": "2.2.2.5/24",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      }
    ],
    "muserId": "",
    "name": "testIp",
    "rule": "DENY",
    "timeRanges": [
      {
        "endTime": "13:35",
        "loginPolicyId": 1001,
        "startTime": "11:00"
      },
      {
        "endTime": "12:30",
        "loginPolicyId": 1001,
        "startTime": "12:00"
      }
    ]
  },
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "c21470ee-7f4d-4282-8be7-959755fe2015"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>7</cost>
<data>
  <cuserId/>
  <default>false</default>
  <description>just for test</description>
  <id>1001</id>
  <ipRanges>
    <ipRange>1.1.1.0/24</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <ipRanges>
    <ipRange>2.2.2.5/24</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <muserId/>
  <name>testIp</name>
  <rule>DENY</rule>
  <timeRanges>
    <endTime>13:35</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>11:00</startTime>
  </timeRanges>
  <timeRanges>
    <endTime>12:30</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>12:00</startTime>
  </timeRanges>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>c21470ee-7f4d-4282-8be7-959755fe2015</asapiRequestId>
```

1.2.4.5.2. CheckLoginPolicy

You can call this operation to check a logon policy.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CheckLoginPolicy	The operation that you want to perform. Set the value to CheckLoginPolicy.
Ip	String	Yes	1.1.1.1	The IP address to be checked. Logon policies include whitelist and blacklist policies. • A whitelist policy is used to check the IP addresses from which you can log on. • A blacklist policy is used to check the IP addresses from which you cannot log on.
LoginName	String	Yes	testu	The logon name to be checked.
Time	Long	No	1575889112	The timestamp to be checked. By default, the current system timestamp is used.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=CheckLoginPolicy
&Ip=1.1.1.1
&LoginName=testu
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 4,
  "data": true,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "8bdd2a12-0986-4574-9f54-73d3ed24297a"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>4</cost>
<data>true</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>8bdd2a12-0986-4574-9f54-73d3ed24297a</asapiRequestId>
```

1.2.4.5.3. ListLoginPolicies

You can call this operation to query the list of logon policies.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListLoginPolicies	The operation that you want to perform. Set the value to ListLoginPolicies.
Name	String	No	test	The name of the logon policy.
CurrentPage	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ListLoginPolicies
&<Common request parameters>
```

- Sample success responses in JSON format


```
{
  "code": "200",
  "cost": 4,
  "data": [
    {
      "cuserId": "",
      "default": true,
      "id": 1,
      "ipRanges": [
        {
          "ipRange": "0.0.0.0/0",
          "loginPolicyId": 1,
          "protocol": "IPV4"
        }
      ],
      "muserId": "",
      "name": "defaultLoginPolicy.name",
      "rule": "ALLOW",
      "timeRanges": [
        {
          "endTime": "24:00",
          "loginPolicyId": 1,
          "startTime": "00:00"
        }
      ]
    },
    {
      "cuserId": "",
      "default": false,
      "description": "just for test",
      "id": 1001,
      "ipRanges": [
        {
          "ipRange": "1.1.1.0/24",
          "loginPolicyId": 1001,
          "protocol": "IPV4"
        }
```

```
            },
            {
              "ipRange": "2.2.2.5/24",
              "loginPolicyId": 1001,
              "protocol": "IPV4"
            }
          ],
          "muserId": "",
          "name": "testlp",
          "rule": "DENY",
          "timeRanges": [
            {
              "endTime": "13:35",
              "loginPolicyId": 1001,
              "startTime": "11:00"
            },
            {
              "endTime": "12:30",
              "loginPolicyId": 1001,
              "startTime": "12:00"
            }
          ]
        }
    ],
    "message": "success",
    "pageInfo": {
      "currentPage": 1,
      "pageSize": 10,
      "total": 2,
      "totalPage": 1
    },
    "redirect": false,
    "success": true,
    "asapiRequestId": "9a972f58-8847-4eca-b0a3-56029e7ff555"
}
```


- Sample success responses in XML format


```
<code>200</code>
<cost>4</cost>
<data>
  <cuserId/>
```

```
    <default>true</default>
    <id>1</id>
    <ipRanges>
      <ipRange>0.0.0.0/0</ipRange>
      <loginPolicyId>1</loginPolicyId>
      <protocol>IPV4</protocol>
    </ipRanges>
    <muserId/>
    <name>defaultLoginPolicy.name</name>
    <rule>ALLOW</rule>
    <timeRanges>
      <endTime>24:00</endTime>
      <loginPolicyId>1</loginPolicyId>
      <startTime>00:00</startTime>
    </timeRanges>
  </data>
  <data>
    <cuserId/>
    <default>false</default>
    <description>just for test</description>
    <id>1001</id>
    <ipRanges>
      <ipRange>1.1.1.0/24</ipRange>
      <loginPolicyId>1001</loginPolicyId>
      <protocol>IPV4</protocol>
    </ipRanges>
    <ipRanges>
      <ipRange>2.2.2.5/24</ipRange>
      <loginPolicyId>1001</loginPolicyId>
      <protocol>IPV4</protocol>
    </ipRanges>
    <muserId/>
    <name>testIp</name>
    <rule>DENY</rule>
    <timeRanges>
      <endTime>13:35</endTime>
      <loginPolicyId>1001</loginPolicyId>
      <startTime>11:00</startTime>
    </timeRanges>
    <timeRanges>
      <endTime>12:30</endTime>
```

```
      <loginPolicyId>1001</loginPolicyId>
      <startTime>12:00</startTime>
    </timeRanges>
</data>
<message>success</message>
<pageInfo>
    <currentPage>1</currentPage>
    <pageSize>10</pageSize>
    <total>2</total>
    <totalPage>1</totalPage>
</pageInfo>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>9a972f58-8847-4eca-b0a3-56029e7ff555</asapiRequestId>
```

1.2.4.5.4. ModifyLoginPolicy

You can call this operation to modify a logon policy.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyLoginPolicy	The operation that you want to perform. Set the value to ModifyLoginPolicy.
Id	Integer	Yes	2	The ID of the logon policy.
Name	String	No	testtttt	The new name of the logon policy.
Description	String	No	tttt	The new description of the logon policy.
Rule	String	No	allow	The new rules of the logon policy.
ipRanges	JSON	No	{ "ipRanges": [{ "ipRange": "0.0.0.0/0", "loginPolicyId": 1, "protocol": "IPV4" }]}	The list of new CIDR blocks.

Parameter	Type	Required	Example	Description
timeRanges	JSON	No	{"timeRanges": [{ "endTime": "24:00", "loginPolicyId": 1, "startTime": "00:00" }]}	The list of new time ranges.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=ModifyLoginPolicy
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 10,
  "data": {
    "cuserId": "",
    "default": false,
    "description": "tttt",
    "id": 1001,
    "ipRanges": [
      {
        "ipRange": "0.0.0.0/0",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      }
    ],
    "mtime": 1575703004583,
    "muserId": "admin",
    "name": "testwwww",
    "rule": "ALLOW",
    "timeRanges": [
      {
        "endTime": "24:00",
        "loginPolicyId": 1001,
        "startTime": "00:00"
      }
    ]
  },
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "47b48a09-a700-4335-a2c1-c87eb228edf6"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>10</cost>
<data>
  <cuserId/>
  <default>false</default>
  <description>tttt</description>
  <id>1001</id>
  <ipRanges>
    <ipRange>0.0.0.0/0</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <mtime>1575703004583</mtime>
  <muserId>admin</muserId>
  <name>testwwww</name>
  <rule>ALLOW</rule>
  <timeRanges>
    <endTime>24:00</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>00:00</startTime>
  </timeRanges>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>47b48a09-a700-4335-a2c1-c87eb228edf6</asapiRequestId>
```


1.2.4.5.5. GetLoginPolicyById

You can call this operation to query the detailed information of a logon policy.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetLoginPolicyById	The operation that you want to perform. Set the value to GetLoginPolicyById.
Id	Integer	Yes	1	The ID of the logon policy.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetLoginPolicyById
&Id=1
&<Common request parameters>
```

- Sample success responses in JSON format


```
{
  "code": "200",
  "cost": 7,
  "data": {
    "cuserId": "",
    "default": false,
    "description": "just for test",
    "id": 1001,
    "ipRanges": [
      {
        "ipRange": "1.1.1.0/24",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      },
      {
        "ipRange": "2.2.2.5/24",
        "loginPolicyId": 1001,
        "protocol": "IPV4"
      }
    ],
    "muserId": "",
    "name": "testIp",
    "rule": "DENY",
    "timeRanges": [
      {
        "endTime": "13:35",
        "loginPolicyId": 1001,
        "startTime": "11:00"
      },
      {
        "endTime": "12:30",
        "loginPolicyId": 1001,
        "startTime": "12:00"
```

```
        }
      ]
    },
    "message": "success",
    "redirect": false,
    "success": true,
    "asapiRequestId": "c21470ee-7f4d-4282-8be7-959755fe2015"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>7</cost>
<data>
  <cuserId/>
  <default>false</default>
  <description>just for test</description>
  <id>1001</id>
  <ipRanges>
    <ipRange>1.1.1.0/24</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <ipRanges>
    <ipRange>2.2.2.5/24</ipRange>
    <loginPolicyId>1001</loginPolicyId>
    <protocol>IPV4</protocol>
  </ipRanges>
  <muserId/>
  <name>testIp</name>
  <rule>DENY</rule>
  <timeRanges>
    <endTime>13:35</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>11:00</startTime>
  </timeRanges>
  <timeRanges>
    <endTime>12:30</endTime>
    <loginPolicyId>1001</loginPolicyId>
    <startTime>12:00</startTime>
  </timeRanges>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>c21470ee-7f4d-4282-8be7-959755fe2015</asapiRequestId>
```


1.2.5. Cloud services

1.2.5.1. DescribeModifiableInstanceType

You can call this operation to query the ECS instance types that can be upgraded or downgraded.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeModifiableInstanceType	The operation that you want to perform. Set the value to DescribeModifiableInstanceType.
destinationResource	String	Yes	sfdsf	Set the value to instanceType.
instanceId	String	Yes	sdfdfsdfsdf	The ID of the instance.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=DescribeModifiableInstanceType
&destinationResource=sfdsf
&instanceId=sdfdfsdfsdf
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "success":true,
  "data":"",
  "message":"success",
  "code":null,
  "httpCode":null,
  "ip":null,
  "requestId":null,
  "httpOk":false,
  "asapiRequestId":"979a704d-9ab7-4ddb-a3bf-cb81c5e752bb"
}
```

1.2.5.2. DescribeSeriesIdFamilies

You can call this operation to query instance families.

Request parameters

Parameter	Type	Required	Example	Description
resourceType	String	Yes	DRDS	The type of the resource.
status	String	Yes	Available	The status of the request.

Examples

- Sample requests

```
http(s)://[Endpoint]/?<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "success":true,
  "data":[
    {
      "id":null,
      "gmtCreate":"2019-12-17T03:57:03.315+0000",
      "gmtModified":"2019-12-17T03:57:03.315+0000",
      "creator":"deploy",
      "modifier":"deploy",
      "isDeleted":"N",
      "pageStart":0,
      "pageSize":8,
      "pageSort":"id",
      "pageOrder":"ASC",
      "orderBy":{
        "id":"ASC"
      },
      "seriesId":"drds.sn2.4c16g",
      "seriesName":"",
      "seriesNameLabel":"drds.sn2.4c16g",
      "resourceType":"DRDS",
      "deleted":false
    }
  ],
  "message":"describeSeriesIdFamilies success",
  "code":null,
  "httpCode":null,
  "ip":null,
  "requestId":null,
  "httpOk":false,
  "asapiRequestId":"0ad9a14b-808a-4706-b4e9-323bb012230f"
}
```


1.2.5.3. GetEnvProducts

You can call this operation to obtain the list of services in an environment.

Request parameters

None

Examples

- Sample requests

```
http(s)://[Endpoint]/?<Common request parameters>
```

- Sample success responses in JSON format

```
{
    "code": 200,
    "result": [
      "middleWare-staragent",
      "opsapi",
      "drds",
      "blink",
      "odps",
      "xnet2",
      "ros",
      "datahub",
      "astc",
      "middleWareAll",
      "arms",
      "dfs",
      "sls",
      "acs",
      "butler",
      "oss",
      "ram",
      "middleWare-redis",
      "xdm",
      "ascm",
      "vpc",
      "ots",
      "hologres",
      "apigateway",
      "oneConsole",
      "dauthProduct",
      "licenceManage",
      "mq",
      "elasticsearch",
      "emr",
      "dts",
      "yundun-beaver",
      "yd",
```

```
        "slb",
        "graphcompute",
        "astoolbox",
        "yundun-scanner",
        "aliware-taokeeper",
        "schedulerx",
        "qitian",
        "asrbr",
        "dataphin",
        "ftp",
        "yundun-cfw",
        "ess",
        "network",
        "baseServiceAll",
        "tianji",
        "csb",
        "nas",
        "config-assist",
        "kms",
        "pai",
        "buffer",
        "dms-enterprise",
        "edas",
        "webappAll",
        "quickbi",
        "asa",
        "base",
        "ecs",
        "yundun-advance",
        "aso",
        "middleWare-histore",
        "tlog",
        "dnsProduct",
        "paas",
        "minirds-mt",
        "rds",
        "acsTest",
        "bcc",
        "middleWare-queqiao",
        "vnet",
        "yundun-common",
```

```
        "middleware-dncs",
        "hbr",
        "analyticdb",
        "gpdb",
        "kvstore",
        "mongodb"
    ],
    "success": true,
    "asapiRequestId": "da4fde1f-0fc6-410d-9ae3-c4bc9c1f1228"
}
```

- Sample success responses in XML format

```
<code>200</code>
<result>middleWare-staragent</result>
<result>opsapi</result>
<result>drds</result>
<result>blink</result>
<result>odps</result>
<result>xnet2</result>
<result>ros</result>
<result>datahub</result>
<result>astc</result>
<result>middleWareAll</result>
<result>arms</result>
<result>dfs</result>
<result>sls</result>
<result>acs</result>
<result>butler</result>
<result>oss</result>
<result>ram</result>
<result>middleWare-redis</result>
<result>xdm</result>
<result>ascm</result>
<result>vpc</result>
<result>ots</result>
<result>hologres</result>
<result>apigateway</result>
<result>oneConsole</result>
<result>dauthProduct</result>
<result>licenceManage</result>
<result>mq</result>
```

```
<result>elasticsearch</result>
<result>emr</result>
<result>dts</result>
<result>yundun-beaver</result>
<result>yd</result>
<result>slb</result>
<result>graphcompute</result>
<result>astoolbox</result>
<result>yundun-scanner</result>
<result>aliware-taokeeper</result>
<result>schedulerx</result>
<result>qitian</result>
<result>asrbr</result>
<result>dataphin</result>
<result>ftp</result>
<result>yundun-cfw</result>
<result>ess</result>
<result>network</result>
<result>baseServiceAll</result>
<result>tianji</result>
<result>csb</result>
<result>nas</result>
<result>config-assist</result>
<result>kms</result>
<result>pai</result>
<result>buffer</result>
<result>dms-enterprise</result>
<result>edas</result>
<result>webappAll</result>
<result>quickbi</result>
<result>asa</result>
<result>base</result>
<result>ecs</result>
<result>yundun-advance</result>
<result>aso</result>
<result>middleWare-histore</result>
<result>tlog</result>
<result>dnsProduct</result>
<result>paas</result>
<result>minirds-mt</result>
```

```
<result>rds</result>
<result>acsTest</result>
<result>bcc</result>
<result>middleWare-queqiao</result>
<result>vnet</result>
<result>yundun-common</result>
<result>middleware-dncs</result>
<result>hbr</result>
<result>analyticdb</result>
<result>gpdb</result>
<result>kvstore</result>
<result>mongodb</result>
<success>true</success>
<asapiRequestId>da4fde1f-0fc6-410d-9ae3-c4bc9c1f1228</asapiRequestId>
```

1.2.5.4. GroupCommonSpec

You can call this operation to query the valid values of the specifications-specific fields.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GroupCommonSpec	The operation that you want to perform. Set the value to GroupCommonSpec.
groupFiled	String	Yes	storageType	The field for which to query valid values.
resourceType	String	Yes	OSS	Valid values: OSS, ADB, DRDS, SLB, NAT, MAXCOMPUTE, POSTGRESQL, ECS, RDS, IPSIX, REDIS, MONGODB, and HITSDB.
label	String	No	true	Specifies whether to internationalize the field. Valid values: true and false.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GroupCommonSpec
&groupFiled=storageType
&resourceType=OSS
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
   "success": true,
   "data": "Standard",
   "message": "",
   "code": "",
   "httpCode": "",
   "ip": "",
   "requestId": "",
   "httpOk": false,
   "asapiRequestId": "e31897b8-dfe7-442f-b292-a4e57b77a6ab"
}
```

- Sample success responses in XML format

```
<success>true</success>
<data>Standard</data>
<message/>
<code/>
<httpCode/>
<ip/>
<requestId/>
<httpOk>false</httpOk>
<asapiRequestId>e31897b8-dfe7-442f-b292-a4e57b77a6ab</asapiRequestId>
```

1.2.5.5. GetRegionsByProduct

You can call this operation to query the region list of a service.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetRegionsByProduct	The operation that you want to perform. Set the value to GetRegionsByProduct.

Parameter	Type	Required	Example	Description
productName	String	Yes	ecs	The name of the service.
organization	String	No	tets	The name of the organization.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetRegionsByProduct
&productName=ecs
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
    "RegionList": [
      {
        "RegionId": "cn-qingdao-env25-d01",
        "RegionType": "center"
      }
    ]
  }
```

- Sample success responses in XML format

```
<RegionList>
    <RegionId>cn-qingdao-env25-d01</RegionId>
    <RegionType>center</RegionType>
</RegionList>
```

1.2.5.6. DescribeInstanceTypeFamilies

You can call this operation to query ECS instance families.

Request parameters

Parameter	Type	Required	Example	Description
status	String	Yes	Available	The status.

Examples

- Sample requests

```
http(s)://[Endpoint]/? <Common request parameters>
```

- Sample success responses in JSON format

```
{
  "success":true,
  "data":{
    "instanceTypeFamilies":[
      {
        "instanceTypeFamilyId":"ecs.i2",
        "generation":""
      }
    ]
  },
  "message":"success",
  "code":null,
  "httpCode":null,
  "ip":null,
  "requestId":null,
  "httpOk":false,
  "asapiRequestId":"979a704d-9ab7-4ddb-a3bf-cb81c5e752bb"
}
```

1.2.5.7. GetClustersByProduct

You can call this operation to query the clusters of a service.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetClustersByProduct	The operation that you want to perform. Set the value to GetClustersByProduct.
productName	String	Yes	ecs	The name of the service.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetClustersByProduct
&productName=ecs
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
        "ClusterList": [
          {
            "cn-qingdao-env25-d01": [
              "ECS-IO11-A-1afd",
              "ECS-CPU-A-7509",
              "ECS-IO11-A-7d32",
              "ECS-IO11-A-1ca8",
              "ECS-CPU-A-7906",
              "ECS-IO8River-A-8b48",
              "ECS-CPU-A-2f75",
              "ECS-CPU-A-1786",
              "ECS-controller-A-177c",
              "ECS-IO8-A-5679",
              "ECS-river-A-177f",
              "ECS-IO7-A-e7a4",
              "ECS-GPU-A-2c01",
              "ECS-controller-A-177d",
              "ECS-IO11-A-54b1"
            ]
          }
        ]
    }
```

- Sample success responses in XML format

```
<ClusterList>
    <cn-qingdao-env25-d01>ECS-IO11-A-1afd</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-CPU-A-7509</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-IO11-A-7d32</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-IO11-A-1ca8</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-CPU-A-7906</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-IO8River-A-8b48</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-CPU-A-2f75</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-CPU-A-1786</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-controller-A-177c</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-IO8-A-5679</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-river-A-177f</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-IO7-A-e7a4</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-GPU-A-2c01</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-controller-A-177d</cn-qingdao-env25-d01>
    <cn-qingdao-env25-d01>ECS-IO11-A-54b1</cn-qingdao-env25-d01>
</ClusterList>
```

1.2.5.8. GetJumpUrlByProduct

You can call this operation to query the URL of a service.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetJumpUrlByProduct	The operation that you want to perform. Set the value to GetJumpUrlByProduct.
productName	String	Yes	dts	The name of the service.
organization	String	No	test	The name of the organization.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=GetJumpUrlByProduct
&productName=dts
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
   "body": "https://dts.console.example.com",
   "code": 0,
   "successResponse": true,
   "asapiRequestId": "85c65db8-7d46-4d14-b11d-9f744c3e0654"
}
```

- Sample success responses in XML format

```
<body>https://dts.console.example.com</body>
<code>0</code>
<successResponse>true</successResponse>
<asapiRequestId>85c65db8-7d46-4d14-b11d-9f744c3e0654</asapiRequestId>
```

1.2.6. Configurations

1.2.6.1. RemoveLoginPolicyById

You can call this operation to delete a logon policy.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveLoginPolicyById	The operation that you want to perform. Set the value to RemoveLoginPolicyById.
Id	Integer	Yes	2	The ID of the logon policy.

Examples

- Sample requests

```
http(s)://[Endpoint]/?Action=RemoveLoginPolicyById
&Id=2
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 4,
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "9e2c88fc-c86e-4c73-bb63-838f23fe68b1"
}
```

- Sample success responses in XML format

```
<code>200</code>
<cost>4</cost>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>9e2c88fc-c86e-4c73-bb63-838f23fe68b1</asapiRequestId>
```

1.2.6.2. GetPasswordPolicy

You can call this operation to query a password policy.

Request parameters

None

Examples

- Sample requests

```
http(s)://[Endpoint]/?<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 7,
  "data": {
    "hardExpiry": true,
    "maxLoginAttemps": 5,
    "maxPasswordAge": 90,
    "minimumPasswordLength": 10,
    "passwordErrorLockPeriod": 30,
    "passwordErrorTolerancePeriod": 60,
    "passwordReusePrevention": 5,
    "requireLowercaseCharacters": true,
    "requireNumbers": true,
    "requireSymbols": true,
    "requireUppercaseCharacters": true
  },
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "71e2744a-28b6-4c7e-9dc2-f22dded4f15c"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>7</cost>
<data>
  <hardExpiry>true</hardExpiry>
  <maxLoginAttemps>5</maxLoginAttemps>
  <maxPasswordAge>90</maxPasswordAge>
  <minimumPasswordLength>10</minimumPasswordLength>
  <passwordErrorLockPeriod>30</passwordErrorLockPeriod>
  <passwordErrorTolerancePeriod>60</passwordErrorTolerancePeriod>
  <passwordReusePrevention>5</passwordReusePrevention>
  <requireLowercaseCharacters>true</requireLowercaseCharacters>
  <requireNumbers>true</requireNumbers>
  <requireSymbols>true</requireSymbols>
  <requireUppercaseCharacters>true</requireUppercaseCharacters>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>71e2744a-28b6-4c7e-9dc2-f22dded4f15c</asapiRequestId>
```

1.2.6.3. ResetPasswordPolicy

You can call this operation to reset a password policy.

Request parameters

None

Examples

- Sample requests

```
http(s)://[Endpoint]/?<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 7,
  "data": {
    "hardExpiry": true,
    "maxLoginAttemps": 5,
    "maxPasswordAge": 90,
    "minimumPasswordLength": 10,
    "passwordErrorLockPeriod": 30,
    "passwordErrorTolerancePeriod": 60,
    "passwordReusePrevention": 5,
    "requireLowercaseCharacters": true,
    "requireNumbers": true,
    "requireSymbols": true,
    "requireUppercaseCharacters": true
  },
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "71e2744a-28b6-4c7e-9dc2-f22dded4f15c"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>7</cost>
<data>
  <hardExpiry>true</hardExpiry>
  <maxLoginAttemps>5</maxLoginAttemps>
  <maxPasswordAge>90</maxPasswordAge>
  <minimumPasswordLength>10</minimumPasswordLength>
  <passwordErrorLockPeriod>30</passwordErrorLockPeriod>
  <passwordErrorTolerancePeriod>60</passwordErrorTolerancePeriod>
  <passwordReusePrevention>5</passwordReusePrevention>
  <requireLowercaseCharacters>true</requireLowercaseCharacters>
  <requireNumbers>true</requireNumbers>
  <requireSymbols>true</requireSymbols>
  <requireUppercaseCharacters>true</requireUppercaseCharacters>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>71e2744a-28b6-4c7e-9dc2-f22dded4f15c</asapiRequestId>
```

1.2.6.4. SetPasswordPolicy

You can call this operation to modify a password policy.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetPasswordPolicy	The operation that you want to perform. Set the value to `SetPasswordPolicy`.
HardExpiry	Boolean	No	true	Specifies whether to disable logon after the password expires.
RequireNumbers	Boolean	No	true	Specifies whether digits are required.
RequireSymbols	Boolean	No	true	Specifies whether special characters are required.
RequireLowercaseCharacters	Boolean	No	true	Specifies whether lowercase letters are required.

Parameter	Type	Required	Example	Description
RequireUppercaseCharacters	Boolean	No	true	Specifies whether uppercase letters are required.
MaxLoginAttemps	Integer	No	5	The maximum number of allowed logon attempts.
MaxPasswordAge	Integer	No	90	The validity period of the password. Unit: days.
MinimumPasswordLength	Integer	No	10	The minimum length of the password.
PasswordReusePrevention	Integer	No	5	The maximum number of allowed password reuse attempts.

Examples

- Sample requests

```
http(s)://[Endpoint]/? Action=SetPasswordPolicy
&<Common request parameters>
```

- Sample success responses in JSON format

```
{
  "code": "200",
  "cost": 7,
  "data": {
    "hardExpiry": true,
    "maxLoginAttemps": 5,
    "maxPasswordAge": 90,
    "minimumPasswordLength": 10,
    "passwordErrorLockPeriod": 30,
    "passwordErrorTolerancePeriod": 60,
    "passwordReusePrevention": 5,
    "requireLowercaseCharacters": true,
    "requireNumbers": true,
    "requireSymbols": true,
    "requireUppercaseCharacters": true
  },
  "message": "success",
  "redirect": false,
  "success": true,
  "asapiRequestId": "71e2744a-28b6-4c7e-9dc2-f22dded4f15c"
}
```


- Sample success responses in XML format

```
<code>200</code>
<cost>7</cost>
<data>
  <hardExpiry>true</hardExpiry>
  <maxLoginAttemps>5</maxLoginAttemps>
  <maxPasswordAge>90</maxPasswordAge>
  <minimumPasswordLength>10</minimumPasswordLength>
  <passwordErrorLockPeriod>30</passwordErrorLockPeriod>
  <passwordErrorTolerancePeriod>60</passwordErrorTolerancePeriod>
  <passwordReusePrevention>5</passwordReusePrevention>
  <requireLowercaseCharacters>true</requireLowercaseCharacters>
  <requireNumbers>true</requireNumbers>
  <requireSymbols>true</requireSymbols>
  <requireUppercaseCharacters>true</requireUppercaseCharacters>
</data>
<message>success</message>
<redirect>false</redirect>
<success>true</success>
<asapiRequestId>71e2744a-28b6-4c7e-9dc2-f22dded4f15c</asapiRequestId>
```


1.2.7. Monitoring

1.2.7.1. Major metrics of Apsara Stack services

Precautions

- The Sum function is used to measure the total amount of an object during a specific time period. For example, you can measure the total amount of Internet traffic (KB/minute) that goes through an Elastic Compute Service (ECS) instance in a period of five minutes.
- You can use OpenAPI to query monitoring data of the last 31 days.
- The value of the Dimensions parameter is a JSON string. Example: `{"instanceId":"i-23gyb3kkd"}` .

Operating system metrics for ECS

Metric	Description	Unit	Dimensions	Statistics
Host.cpu.total	The total number of CPU cores of an ECS instance.	%	instanceId	Average, Minimum, and Maximum
Host.mem.usedutilization	The memory usage of an ECS instance.	%	instanceId	Average, Minimum, and Maximum

Metric	Description	Unit	Dimensions	Statistics
Host.disk.utilization	The disk usage of an ECS instance.	%	instanceId and device	Average, Minimum, and Maximum

Basic metrics for ECS

Metric	Description	Unit	Dimensions	Statistics
CPU utilization	The CPU utilization of an ECS instance.	%	instanceId	Average, Minimum, and Maximum
System disk BPS	The number of bytes read from and written to the system disk per second.	byte/s	instanceId	Average, Minimum, and Maximum
System disk IOPS	The number of reads from and writes to the system disk per second.	count/s	instanceId	Average, Minimum, and Maximum

Metrics for ApsaraDB for RDS whose namespace is acs_rds_dashboard

- Set Project to *acs_rds_dashboard* and the statistical period to 300 seconds. Set Period to an integer multiple of 300.
- If you have set the monitoring frequency to one minute in the ApsaraDB for RDS console, you must set the statistical period to 60 seconds or more.
- Assign the actual instance ID to instanceId in the Dimensions parameter.

Metric	Description	Unit	Dimensions	Statistics
CpuUsage	The CPU utilization of an ApsaraDB for RDS instance.	%	instanceId	Average, Minimum, and Maximum
DiskUsage	The disk usage of an ApsaraDB for RDS instance.	%	instanceId	Average, Minimum, and Maximum
IOPSUsage	The IOPS usage of an ApsaraDB for RDS instance.	%	instanceId	Average, Minimum, and Maximum
ConnectionUsage	The connection usage of an ApsaraDB for RDS instance.	%	instanceId	Average, Minimum, and Maximum

Metric	Description	Unit	Dimensions	Statistics
MemoryUsage	The memory usage of an ApsaraDB for RDS instance.	%	instanceId	Average, Minimum, and Maximum

Metrics for ApsaraDB for Redis whose namespace is acs_kvstore

- Set Project to *acs_kvstore* and the statistical period to 60 seconds. Set Period to an integer multiple of 60.
- Assign the actual instance ID to instanceId in the Dimensions parameter.

Metric	Description	Unit	Dimensions	Statistics
ConnectionUsage	The connection usage on the specified node.	%	instanceId and nodeId	Average and Maximum
FailedCount	The number of failed commands.	count/s	instanceId	Average and Maximum
IntranetIn	The inbound bandwidth of the specified node.	bit/s	instanceId and nodeId	Average and Maximum
IntranetInRatio	The inbound bandwidth usage of the specified node.	%	instanceId and nodeId	Average and Maximum
IntranetOut	The outbound bandwidth of the specified node.	bit/s	instanceId and nodeId	Average and Maximum
IntranetOutRatio	The outbound bandwidth usage of the specified node.	%	instanceId and nodeId	Average and Maximum
MemoryUsage	The memory usage of the specified node.	%	instanceId and nodeId	Average and Maximum
UsedQPS	The average number of queries occurred on the specified node per second.	N/A	instanceId and nodeId	Average and Maximum

1.2.7.2. Monitoring data on time series metrics of cloud services

1.2.7.2.1. DescribeProjectMeta

You can call this operation to query information about monitored services in CloudMonitor.

The information that can be obtained by this operation includes the service description, namespace, and tags.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeProjectMeta	The operation that you want to perform. Set the value to DescribeProjectMeta.
				The tags of metrics. You can specify tags of metrics to filter services. A tag of a metric is used as a special mark of alerts triggered by the metric. For example, a tag can specify the service specification to which an alert belongs and whether an alert rule can be configured for a metric. The format is `[{"name":"Tag name","value":"Tag value"},{"name":"Tag name","value":"Tag value"}]`. Common tags are as follows: • alertUnit: the unit of the metric in the alert.

Parameter	Type	Required	Example	Description
Labels	String	No	[{"name":"product","value":"ECS"}]	Note The original metric value may be too large due to a small unit. In this case, you can use the alertUnit tag to specify an appropriate unit. Currently, this tag is mainly used in CloudMonitor. • minAlertPeriod: the minimum time interval to raise a new alert. Generally, the interval is 1 minute. • metricCategory: the service specification, for example, kvstore_sharding. Note Some services of Apsara Stack have different specifications with the same namespace. The metricCategory tag can be used to distinguish between the specifications. • is_alarm: specifies whether an alert rule can be configured for a metric. Note Apsara Stack has some other tags that are dedicated for CloudMonitor. We recommend that you do not use such tags.
PageNumber	Integer	No	1	The number of the page to return. Default value: 1.
PageSize	Integer	No	30	The number of entries to return on each page. Default value: 30. Note Apsara Stack does not limit the value of this parameter. If you want to view all desired entries on a small number of pages, set this parameter to a greater value.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Message	String	success	The error message.
PageNumber	String	1	The page number of the page returned.
PageSize	String	10	The number of entries returned on each page.
RequestId	String	4C2061B2-3B1B-43BF-A4A4-C53426F479C0	The ID of the request.
Resources	Array		The information about services found.
Resource			The information about a service found.
Description	String	CDN	The description of the service.

Parameter	Type	Example	Description
Labels	String	[{"groupFlag":true}]	The tags of metrics. Services were filtered based on the specified tags. A tag of a metric is used as a special mark of alerts triggered by the metric. For example, a tag can specify the service specification to which an alert belongs and whether an alert rule can be configured for a metric. The format is `[{"name":"Tag name","value":"Tag value"},{"name":"Tag name","value":"Tag value"}]`. Common tags are as follows: • alertUnit: the unit of the metric in the alert. Note The original metric value may be too large due to a small unit. In this case, you can use the `alertUnit` tag to specify an appropriate unit. Currently, this tag is mainly used in CloudMonitor. • minAlertPeriod: the minimum time interval to raise a new alert. Generally, the interval is 1 minute. • metricCategory: the service specification, for example, kvstore_sharding. Note Some services of Apsara Stack have different specifications with the same namespace. The metricCategory tag can be used to distinguish between the specifications. • is_alarm: specifies whether an alert rule can be configured for a metric. Note Apsara Stack has some other tags that are dedicated for CloudMonitor. We recommend that you do not use such tags.

Parameter	Type	Example	Description
Namespace	String	acs_cdn	The namespace of the service, which is used to distinguish between services. Generally, the value is in the format `acs_Abbreviation of the service name`.
Success	Boolean	true	Indicates whether the request was successful.
Total	String	12	The total number of entries returned.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeProjectMeta
&<Common request parameters>
```

Sample success responses

> **Note** The response content varies according to the services that are actually monitored.

`JSON` format

```
{
 "PageNumber":1,
 "PageSize":5,
 "RequestId":"4C2061B2-3B1B-43BF-A4A4-C53426F479C0",
 "Success":true,
 "Code":200,
 "Total":62,
 "Resources":{
  "Resource":[{
    "Description": "ECS",
    "Labels": "[{\"name\":\"product\",\"value\":\"ECS\"},{\"name\":\"productCategory\",\"value\":\"ecs,
ecs_net_interface,hpc_node,hpc_cluster\"},{\"name\":\"groupFlag\",\"value\":\"true\"},{\"name\":\"cnN
ame\",\"value\":\"ECS\"},{\"name\":\"enName\",\"value\":\"ECS\"}]",
    "Namespace": "acs_ecs_dashboard"
   },

  ]
 }
}
```


1.2.7.2.2. DescribeMetricLast

You can call this operation to query the latest monitoring data of a monitored object.

For more information about how to assign values to the Project, Metric, Period, and Dimensions parameters for Apsara Stack services, see DescribeMetricMetaList.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeMetricLast	The operation that you want to perform. Set the value to DescribeMetricLast.
MetricName	String	Yes	CPUUtilization	The name of the metric.

Parameter	Type	Required	Example	Description
Namespace	String	Yes	acs_ecs_dashboard	The namespace of the monitored service. Specify the value in the format of acs_Service name.
Dimensions	String	No	[{"instanceId":"XXXXXX"}]	The map of the monitored resources. The value is a collection of `key-value` pairs. A typical `key-value` pair is `instanceId:XXXXXX`. The `key` and `value` must be 1 to 64 bytes in length, respectively. If the length exceeds 64 bytes, excess bytes are truncated from the right. The `key` and `value` can contain letters, digits, periods (.), hyphens (-), underscores (_), forward slashes (/), and backslashes (\). **Note** If you hope to see a key-value map expressed by JSON strings, set `Dimensions` to ordered strings.
EndTime	String	No	2019-01-31 10:10:00	The end of the time range for the query. The value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC. **Note** You can only specify a time range within the last 270 days for the query.
Express	String	No	{"groupby":["userId","instanceId"]}	The expression for real-time computation on the query results.

Parameter	Type	Required	Example	Description
Length	String	No	1000	The number of entries to return on each page. Default value: 1000.
NextToken	String	No	15761432850009dd70bb64cff1f0fff6c0b08ffff073be5fb1e785e2b020f7fed9b5e137bd810a6d6cff5ae****	The pagination cursor. • If the number of entries that match the search criteria exceeds the maximum number allowed on a single page, a pagination cursor is returned. • This pagination cursor will be used as an input parameter for obtaining entries on the next page. If a response contains no pagination cursors, all the query result has been returned.
Period	String	No	60	The time interval at which monitoring data will be queried. Generally, the time interval equals the frequency for reporting metric data. Unit: seconds. • If you do not specify this parameter, raw data will be queried at the reporting frequency specified during registration of the corresponding metric. • If a statistical period was specified for the corresponding alert rule, raw data will be queried based on the statistical period.

Parameter	Type	Required	Example	Description
StartTime	String	No	2019-01-31 10:00:00	The beginning of the time range for the query. The value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC. Note You can only specify a time range covering a maximum of 31 days within the last 270 days.

Response parameters

Parameter	Type	Example	Description
RequestId	String	021472A6-25E3-4094-8D00-BA4B6A5486C3	The ID of the request.
Code	String	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Success	Boolean	true	Indicates whether the request was successful.
Period	String	60	The time interval at which monitoring data was queried. Unit: seconds.
NextToken	String	xxxxxx	The pagination cursor.
Datapoints	String	[{"timestamp":1548777660000,"userId":"123","instanceId":"i-abc","Minimum":9.92,"Average":9.92,"Maximum":9.92}]	The list of monitoring data.
Message	String	Success	The error message.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeMetricLast
&MetricName=CPUUtilization
&Namespace=acs_ecs_dashboard
&<Common request parameters>
```

Sample success responses

`XML` format

```
<QueryMetricListResponse>
  <Period>60</Period>
  <Datapoints>
    <Datapoints>
      <timestamp>1490152860000</timestamp>
      <Maximum>100</Maximum>
      <userId> 123456789876****</userId>
      <Minimum>93.1</Minimum>
      <instanceId>i-abcdefgh12****</instanceId>
      <Average>99.52</Average>
    </Datapoints>
    <Datapoints>
      <timestamp>1490152920000</timestamp>
      <Maximum>100</Maximum>
      <userId> 123456789876**** </userId>
      <Minimum>92.59</Minimum>
      <instanceId>i-abcdefgh12****</instanceId>
      <Average>99.49</Average>
    </Datapoints>
    <Datapoints>
      <timestamp>1490152980000</timestamp>
      <Maximum>100</Maximum>
      <userId>123456789876****</userId>
      <Minimum>92.86</Minimum>
      <instanceId>i-abcdefgh12****</instanceId>
      <Average>99.44</Average>
    </Datapoints>
    <Datapoints>
      <timestamp>1490153040000</timestamp>
      <Maximum>100</Maximum>
```

```
    <userId>123456789876****</userId>
    <Minimum>91.43</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.36</Average>
  </Datapoints>
  <Datapoints>
    <timestamp>1490153100000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>93.55</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.51</Average>
  </Datapoints>
  <Datapoints>
    <timestamp>1490153160000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>93.1</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.52</Average>
  </Datapoints>
  <Datapoints>
    <timestamp>1490153220000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>92.59</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.42</Average>
  </Datapoints>
  <Datapoints>
    <timestamp>1490153280000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>91.18</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.34</Average>
  </Datapoints>
  <Datapoints>
    <timestamp>1490153340000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
```

```
        <userId>123456789876****</userId>
        <Minimum>92.86</Minimum>
        <instanceId>i-abcdefgh12****</instanceId>
        <Average>99.46</Average>
      </Datapoints>
      <Datapoints>
        <timestamp>1490153400000</timestamp>
        <Maximum>100</Maximum>
        <userId>123456789876****</userId>
        <Minimum>91.18</Minimum>
        <instanceId>i-abcdefgh12****</instanceId>
        <Average>99.35</Average>
      </Datapoints>
    </Datapoints>
    <RequestId>6661EC50-8625-4161-B349-E0DD59002AB7</RequestId>
    <Success>true</Success>
    <Code>200</Code>
</QueryMetricListResponse>
```

`JSON` format

```
{
 "Period":"60",
 "Datapoints":[
  {
   "timestamp":1490152860000,
   "Maximum":100,
   "userId":"123456789876****",
   "instanceId":"i-abcdefgh12****",
   "Minimum":93.1,
   "Average":99.52
  },
  {
   "timestamp":1490152920000,
   "Maximum":100,
   "userId":"123456789876****",
   "instanceId":"i-abcdefgh12****",
   "Minimum":92.59,
   "Average":99.49
  },
  {
```

```
 "timestamp":1490152980000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":92.86,
 "Average":99.44
},
{
 "timestamp":1490153040000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":91.43,
 "Average":99.36
},
{
 "timestamp":1490153100000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":93.55,
 "Average":99.51
},
{
 "timestamp":1490153160000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":93.1,
 "Average":99.52
},
{
 "timestamp":1490153220000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":92.59,
 "Average":99.42
},
{
 "timestamp":1490153280000,
```

```
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.34
 },
 {
  "timestamp":1490153340000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.86,
  "Average":99.46
 },
 {
  "timestamp":1490153400000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.35
 }
],
"RequestId":"6A5F022D-AC7C-460E-94AE-B9E75083D027",
"Success":true,
"Code":"200"
}
```


1.2.7.2.3. DescribeMetricList

You can call this operation to query the monitoring data of time series metrics of Apsara Stack services in a specified period.

For more information about how to assign values to the Namespace, Project, Metric, Period, and Dimensions parameters for Apsara Stack services, see DescribeMetricMetaList or Major metrics of Apsara Stack services.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeMetricList	The operation that you want to perform. Set the value to DescribeMetricList.
MetricName	String	Yes	cpu_idle	The name of the metric.
Namespace	String	Yes	acs_ecs_dashboard	The namespace of the monitored service. Specify the value in the format of acs_Service name.
Dimensions	String	No	[{"instanceId": "i-abcdefgh12****"}]	The map of the monitored resources. The value is a collection of `key-value` pairs. A typical `key-value` pair is `instanceId:XXXXXX`. The `key` and `value` must be 1 to 64 bytes in length, respectively. If the length exceeds 64 bytes, excess bytes are truncated from the right. The `key` and `value` can contain letters, digits, periods (.), hyphens (-), underscores (_), forward slashes (/), and backslashes (\). Note If you hope to see a key-value map expressed by JSON strings, set `Dimensions` to ordered strings.
EndTime	String	No	2019-01-30 00:10:00	The end of the time range for the query. Supported formats: • Unix timestamp: The value can be a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC. • Time format: The value can be in the YYYY-MM-DDThh:mm:ssZ format.

Parameter	Type	Required	Example	Description
Express	String	No	{"groupby":["userId","instanceId"]}	The expression for real-time computation on the query results. Currently, only the `groupby` expression is supported, which is similar to the `groupby` statement used in databases.
Length	String	No	1000	The number of entries to return on each page. Default value: 1000.
NextToken	String	No	15761485350009dd70bb64cff1f0fff750b08ffff073be5fb1e785e2b020f1a949d5ea14aea7fed82f01dd87311	The pagination cursor. **Note** You do not need to specify this parameter if you only want to obtain entries on the first page. The pagination cursor returned in a response can be used as an input parameter for obtaining entries on the next page. If a response contains no pagination cursors, all the query result has been returned.

Parameter	Type	Required	Example	Description
Period	String	No	60	The time interval at which monitoring data will be queried. Unit: seconds. Valid values: 60, 300, and 900, which represent 1 minute, 5 minutes, and 15 minutes, respectively. Note Set this parameter as required. For example, if you set this parameter to 60, only 1,000 records will be returned, though a total of 1,440 records will be generated. This is because the maximum number of records that can be returned for each query is 1,000. If you set this parameter to 300, 288 records will be returned.
StartTime	String	No	2019-01-30 00:00:00	The beginning of the time range for the query. Supported formats: • Unix timestamp: The value can be a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC. • Time format: The value can be in the YYYY-MM-DDThh:mm:ssZ format. Note The period specified by StartTime and EndTime includes the time point specified by EndTime but does not include the time point specified by StartTime. StartTime must be earlier than EndTime.

Response parameters

Parameter	Type	Example	Description
RequestId	String	3121AE7D-4AFF-4C25-8F1D-C8226EBB1F42	The ID of the request.

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Success	Boolean	true	Indicates whether the request was successful.
NextToken	String	15761441850009dd70bb64cff1f0fff6d0b08ffff073be5fb1e785e2b020f7fed9b5e137bd810a6d6cff5aee816	The pagination cursor. Note If this parameter was not specified, only entries on the first page were returned. If a response contains a pagination cursor, more entries are to be returned. If a response contains no pagination cursors, all the query result has been returned.
Period	String	60	The time interval at which monitoring data was queried. Unit: seconds. Valid values: 60, 300, and 900.
Datapoints	String	[{"timestamp":1548777660000,"userId":"123","instanceId":"i-abc","Minimum":9.92,"Average":9.92,"Maximum":9.92}]	The list of monitoring data.
Message	String	Success	The error message.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeMetricList
&MetricName=cpu_idle
&Namespace=acs_ecs_dashboard
&<Common request parameters>
```

Sample success responses

XML format

```
<Period>60</Period>
<Datapoints>
  <timestamp>1490152860000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.1</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.52</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490152920000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.59</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.49</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490152980000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.86</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.44</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153040000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.43</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.36</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153100000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.55</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
```

```
    <Average>99.51</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153160000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>93.1</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.52</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153220000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>92.59</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.42</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153280000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>91.18</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.34</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153340000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>92.86</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.46</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153400000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>91.18</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.35</Average>
```

```
</Datapoints>
<RequestId>6A5F022D-AC7C-460E-94AE-B9E75083D027</RequestId>
<Success>true</Success>
<Code>200</Code>
```

`JSON` format

```
{
 "Period":"60",
 "Datapoints":[
  {
   "timestamp":1490152860000,
   "Maximum":100,
   "userId":"123456789876****",
   "instanceId":"i-abcdefgh12****",
   "Minimum":93.1,
   "Average":99.52
  },
  {
   "timestamp":1490152920000,
   "Maximum":100,
   "userId":"123456789876****",
   "instanceId":"i-abcdefgh12****",
   "Minimum":92.59,
   "Average":99.49
  },
  {
   "timestamp":1490152980000,
   "Maximum":100,
   "userId":"123456789876****",
   "instanceId":"i-abcdefgh12****",
   "Minimum":92.86,
   "Average":99.44
  },
  {
   "timestamp":1490153040000,
   "Maximum":100,
   "userId":"123456789876****",
   "instanceId":"i-abcdefgh12****",
   "Minimum":91.43,
```

```
 "Average":99.36
},
{
 "timestamp":1490153100000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":93.55,
 "Average":99.51
},
{
 "timestamp":1490153160000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":93.1,
 "Average":99.52
},
{
 "timestamp":1490153220000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":92.59,
 "Average":99.42
},
{
 "timestamp":1490153280000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":91.18,
 "Average":99.34
},
{
 "timestamp":1490153340000,
 "Maximum":100,
 "userId":"123456789876****",
 "instanceId":"i-abcdefgh12****",
 "Minimum":92.86,
 "Average":99.46
```

```
 },
 {
  "timestamp":1490153400000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.35
 }
],
"RequestId":"6A5F022D-AC7C-460E-94AE-B9E75083D027",
"Success":true,
"Code":"200"
}
```


1.2.7.2.4. DescribeMetricMetaList

You can call this operation to query the descriptions of time series metrics that are supported in CloudMonitor.

This operation is usually used together with DescribeMetricList and DescribeMetricLast to query monitoring data.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeMetricMetaList	The operation that you want to perform. Set the value to DescribeMetricMetaList.
Namespace	String	Yes	acs_kvstore	The namespace of the monitored service. You can call the DescribeProjectMeta operation to query the namespace of each service.

Parameter	Type	Required	Example	Description
Labels	String	No	[{"name":"productCategory","value":"kvstore_old"}]	The tags of the metric. You must specify the tags in the following format: [{"name":"Tag name","value":"Tag value"},{"name":"Tag name","value":"Tag value"}] . The available tag names are as follows: • metricCategory: the category of the metrics. • alertEnable: specifies whether to enable alerts. • alertUnit: the unit of the metric in the alert. • unitFactor: the factor for metric unit conversion. • minAlertPeriod: the minimum time interval to raise a new alert. • productCategory: the category of the service.
MetricName	String	No	CPUUtilization	The name of the metric. The metric with the specified name is matched.
PageNumber	Integer	No	1	The number of the page to return. Default value: 1.
PageSize	Integer	No	30	The number of entries to return on each page. Default value: 30.

Response parameters

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request was successful.
Code	String	200	The HTTP status code. The code 200 indicates that the call was successful.

Parameter	Type	Example	Description
RequestId	String	0CCE0AF0-053C-4B13-A583-DC9A85785D49	The ID of the request, which can be used for troubleshooting.
Message	String	success	The error message.
Resources			The information about metrics found.
Resource			The information about a metric found.
Description	String	CPUUtilization	The description of the metric.
Dimensions	String	instanceId	The dimensions of the metric. Multiple dimensions are separated with commas (,), for example, userId,instanceId.
Labels	String	[{\"name\":\"alertUnit\",\"value\":\"Bytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"instanceType\",\"value\":\"disaster\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"productCategory\",\"value\":\"kvstore_old\"}]	The tags of the metric. The value is a JSON array string. The array can include repeated tag names. Sample value: `[{"name":"Tag name","value":"Tag value"}]`. The available tag names are as follows: • metricCategory: the category of the metrics. • alertEnable: indicates whether the alert is enabled. • alertUnit: the unit of the metric in the alert. • unitFactor: the factor for metric unit conversion. • minAlertPeriod: the minimum time interval to raise a new alert. • productCategory: the category of the service.
MetricName	String	CPUUtilization	The name of the metric.
Namespace	String	acs_kvstore	The namespace of the monitored service. The namespace is usually in the format of acs_Service name.

Parameter	Type	Example	Description
Periods	String	60,300	The statistical period of the metric. Multiple periods are separated with commas (,), for example, 15,60,900.
Statistics	String	Average,Minimum, Maximum	The statistical method of the metric. Multiple statistical methods are separated with commas (,), for example, Average,Minimum,Maximum.
Unit	String	%	The unit of the metric.
TotalCount	String	12	The total number of returned records.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeMetricMetaList
&Namespace=acs_ecs_dashboard
&<Common request parameters>
```

Sample success responses

> **Note** The response content varies according to the services that are actually monitored.

`XML` format

```
<DescribeMetricMetaListResponse>
  <TotalCount>120</TotalCount>
  <RequestId>EA94B362-973C-4D6B-BE94-6774E18915C2</RequestId>
  <Success>true</Success>
  <Code>200</Code>
  <Resources>
    <Resource>
      <Description>ECS.CPUUtilization</Description>
      <Statistics>Average,Minimum,Maximum</Statistics>
      <MetricName>CPUUtilization</MetricName>
      <Labels>[{"name":"alertUnit","value":"%"},{"name":"alertDefault","value":"80"},{"name":"minAlertPeriod","value":"60"},{"name":"metricCategory","value":"instanceId"},{"name":"is_alarm","value":"true"}]</Labels>
      <Dimensions>userId,instanceId</Dimensions>
```

<Dimensions>userId,instanceId</Dimensions>
<Namespace>acs_ecs_dashboard</Namespace>
<Periods>60,300</Periods>
<Unit>%</Unit>
</Resource>
<Resource>
<Description></Description>
<Statistics>Average,Maximum,Minimum</Statistics>
<MetricName>DiskReadBPS</MetricName>
<Labels>[{"name":"alertUnit","value":"Mbytes/s"},{"name":"minAlertPeriod","value":"60"},{"name":"unitFactor","value":"1048576"},{"name":"metricCategory","value":"instanceId"},{"name":"is_alarm","value":"true"}]</Labels>
<Dimensions>userId,instanceId</Dimensions>
<Namespace>acs_ecs_dashboard</Namespace>
<Periods>60,300</Periods>
<Unit>bytes/s</Unit>
</Resource>
<Resource>
<Description>The read IOPS of the system disk</Description>
<Statistics>Average,Minimum,Maximum</Statistics>
<MetricName>DiskReadIOPS</MetricName>
<Labels>[{"name":"alertUnit","value":"Count/Second"},{"name":"minAlertPeriod","value":"60"},{"name":"metricCategory","value":"instanceId"},{"name":"is_alarm","value":"true"}]</Labels>
<Dimensions>userId,instanceId</Dimensions>
<Namespace>acs_ecs_dashboard</Namespace>
<Periods>60,300</Periods>
<Unit>Count/Second</Unit>
</Resource>
<Resource>
<Description></Description>
<Statistics>Average,Maximum,Minimum</Statistics>
<MetricName>DiskWriteBPS</MetricName>
<Labels>[{"name":"alertUnit","value":"Mbytes/s"},{"name":"minAlertPeriod","value":"60"},{"name":"unitFactor","value":"1048576"},{"name":"metricCategory","value":"instanceId"},{"name":"is_alarm","value":"true"}]</Labels>
<Dimensions>userId,instanceId</Dimensions>
<Namespace>acs_ecs_dashboard</Namespace>
<Periods>60,300</Periods>
<Unit>bytes/s</Unit>
</Resource>
<Resource>

```
        <Description>The write IOPS of the system disk</Description>
        <Statistics>Average,Minimum,Maximum</Statistics>
        <MetricName>DiskWriteIOPS</MetricName>
        <Labels>[{"name":"alertUnit","value":"Count/Second"},{"name":"minAlertPeriod","value":"60"},{"
name":"metricCategory","value":"instanceId"},{"name":"is_alarm","value":"true"}]</Labels>
        <Dimensions>userId,instanceId</Dimensions>
        <Namespace>acs_ecs_dashboard</Namespace>
        <Periods>60,300</Periods>
        <Unit>Count/Second</Unit>
      </Resource>
      <Resource>
        <Description></Description>
        <Statistics>Average,Minimum,Maximum</Statistics>
        <MetricName>GroupCPUUtilization</MetricName>
        <Labels>[{"name":"alertUnit","value":"%"},{"name":"minAlertPeriod","value":"60"},{"name":"metri
cCategory","value":"groupId"}]</Labels>
        <Dimensions>groupId</Dimensions>
        <Namespace>acs_ecs_dashboard</Namespace>
        <Periods>60,300</Periods>
        <Unit>%</Unit>
      </Resource>
      <Resource>
        <Description></Description>
        <Statistics>Average,Minimum,Maximum</Statistics>
        <MetricName>GroupDiskReadBPS</MetricName>
        <Labels>[{"name":"alertUnit","value":"Mbytes/s"},{"name":"minAlertPeriod","value":"60"},{"name
":"unitFactor","value":"1048576"},{"name":"metricCategory","value":"groupId"}]</Labels>
        <Dimensions>groupId</Dimensions>
        <Namespace>acs_ecs_dashboard</Namespace>
        <Periods>60,300</Periods>
        <Unit>bytes/s</Unit>
      </Resource>
      <Resource>
        <Description></Description>
        <Statistics>Average,Minimum,Maximum</Statistics>
        <MetricName>GroupDiskReadIOPS</MetricName>
        <Labels>[{"name":"alertUnit","value":"Count/Second"},{"name":"minAlertPeriod","value":"60"},{"
name":"metricCategory","value":"groupId"}]</Labels>
        <Dimensions>groupId</Dimensions>
        <Namespace>acs_ecs_dashboard</Namespace>
        <Periods>60,300</Periods>
```

```
        <Unit>Count/Second</Unit>
      </Resource>
      <Resource>
        <Description></Description>
        <Statistics>Average,Minimum,Maximum</Statistics>
        <MetricName>GroupDiskWriteBPS</MetricName>
        <Labels>[{"name":"alertUnit","value":"Mbytes/s"},{"name":"minAlertPeriod","value":"60"},{"name
":"unitFactor","value":"1048576"},{"name":"metricCategory","value":"groupId"}]</Labels>
        <Dimensions>groupId</Dimensions>
        <Namespace>acs_ecs_dashboard</Namespace>
        <Periods>60,300</Periods>
        <Unit>bytes/s</Unit>
      </Resource>
      <Resource>
        <Description></Description>
        <Statistics>Average,Minimum,Maximum</Statistics>
        <MetricName>GroupDiskWriteIOPS</MetricName>
        <Labels>[{"name":"alertUnit","value":"Count/Second"},{"name":"minAlertPeriod","value":"60"},{"
name":"metricCategory","value":"groupId"}]</Labels>
        <Dimensions>groupId</Dimensions>
        <Namespace>acs_ecs_dashboard</Namespace>
        <Periods>60,300</Periods>
        <Unit>Count/Second</Unit>
      </Resource>
    </Resources>
</DescribeMetricMetaListResponse>
```

`JSON` format

```
{
 "TotalCount":1853,
 "RequestId":"CDE9EAFF-D54E-4024-BBFC-B0AAC883143B",
 "Success":true,
 "Code":200,
 "Resources":{
  "Resource":[
   {
    "Description": "The disk size",
    "MetricName":"ads.diskSize",
    "Statistics":"Average,Minimum,Maximum",
    "Labels":"[{\"name\":\"minAlertPeriod\",\"value\":\"300\"},{\"name\":\"alertDefault\",\"value\":\"\"},
```

{\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\",\"value\":\"Mbytes\"},{\"name\":\"productCategory\",\"value\":\"ads\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"metricCategory\",\"value\":\"workerId\"}]",
"Dimensions":"userId,instanceId,tableSchema,workerId",
"Namespace":"acs_ads",
"Periods":"300",
"Unit":"Mbytes"
},
{
"Description": "The used space of the disk",
"MetricName":"ads.diskUsed",
"Statistics":"Average,Minimum,Maximum",
"Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"300\"},{\"name\":\"metricCategory\",\"value\":\"workerId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
"Dimensions":"userId,instanceId,tableSchema,workerId",
"Namespace":"acs_ads",
"Periods":"300",
"Unit":"Mbytes"
},
{
"Description":"The disk usage",
"MetricName":"ads.diskUsedPercent",
"Statistics":"Average,Minimum,Maximum",
"Labels":"[{\"name\":\"alertUnit\",\"value\":\"%\"},{\"name\":\"minAlertPeriod\",\"value\":\"300\"},{\"name\":\"metricCategory\",\"value\":\"workerId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
"Dimensions":"userId,instanceId,tableSchema,workerId",
"Namespace":"acs_ads",
"Periods":"300",
"Unit":"%"
},
{
"Description": "The number of unconsumed messages in the queue",
"MetricName":"QueueMessageAccumulation",
"Statistics":"Maximum",
"Labels":"[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"alertDefault\",\"value\":\"\"},{\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\",\"value\":\"count/min\"},{\"name\":\"productCategory\",\"value\":\"amqp\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"metricCategory\",\"value\":\"queue\"}]",
"Dimensions":"userId,regionId,vhostName,queueName",
"Namespace":"acs_amqp",

```
  "Periods":"60,300",
  "Unit":"count/min"
 },
 {
  "Description": "The number of new messages in the queue per minute",
  "MetricName":"QueueMessageInput",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"alertDefault\",\"value\":\"\"},{
\"n ame\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\",\"value\":\"count/min\"},{\"name\":\"
productCategory\",\"value\":\"amqp\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"metricCa
tegory\",\"value\":\"queue\"}]",
  "Dimensions":"userId,regionId,vhostName,queueName",
  "Namespace":"acs_amqp",
  "Periods":"60,300",
  "Unit":"count/min"
 },
 {
  "Description": "The number of unconsumed messages in the queue",
  "MetricName":"QueueMessageOutput",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"alertDefault\",\"value\":\"\"},{
\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\",\"value\":\"count/min\"},{\"name\":\"p
roductCategory\",\"value\":\"amqp\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"metricCat
egory\",\"value\":\"queue\"}]",
  "Dimensions":"userId,regionId,vhostName,queueName",
  "Namespace":"acs_amqp",
  "Periods":"60,300",
  "Unit":"count/min"
 },
 {
  "Description":"The number of messages generated by the instance per minute",
  "MetricName":"VhostMessageInput",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"alertDefault\",\"value\":\"\"},{
\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\",\"value\":\"count/min\"},{\"name\":\"p
roductCategory\",\"value\":\"amqp\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"metricCat
egory\",\"value\":\"vhost\"}]",
  "Dimensions":"userId,regionId,vhostName",
  "Namespace":"acs_amqp",
  "Periods":"60,300",
  "Unit":"count/min"
```

```
  },
  {
   "Description":"The number of messages consumed by the instance per minute",
   "MetricName":"VhostMessageOutput",
   "Statistics":"Value",
   "Labels":"[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"alertDefault\",\"value\":\"\"},{
\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\",\"value\":\"count/min\"},{\"name\":\"p
roductCategory\",\"value\":\"amqp\"},{\"name\":\"is_alarm\",\"value\":\"true\"},{\"name\":\"metricCat
egory\",\"value\":\"vhost\"}]",
   "Dimensions":"userId,regionId,vhostName",
   "Namespace":"acs_amqp",
   "Periods":"60,300",
   "Unit":"count/min"
  },
  {
   "Description":"",
   "MetricName":"Latency",
   "Statistics":"Average",
   "Labels":"[{\"name\":\"alertUnit\",\"value\":\"ms\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\
"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
   "Dimensions":"userId,region,apiUid",
   "Namespace":"acs_apigateway_dashboard",
   "Periods":"60,300,900",
   "Unit":"ms"
  },
  {
   "Description":"",
   "MetricName":"SumQPS",
   "Statistics":"Count",
   "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
   "Dimensions":"userId,region,apiUid",
   "Namespace":"acs_apigateway_dashboard",
   "Periods":"60,300,900",
   "Unit":"Count"
  },
  {
   "Description":"",
   "MetricName":"TrafficRX",
   "Statistics":"Sum",
   "Labels":"[{\"name\":\"alertUnit\",\"value\":\"KBytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
```

```
    "Labels": [{\"name\":\"alertUnit\",\"value\":\"KBytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
    "Dimensions":"userId,region,apiUid",
    "Namespace":"acs_apigateway_dashboard",
    "Periods":"60,300,900",
    "Unit":"KBytes"
   },
   {
    "Description":"",
    "MetricName":"TrafficTX",
    "Statistics":"Sum",
    "Labels":"[{\"name\":\"alertUnit\",\"value\":\"KBytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
    "Dimensions":"userId,region,apiUid",
    "Namespace":"acs_apigateway_dashboard",
    "Periods":"60,300,900",
    "Unit":"KBytes"
   },
   {
    "Description":"",
    "MetricName":"code2XX",
    "Statistics":"Value",
    "Labels":"[{\"name\":\"alertUnit\",\"value\":\"count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
    "Dimensions":"userId,region,apiUid",
    "Namespace":"acs_apigateway_dashboard",
    "Periods":"60,300",
    "Unit":"count"
   },
   {
    "Description":"",
    "MetricName":"code4XX",
    "Statistics":"Value",
    "Labels":"[{\"name\":\"alertUnit\",\"value\":\"count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"tru\"}]",
    "Dimensions":"userId,region,apiUid",
    "Namespace":"acs_apigateway_dashboard",
    "Periods":"60,300",
    "Unit":"count"
   },
   {
```

```
  "Description":"",
  "MetricName":"code5XX",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,region,apiUid",
  "Namespace":"acs_apigateway_dashboard",
  "Periods":"60,300",
  "Unit":"count"
 },
 {
  "Description":"",
  "MetricName":"net_rx.Pkgs",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"pps\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{
\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_bandwidth_package",
  "Periods":"60,300,600",
  "Unit":"pps"
 },
 {
  "Description":"",
  "MetricName":"net_rx.rate",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbits/s\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"unitFactor\",\"value\":\"1048576\"},{\"name\":\"metricCategory\",\"value\":\"instanceI
d\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_bandwidth_package",
  "Periods":"60,300,600",
  "Unit":"bits/s"
 },
 {
  "Description":"",
  "MetricName":"net_tx.Pkgs",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"pps\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{
\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_bandwidth_package",
```

```
  "Periods":"60,300,600",
  "Unit":"pps"
 },
 {
  "Description":"",
  "MetricName":"net_tx.rate",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbits/s\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"unitFactor\",\"value\":\"1048576\"},{\"name\":\"metricCategory\",\"value\":\"instanceI
d\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_bandwidth_package",
  "Periods":"60,300,600",
  "Unit":"bits/s"
 },
 {
  "Description":"",
  "MetricName":"net_tx.ratePercent",
  "Statistics":"Value",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"%\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"
name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_bandwidth_package",
  "Periods":"60,300,600",
  "Unit":"%"
 },
 {
  "Description":"The maximum network traffic per unit time",
  "MetricName":"BPS",
  "Statistics":"Average,Minimum,Maximum",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbits/s\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"unitFactor\",\"value\":\"1000000\"},{\"name\":\"metricCategory\",\"value\":\"instanceI
d\"},{\"name\":\"unitFactor\",\"value\":\"1000\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_cdn",
  "Periods":"60,300",
  "Unit":"bits/s"
 },
 {
  "Description":"",
  "MetricName":"GroupBPS"
```

```
  "MetricName":"GroupBPS",
  "Statistics":"Sum",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbits/s\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"unitFactor\",\"value\":\"1000000\"},{\"name\":\"metricCategory\",\"value\":\"groupId\"}
,{\"name\":\"unitFactor\",\"value\":\"1000\"}]",
  "Dimensions":"userId,groupId",
  "Namespace":"acs_cdn",
  "Periods":"60,300",
  "Unit":"bits/s"
 },
 {
  "Description":"",
  "MetricName":"GroupInternetOut",
  "Statistics":"Sum",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"unitFactor\",\"value\":\"1000000\"},{\"name\":\"metricCategory\",\"value\":\"groupId\"}
,{\"name\":\"unitFactor\",\"value\":\"1000\"}]",
  "Dimensions":"userId,groupId",
  "Namespace":"acs_cdn",
  "Periods":"60,300",
  "Unit":"bytes"
 },
 {
  "Description":"The downstream traffic",
  "MetricName":"InternetOut",
  "Statistics":"Sum",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Mbytes\"},{\"name\":\"minAlertPeriod\",\"value\":\"60
\"},{\"name\":\"unitFactor\",\"value\":\"1000000\"},{\"name\":\"metricCategory\",\"value\":\"instanceI
d\"},{\"name\":\"unitFactor\",\"value\":\"1000\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_cdn",
  "Periods":"60,300",
  "Unit":"bytes"
 },
 {
  "Description": "The average number of accesses in the specified time range",
  "MetricName":"QPS",
  "Statistics":"Average,Minimum,Maximum",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
```

```
    "Namespace":"acs_cdn",
    "Periods":"60,300",
    "Unit":"Count"
   },
   {
    "Description":"",
    "MetricName":"UserQPS",
    "Statistics":"Average",
    "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Count\"},{\"name\":\"alertDefault\",\"value\":\"Count
\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"metricCategory\",\"value\":\"userId\"}]",
    "Dimensions":"userId",
    "Namespace":"acs_cdn",
    "Periods":"60,300",
    "Unit":""
   },
   {
    "Description":"",
    "MetricName":"Usercode4xx",
    "Statistics":"Average",
    "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"userId\"}]",
    "Dimensions":"userId",
    "Namespace":"acs_cdn",
    "Periods":"60",
    "Unit":"Count"
   },
   {
    "Description":"",
    "MetricName":"Usercode5xx",
    "Statistics":"Average",
    "Labels":"[{\"name\":\"alertUnit\",\"value\":\"Count\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"
},{\"name\":\"metricCategory\",\"value\":\"userId\"}]",
    "Dimensions":"userId",
    "Namespace":"acs_cdn",
    "Periods":"60",
    "Unit":"Count"
   },
   {
    "Description":"",
    "MetricName":"UserhitRate",
    "Statistics":"Average",
```

```
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"%\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"
name\":\"metricCategory\",\"value\":\"userId\"}]",
  "Dimensions":"userId",
  "Namespace":"acs_cdn",
  "Periods":"60",
  "Unit":"%"
 },
 {
  "Description": "The percentage of 4XX HTTP status codes over total status codes in a specified time
range",
  "MetricName":"code4xx",
  "Statistics":"Average,Minimum,Maximum",
  "Labels":"[{\"name\":\"alertUnit\",\"value\":\"%\"},{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"
name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is_alarm\",\"value\":\"true\"}]",
  "Dimensions":"userId,instanceId",
  "Namespace":"acs_cdn",
  "Periods":"60,300",
  "Unit":"%"
 }
 ]
}
}
```


1.2.7.3. Alert rules for time series metrics

1.2.7.3.1. DescribeMetricRuleList

You can call this operation to query alert rules.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeMetricRuleList	The operation that you want to perform. Set the value to DescribeMetricRuleList.

Parameter	Type	Required	Example	Description
Namespace	String	No	acs_ecs_dashboard	The namespace of the monitored service. Note For more information, see DescribeSystemEventMetaList or Major metrics of Apsara Stack services.
MetricName	String	No	cpu_total	The name of the metric. Note For more information, see DescribeSystemEventMetaList or Major metrics of Apsara Stack services.
EnableState	Boolean	No	true	Specifies whether to query enabled or disabled alert rules. Valid values: • true: indicates that enabled alert rules will be queried. • false: indicates that disabled alert rules will be queried. This parameter is not specified by default, indicating that both enabled and disabled rules will be queried.
Page	String	No	1	The number of the page to return. Default value: 1.
PageSize	String	No	10	The number of entries to return on each page. Default value: 10.
AlertState	String	No	ALARM	The status of alert rules to query. Valid values: • OK: Alert rules with no active alerts will be queried. • ALARM: Alert rules with active alerts will be queried. • INSUFFICIENT_DATA: Alert rules with no data will be queried.

Parameter	Type	Required	Example	Description
Dimensions	String	No	{"instanceId":"i-xy123****"}	The map of the monitored resources. The value is a collection of `key-value` pairs. A typical `key-value` pair is `instanceId:XXXXXX`. The `key` and `value` must be 1 to 64 bytes in length, respectively. If the length exceeds 64 bytes, excess bytes are truncated from the right. The `key` and `value` can contain letters, digits, periods (.), hyphens (-), underscores (_), forward slashes (/), and backslashes (\). **Note** If you hope to see a key-value map expressed by JSON strings, set `Dimensions` to ordered strings.
RuleName	String	No	ECS CPU alert rule	The name of the alert rule. **Note** This parameter supports fuzzy match.
GroupId	String	No	123456	The ID of the application group.
RuleIds	String	No	a151cd6023eacee2f0978e03863cc1697c8950812****	The ID of the alert rule. **Note** You can specify a maximum of 20 alert rule IDs separated with commas (,) in this parameter.

Response parameters

Parameter	Type	Example	Description
RequestId	String	0E657631-CD6C-4C24-9637-98D000B9272C	The ID of the request.

Parameter	Type	Example	Description
Code	Integer	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Total	String	21	The total number of entries returned.
Success	Boolean	true	Indicates whether the request was successful.
Alarms	Array		The list of alert rules.
Alarm			The information about an alert rule.
AlertState	String	OK	The status of the alert rule. Valid values: • OK: There are no active alerts triggered based on the alert rule. • ALARM: There is at least one alert triggered based on the alert rule. • INSUFFICIENT_DATA: No data is available.
ContactGroups	String	Default alert contacts	The information about the alert contacts.

Parameter	Type	Example	Description
Dimensions	String	[{}]	The map of the monitored resources. The value is a collection of `key-value` pairs. A typical `key-value` pair is `instanceId:XXXXXX`. The `key` and `value` are 1 to 64 bytes in length, respectively. If the length exceeds 64 bytes, excess bytes are truncated from the right. The `key` and `value` may contain letters, digits, periods (.), hyphens (-), underscores (_), forward slashes (/), and backslashes (\). **Note** If you set `Dimensions` to ordered strings in the request, the value of this parameter is a key-value map expressed by JSON strings.
EffectiveInterval	String	00:00-23:59	The time period during which the alert rule is effective.
EnableState	Boolean	true	Indicates whether the alert rule was enabled. Valid values: • true: indicates that the alert rule was enabled. • false: indicates that the alert rule was disabled. This parameter is not specified in the request by default. In this case, both enabled and disabled rules are returned.
Escalations			The conditions for triggering different levels of alerts.
Critical			The conditions for triggering critical-level alerts.

Parameter	Type	Example	Description
ComparisonOperator	String	GreaterThanOrEqualToThreshold	The comparison operator of the threshold for critical-level alerts. Valid values: • GreaterThanOrEqualToThreshold: greater than or equal to • GreaterThanThreshold: greater than • LessThanOrEqualToThreshold: less than or equal to • LessThanThreshold: less than • NotEqualToThreshold: not equal to • GreaterThanYesterday: greater than the metric value at the same time yesterday • LessThanYesterday: less than the metric value at the same time yesterday • GreaterThanLastWeek: greater than the metric value at the same time last week • LessThanLastWeek: less than the metric value at the same time last week • GreaterThanLastPeriod: greater than the metric value in the last monitoring cycle • LessThanLastPeriod: less than the metric value in the last monitoring cycle
Statistics	String	Average	The statistical method for critical-level alerts.
Threshold	String	90	The threshold for critical-level alerts.
Times	String	3	The consecutive number of times for which the metric value exceeds the threshold for critical-level alerts before a critical-level alert is triggered.
Info			The conditions for triggering info-level alerts.

Parameter	Type	Example	Description
ComparisonOperator	String	GreaterThanOrEqualToThreshold	The comparison operator of the threshold for info-level alerts. Valid values: • GreaterThanOrEqualToThreshold: greater than or equal to • GreaterThanThreshold: greater than • LessThanOrEqualToThreshold: less than or equal to • LessThanThreshold: less than • NotEqualToThreshold: not equal to • GreaterThanYesterday: greater than the metric value at the same time yesterday • LessThanYesterday: less than the metric value at the same time yesterday • GreaterThanLastWeek: greater than the metric value at the same time last week • LessThanLastWeek: less than the metric value at the same time last week • GreaterThanLastPeriod: greater than the metric value in the last monitoring cycle • LessThanLastPeriod: less than the metric value in the last monitoring cycle
Statistics	String	Average	The statistical method for info-level alerts.
Threshold	String	90	The threshold for info-level alerts.
Times	String	1	The consecutive number of times for which the metric value exceeds the threshold for info-level alerts before an info-level alert is triggered.
Warn			The conditions for triggering warn-level alerts.

Parameter	Type	Example	Description
ComparisonOperator	String	GreaterThanOrEqualToThreshold	The comparison operator of the threshold for warn-level alerts. Valid values: • GreaterThanOrEqualToThreshold: greater than or equal to • GreaterThanThreshold: greater than • LessThanOrEqualToThreshold: less than or equal to • LessThanThreshold: less than • NotEqualToThreshold: not equal to • GreaterThanYesterday: greater than the metric value at the same time yesterday • LessThanYesterday: less than the metric value at the same time yesterday • GreaterThanLastWeek: greater than the metric value at the same time last week • LessThanLastWeek: less than the metric value at the same time last week • GreaterThanLastPeriod: greater than the metric value in the last monitoring cycle • LessThanLastPeriod: less than the metric value in the last monitoring cycle
Statistics	String	Average	The statistical method for warn-level alerts.
Threshold	String	90	The threshold for warn-level alerts.
Times	String	3	The consecutive number of times for which the metric value exceeds the threshold for warn-level alerts before a warn-level alert is triggered.
GroupId	String	123456	The ID of the application group.

Parameter	Type	Example	Description
GroupName	String	My application group	The name of the application group. Note If the alert rule is associated with an application group, the name of the application group is returned in this parameter.
MailSubject	String	An alert occurs	The subject of the alert notification email.
MetricName	String	cpu_total	The name of the metric.
Namespace	String	acs_ecs_dashboard	The namespace of the monitored service.
NoEffectiveInterval	String	00:00-23:59	The time period during which the alert rule is ineffective.
Period	String	60	The statistical period.
Resources	String	[{\"instanceId\":\"i-a2d5q7pm3f9yr29e****\"},{\"instanceId\":\"i-a2d5q7pm3f9yr29e****\"}	The resources associated with the alert rule.
RuleId	String	a151cd6023eacee2f0978e03863cc1697c895012****	The ID of the alert rule.
RuleName	String	My alert rule	The name of the alert rule.
SilenceTime	String	86400	The mute period during which new alerts are not reported even if the alert trigger conditions are met. Unit: seconds. Default value: 86400 (one day). Minimum value: 3600 (one hour). Only one alert is reported during each mute period even if the metric value consecutively exceeds the alert rule threshold several times.

Parameter	Type	Example	Description
SourceType	String	METRIC	The type of the alert rule. Valid values: METRIC: the alert rule for time series metrics. EVENT: the alert rule for event-type metrics. This type was used in earlier versions and has been discarded.
Webhook	String	http://www.aliyun.com	The callback URL.
Message	String	success	The error message.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeMetricRuleList
&<Common request parameters>
```

Sample success responses

`XML` format

```
<Alarms>
  <Alarm>
    <Period>60</Period>
    <Statistics>Average</Statistics>
    <GroupName>Group name</GroupName>
    <Webhook/>
    <Subject>aaabbb</Subject>
    <EffectiveInterval>00:00-23:59</EffectiveInterval>
    <RuleName>My alert rule</RuleName>
    <NoEffectiveInterval/>
    <GroupId>3596360</GroupId>
    <EvaluationCount>3</EvaluationCount>
    <MetricName>cpu_total</MetricName>
    <Threshold>92</Threshold>
    <State>OK</State>
    <Enable>true</Enable>
    <SilenceTime>86400</SilenceTime>
    <Dimensions>[{}]</Dimensions>
    <RuleId>a151cd6023eacee2f0978e03863cc1697c89508e2b61****</RuleId>
```

```
    <Namespace>acs_ecs_dashboard</Namespace>
    <ContactGroups>Test</ContactGroups>
    <Escalations>
      <Critical>
        <Statistics>Average</Statistics>
        <Threshold>92</Threshold>
        <Times>3</Times>
        <ComparisonOperator>GreaterThanOrEqualToThreshold</ComparisonOperator>
      </Critical>
      <Info/>
      <Warn/>
    </Escalations>
    <ComparisonOperator>&gt;=</ComparisonOperator>
    <Resources>[{"instanceId":"i-a2d5q7pm3f12****"}]</Resources>
  </Alarm>
  <Alarm>
    <Period>60</Period>
    <Statistics>Average</Statistics>
    <GroupName>demo</GroupName>
    <Webhook/>
    <Subject>${serviceType}-${metricName}-${levelDescription}Notification(${dimensions})</Subject>
    <EffectiveInterval>00:00-23:59</EffectiveInterval>
    <RuleName>xxxxxxxxxxxxxxx</RuleName>
    <NoEffectiveInterval/>
    <GroupId>12345</GroupId>
    <EvaluationCount>3</EvaluationCount>
    <MetricName>cpu_total</MetricName>
    <Threshold>90</Threshold>
    <State>OK</State>
    <Enable>true</Enable>
    <SilenceTime>86400</SilenceTime>
    <Dimensions>[{}]</Dimensions>
    <RuleId>alarmRule_uuid_adbe286748952b82c6a4ebe3a07521697bb706****</RuleId>
    <Namespace>acs_ecs_dashboard</Namespace>
    <ContactGroups>rongfei</ContactGroups>
    <Escalations>
      <Critical>
        <Statistics>Average</Statistics>
        <Threshold>6666</Threshold>
        <Times>3</Times>
        <ComparisonOperator>GreaterThanOrEqualToThreshold</ComparisonOperator>
```

```
        <ComparisonOperator> GreaterThanOrEqualThreshold </ComparisonOperator>
        </Critical>
        <Info/>
        <Warn/>
      </Escalations>
      <ComparisonOperator>&gt;=</ComparisonOperator>
      <Resources>[{"instanceId":"host-fTcUci7****"}</Resources>
    </Alarm>
</Alarms>
<RequestId>0F05309F-8863-4B2A-8F52-C6AA00E701B4</RequestId>
<Success>true</Success>
<Code>200</Code>
<Total>102</Total>
```

`JSON` format

```
{
 "RequestId":"0F05309F-8863-4B2A-8F52-C6AA00E701B4",
 "Alarms":{
  "Alarm":[
   {
    "Period":60,
    "Statistics":"Average",
    "GroupName":"Group name",
    "Webhook":"",
    "Subject":"aaabbb",
    "EffectiveInterval":"00:00-23:59",
    "RuleName":"My alert rule",
    "NoEffectiveInterval":"",
    "GroupId":"3596360",
    "EvaluationCount":3,
    "MetricName":"cpu_total",
    "State":"OK",
    "Threshold":"92",
    "Enable":true,
    "SilenceTime":86400,
    "Dimensions":"[{}]",
    "RuleId":"a151cd6023eacee2f0978e03863cc1697c89508e2b61****",
    "Namespace":"acs_ecs_dashboard",
    "ContactGroups":"Test",
    "ComparisonOperator":">=",
    "Escalations":{
```

```
 "Escalations":{
  "Critical":{
   "Statistics":"Average",
   "Threshold":"92",
   "Times":3,
   "ComparisonOperator":"GreaterThanOrEqualToThreshold"
  },
  "Info":{},
  "Warn":{}
 },
 "Resources":"[{\"instanceId\":\"i-a2d5q7pm3f12****\"}]"
},
{
 "Period":60,
 "Statistics":"Average",
 "GroupName":"demo",
 "Webhook":"",
 "Subject":"${serviceType}-${metricName}-${levelDescription}Notification(${dimensions})",
 "EffectiveInterval":"00:00-23:59",
 "RuleName":"xxxxxxxxxxxxxxx",
 "NoEffectiveInterval":"",
 "GroupId":"12345",
 "EvaluationCount":3,
 "MetricName":"cpu_total",
 "State":"OK",
 "Threshold":"90",
 "Enable":true,
 "SilenceTime":86400,
 "Dimensions":"[{}]",
 "RuleId":"alarmRule_uuid_adbe286748952b82c6a4ebe3a07521697bb706****",
 "Namespace":"acs_ecs_dashboard",
 "ContactGroups":"rongfei",
 "ComparisonOperator":">=",
 "Escalations":{
  "Critical":{
   "Statistics":"Average",
   "Threshold":"6666",
   "Times":3,
   "ComparisonOperator":"GreaterThanOrEqualToThreshold"
  },
  "Info":{},
```

```
    "Warn":{}
   },
   "Resources":"[{\"instanceId\":\"host-fTcUci7****\"}"
  }
 ]
},
"Success":true,
"Code":"200",
"Total":102
}
```


1.2.7.3.2. PutResourceMetricRule

You can call this operation to configure an alert rule for a performance metric of a resource.

For more information about example values of the cloud service parameters, such as `Namespace`, `MetricName`, and `Dimensions`, that you must set when configuring an alert rule, see DescribeMetricMetaList or Major metrics of Apsara Stack services.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	PutResourceMetricRule	The operation that you want to perform. Set the value to PutResourceMetricRule.
ContactGroups	String	Yes	Default contact group	The alert contact group. Note Separate multiple groups with commas (,).

Parameter	Type	Required	Example	Description
MetricName	String	Yes	cpu_total	The name of the metric. Note For more information, see DescribeMetricMetaList or Major metrics of Apsara Stack services.
Namespace	String	Yes	acs_ecs_dashboard	The namespace of the monitored service. Note For more information, see DescribeMetricMetaList or Major metrics of Apsara Stack services.
Resources	String	Yes	[{"instanceId":"xxxxxx"}]	The resource for which the alert rule is to be configured. Note For more information, see Dimensions in Major metrics of Apsara Stack services.
RuleId	String	Yes	a151cd6023eacee2f0978e03863cc1697c89508****	The ID of the alert rule. Note The ID is generated by the caller. You can specify an existing ID to modify the corresponding alert rule or specify a new ID to create an alert rule.
RuleName	String	Yes	My alert rule	The name of the alert rule.
Webhook	String	No	http://ww.aliyun.com	The callback URL to which a POST request is sent when an alert is triggered based on the alert rule.

Parameter	Type	Required	Example	Description
EffectiveInterval	String	No	00:00-23:59	The time period during which the alert rule is effective.
NoEffectiveInterval	String	No	00:00-06:00	The time period during which the alert rule is ineffective.
SilenceTime	Integer	No	86400	The mute period during which new alerts are not reported even if the alert trigger conditions are met. Unit: seconds. Default value: 86400 (one day). Note Only one alert is reported during each mute period even if the metric value consecutively exceeds the alert rule threshold several times.
Interval	String	No	60	The interval at which CloudMonitor checks whether the alert rule is triggered. The default value is the highest frequency at which the metric is polled. Unit: seconds. Note For more information about the frequency at which monitoring data is collected, see DescribeMetricMetaList.
EmailSubject	String	No	An alert occurs	The subject of the alert notification email.
Period	String	No	60	The aggregation cycle of monitoring data. The value is an integral multiple of 60. Unit: seconds. The default value is the frequency at which monitoring data is collected.
Escalations.Critical.Statistics	String	No	Average	The statistical method for critical-level alerts.

Parameter	Type	Required	Example	Description
Escalations.Critical.ComparisonOperator	String	No	GreaterThanOrEqualToThreshold	The comparison operator of the threshold for critical-level alerts. Valid values: • `GreaterThanOrEqualToThreshold`: greater than or equal to • `GreaterThanThreshold`: greater than • `LessThanOrEqualToThreshold`: less than or equal to • `LessThanThreshold`: less than • `NotEqualToThreshold`: not equal to • `GreaterThanYesterday`: greater than the metric value at the same time yesterday • `LessThanYesterday`: less than the metric value at the same time yesterday • `GreaterThanLastWeek`: greater than the metric value at the same time last week • `LessThanLastWeek`: less than the metric value at the same time last week • `GreaterThanLastPeriod`: greater than the metric value in the last monitoring cycle • `LessThanLastPeriod`: less than the metric value in the last monitoring cycle
Escalations.Critical.Threshold	String	No	90	The threshold for critical-level alerts.
Escalations.Critical.Times	Integer	No	3	The consecutive number of times for which the metric value exceeds the threshold for critical-level alerts before a critical-level alert is triggered.

Parameter	Type	Required	Example	Description
Escalations.Warn.Statistics	String	No	Average	The statistical method for warn-level alerts.
Escalations.Warn.ComparisonOperator	String	No	GreaterThanOrEqualToThreshold	The comparison operator of the threshold for warn-level alerts. Valid values: Valid values: • `GreaterThanOrEqualToThreshold`: greater than or equal to • `GreaterThanThreshold`: greater than • `LessThanOrEqualToThreshold`: less than or equal to • `LessThanThreshold`: less than • `NotEqualToThreshold`: not equal to • `GreaterThanYesterday`: greater than the metric value at the same time yesterday • `LessThanYesterday`: less than the metric value at the same time yesterday • `GreaterThanLastWeek`: greater than the metric value at the same time last week • `LessThanLastWeek`: less than the metric value at the same time last week • `GreaterThanLastPeriod`: greater than the metric value in the last monitoring cycle • `LessThanLastPeriod`: less than the metric value in the last monitoring cycle
Escalations.Warn.Threshold	String	No	90	The threshold for warn-level alerts.

Parameter	Type	Required	Example	Description
Escalations.Warn.Times	Integer	No	3	The consecutive number of times for which the metric value exceeds the threshold for warn-level alerts before a warn-level alert is triggered.
Escalations.Info.Statistics	String	No	Average	The statistical method for info-level alerts.
Escalations.Info.ComparisonOperator	String	No	GreaterThanOrEqualToThreshold	The comparison operator of the threshold for info-level alerts. Valid values: • `GreaterThanOrEqualToThreshold`: greater than or equal to • `GreaterThanThreshold`: greater than • `LessThanOrEqualToThreshold`: less than or equal to • `LessThanThreshold`: less than • `NotEqualToThreshold`: not equal to • `GreaterThanYesterday`: greater than the metric value at the same time yesterday • `LessThanYesterday`: less than the metric value at the same time yesterday • `GreaterThanLastWeek`: greater than the metric value at the same time last week • `LessThanLastWeek`: less than the metric value at the same time last week • `GreaterThanLastPeriod`: greater than the metric value in the last monitoring cycle • `LessThanLastPeriod`: less than the metric value in the last monitoring cycle

Parameter	Type	Required	Example	Description
Escalations.Info.Threshold	String	No	90	The threshold for info-level alerts.
Escalations.Info.Times	Integer	No	3	The consecutive number of times for which the metric value exceeds the threshold for info-level alerts before an info-level alert is triggered.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Message	String	success	The error message.
RequestId	String	65D50468-ECEF-48F1-A6E1-D952E89D9436	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=PutResourceMetricRule
&ContactGroups=Default contact group
&MetricName=cpu_total
&Namespace=acs_ecs_dashboard
&Resources=[{"instanceId":"xxxxxxx"}]
&RuleId=a151cd6023eacee2f0978e03863cc1697c89508****
&RuleName=My alert rule
&<Common request parameters>
```

Sample success responses

`XML` format

```
<Code>200</Code>
<Message>success</Message>
<RequestId>65D50468-ECEF-48F1-A6E1-D952E89D9436</RequestId>
<Success>true</Success>
```

`JSON` format

```
{
 "Message":"success",
 "RequestId":"65D50468-ECEF-48F1-A6E1-D952E89D9436",
 "Success":true,
 "Code":"200"
}
```

1.2.7.3.3. DescribeMetricRuleCount

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	No	DescribeMetricRuleCount	The operation that you want to perform. Set the value to DescribeMetricRuleCount.
MetricName	String	No	cpu_total	The name of the metric. You can call the DescribeMetricMetaList operation to obtain metric names.
Namespace	String	No	acs_ecs_dashboard	The namespace of the service. You can call the DescribeMetricMetaList operation to obtain namespaces.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. The code 200 indicates that the call was successful.
Message	String	success	The error message.
MetricRuleCount			The number of alert rules in each state.
Alarm	Integer	5	The number of alert rules with active alerts.
Disable	Integer	0	The number of disabled alert rules.
Nodata	Integer	0	The number of alert rules with no data.
Ok	Integer	40	The number of alert rules with no active alerts.
Total	Integer	45	The total number of alert rules.
RequestId	String	FF38D33A-67C1-40EB-AB65-FAEE51EDB644	The ID of the request, which can be used for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeMetricRuleCount
&<Common request parameters>
```

Sample success responses

`XML` format

```
<DescribeMetricRuleCountResponse>
    <Success>true</Success>
    <Code>200</Code>
    <RequestId>FF38D33A-67C1-40EB-AB65-FAEE51EDB644</RequestId>
    <MetricRuleCount>
        <Ok>40</Ok>
        <Disable>0</Disable>
        <Total>45</Total>
        <Nodata>0</Nodata>
        <Alarm>5</Alarm>
    </MetricRuleCount>
</DescribeMetricRuleCountResponse>
```

`JSON` format

```
{
 "RequestId":"FF38D33A-67C1-40EB-AB65-FAEE51EDB644",
 "Success":true,
 "Code":"200",
 "MetricRuleCount":{
  "Ok":40,
  "Disable":0,
  "Nodata":0,
  "Total":45,
  "Alarm":5
 }
}
```

1.2.7.3.4. DeleteMetricRules

You can call this operation to delete one or more alert rules.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Id.N	RepeatList	Yes	ab05733c97b7ce239fb1b53393dc1697c7e12****	The ID of the alert rule.
Action	String	No	DeleteMetricRules	The operation that you want to perform. Set the value to DeleteMetricRules.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. The code 200 indicates that the call was successful.
Message	String	success	The error message.
RequestId	String	E5599964-8D0D-40DC-8E90-27A619384B81	The ID of the request, which can be used for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteMetricRules
&Id.1=ab05733c97b7ce239fb1b53393dc1697c7e12****
&<Common request parameters>
```

Sample success responses

`XML` format

```
<DeleteMetricRulesResponse>
    <Code>200</Code>
    <Message>success</Message>
    <RequestId>FF38D33A-67C1-40EB-AB65-FAEE51EDB644</RequestId>
    <Success>true</Success>
</DeleteMetricRulesResponse>
```

JSON format

```
{
 "Message":"success",
 "RequestId":"E5599964-8D0D-40DC-8E90-27A619384B81",
 "Success":true,
 "Code":"200"
}
```

1.2.7.3.5. EnableMetricRules

You can call this operation to enable one or more alert rules.

Debugging

You can use API Explorer to perform debugging. API Explorer allows you to perform various operations to simplify API usage. For example, you can retrieve APIs, call APIs, and dynamically generate SDK example code.

Request parameters

Parameter	Type	Required	Example	Description
RuleId.N	RepeatList	Yes	ab05733c97b7ce239fb1b53393dc1697c123****	The ID of the alert rule.
Action	String	No	EnableMetricRules	The operation that you want to perform. Set the value to EnableMetricRules.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The status code. A value of 200 indicates that the call is successful.
Message	String	success	The error message.
RequestId	String	FF38D33A-67C1-40EB-AB65-FAEE51EDB644	The request ID for troubleshooting.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the call is successful. A value of true indicates that the call is successful. A value of false indicates that the call has failed.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=EnableMetricRules
&RuleId. 1=ab05733c97b7ce239fb1b53393dc1697c123****
&<Common request parameters>
```

Successful response examples

`XML` format

```
<EnableMetricRulesResponse>
 <Code>200</Code>
 <Message>success</Message>
 <RequestId>FF38D33A-67C1-40EB-AB65-FAEE51EDB644</RequestId>
 <Success>true</Success>
</EnableMetricRulesResponse>
```

`JSON` format

```
{
 "Message":"success",
 "RequestId":"FF38D33A-67C1-40EB-AB65-FAEE51EDB644",
 "Success":true,
 "Code":200
}
```

Error code

View error codes

1.2.7.3.6. DisableMetricRules

You can call this operation to disable one or more alert rules.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DisableMetricRules	The operation that you want to perform. Set the value to DisableMetricRules.
RuleId.N	RepeatList	Yes	ab05733c97b7ce239fb1b53393dc1697c7e012****	The ID of the alert rule. Valid values of N: 1 to 20.

Response parameters

Parameter	Type	Example	Description
Code	String	200	The HTTP status code. The code 200 indicates that the call was successful.
Message	String	success	The error message.
RequestId	String	FF38D33A-67C1-40EB-AB65-FAEE51EDB644	The ID of the request, which can be used for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DisableMetricRules
&RuleId.1=ab05733c97b7ce239fb1b53393dc1697c7e012****
&<Common request parameters>
```

Sample success responses

`XML` format

```
<DisableMetricRulesResponse>
    <Code>200</Code>
    <Message>success</Message>
    <RequestId>FF38D33A-67C1-40EB-AB65-FAEE51EDB644</RequestId>
    <Success>true</Success>
</DisableMetricRulesResponse>
```

`JSON` format

```
{
 "Message":"success",
 "RequestId":"FF38D33A-67C1-40EB-AB65-FAEE51EDB644",
 "Success":true,
 "Code":"200"
}
```

1.2.7.3.7. DescribeAlertingMetricRuleResources

You can call this operation to query the resources with active alerts triggered based on an alert rule.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeAlertingMetricRuleResources	The operation that you want to perform. Set the value to DescribeAlertingMetricRuleResources.
GroupId	String	No	123456	The ID of the application group.
Page	Integer	No	1	The number of the page to return. Default value: 1.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10.
RuleId	String	No	alerRuleId****	The ID of the alert rule. Note You must specify GroupId, RuleId, or both. If you specify both parameters, only entries that meet both conditions are returned.

Response parameters

Parameter	Type	Example	Description
RequestId	String	9B868619-77B4-4623-AD64-6181EA7CF8FA	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Code	Integer	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Total	Integer	1	The total number of entries returned.
Resources	Array		The list of resources to which the alert rule applies.
Resource			The information about a resource to which the alert rule applies.
Enable	String	true	Indicates whether the alert rule was enabled for the resource.

Parameter	Type	Example	Description
GroupId	String	123456	The ID of the application group with which the alert rule was associated. Note If the alert rule is associated with an application group, the ID of the application group is returned in this parameter.
LastAlertTime	String	1554718182000	The last time when an alert was triggered for the resource based on the alert rule. The value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
LastModifyTime	String	1554723762000	The last time when the resource was modified. The value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
MetricValues	String	{"timestamp":1554723720000,"Value":631}	The metric value that triggered an alert based on the alert rule. The value is a JSON string.
Resource	String	userId=177541323192****,taskId=3****,instanceId=i-bp1a0v6yzqrms01k****	The resource for which an alert was triggered based on the alert rule.
RetryTimes	String	3	The consecutive number of times for which the metric value was measured before an alert was triggered.
RuleId	String	alerRuleId****	The ID of the alert rule.
RuleName	String	cpuAlert	The name of the alert rule.

Parameter	Type	Example	Description
StartTime	String	1545901080000	The time when the resource was associated with the alert rule. The value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
Threshold	String	400	The alert threshold.
Message	String	success	The error message.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeAlertingMetricRuleResources
&<Common request parameters>
```

Sample success responses

`XML` format

```
<Total>1</Total>
<Resources>
  <Resource>
    <Enable>true</Enable>
    <Resource>userId=12****,instanceId=i-bp1a012****</Resource>
    <RuleId>alerRuleId****</RuleId>
    <RetryTimes>3</RetryTimes>
    <LastAlertTime>1554718182000</LastAlertTime>
    <MetricValues>{"timestamp":1554723720000,"Value":631}</MetricValues>
    <LastModifyTime>1554723762000</LastModifyTime>
    <StartTime>1545901080000</StartTime>
    <RuleName>cpuAlert</RuleName>
    <Threshold>400</Threshold>
    <GroupId>123456</GroupId>
  </Resource>
</Resources>
<Success>true</Success>
```

`JSON` format

```
{
 "Success":true,
 "Total":1,
 "Resources":{
  "Resource":[
   {
    "LastAlertTime":1554718182000,
    "Resource":"userId=12****,instanceId=i-bp1a012****",
    "Threshold":"400",
    "Enable":true,
    "LastModifyTime":1554723762000,
    "RuleId":"alerRuleId****",
    "MetricValues":"{\"timestamp\":1554723720000,\"Value\":631}",
    "RetryTimes":3,
    "StartTime":1545901080000,
    "RuleName":"cpuAlert",
    "GroupId":"123456"
   }
  ]
 }
}
```


1.2.7.3.8. DeleteMetricRuleResources

You can call this operation to delete resources associated with an alert rule.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteMetricRuleResources	The operation that you want to perform. Set the value to DeleteMetricRuleResources.
Resources	String	Yes	[{"instanceId":"*****ixxxId1"}]	The resources associated with the alert rule. The value is a JSON array.

Parameter	Type	Required	Example	Description
RuleId	String	Yes	rr-bp18017n6iolv****	The ID of the alert rule.

Response parameters

Parameter	Type	Example	Description
RequestId	String	D8A35882-90C6-4F03-BBEB-153C180398EA	The ID of the request.
Success	Boolean	true	Indicates whether the call was successful.
Code	String	200	The HTTP status code. Note The code 200 indicates that the call was successful.
Message	String	success	The error message.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteMetricRuleResources
&Resources=[{"instanceId":"*****ixxxId1"}]
&RuleId=rr-bp18017n6iolv****
&<Common request parameters>
```

Sample success responses

`XML` format

```
<RequestId>0671A721-0D7A-4F11-BB77-2416325D65AB</RequestId>
<Success>true</Success>
<Message>success</Message>
<Code>200</Code>
```

`JSON` format

```
{
   "Message":"success",
   "RequestId":"0671A721-0D7A-4F11-BB77-2416325D65AB",
   "Success":true,
   "Code":"200"
}
```

2.Object Storage Service (OSS)

2.1. Java SDK

2.1.1. Preface

This topic is written for OSS SDK for Java version 3.8.0.

Compatibility

- For SDKs version 3.×.×:
  - API operation: compatible
  - Namespace: compatible
- For SDKs version 2.×.×:
  - API operation: compatible
  - Namespace: compatible
- For SDKs version 1.0.×:
  - 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.

Sample file	Description
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.

2.1.2. 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:

  i. Download the OSS SDK for Java package.
  ii. Decompress the package.
  iii. Copy the *aliyun-sdk-oss-3.8.0.jar* file and all files in the lib directory to your project.
  iv. Right-click your project name in Eclipse. Choose **Properties** > **Java Build Path** > **Add JARs**.
  v. 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:

  i. 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.

2.1.3. 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 requirem
ents.
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 ca
ses, 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 version
s 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, securityToke
n);

// 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

Parameter	Description	Configuration method
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 valu
e: 1024.
conf.setMaxConnections(200);
// Configure the timeout period for data transmission at the socket layer. Unit: milliseconds. Default v
alue: 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: millisecond
s. 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 val
ue: 3.
conf.setMaxErrorRetry(5);
// Configure whether CNAME can be used as an endpoint. By default, CNAME can be used as an endpoi
nt.
conf.setSupportCname(true);
// Configure whether access from second-level domains is enabled. By default, access from second-le
vel 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();
```

2.1.4. 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 cont
ent 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 q
uery 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 m
ay 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 th
e 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();
```

2.1.5. Buckets

2.1.5.1. 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();
```

2.1.5.2. 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());
}
```

2.1.5.3. 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();
```

2.1.5.4. 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();
```

2.1.5.5. 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.

> **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();
```

2.1.5.6. 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 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:

```
// 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();
```

2.1.5.7. 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 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();
```

2.1.5.8. 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 buck
et.
```

```
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 preflig
ht (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 origins 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 specifi
c 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();
```

2.1.5.9. 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:

```
// 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();
```

2.1.5.10. 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();
```

2.1.6. Upload objects

2.1.6.1. 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.

2.1.6.2. 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 pat
h must include the extension of the object name. Example: abc/efg/123.jpg.
PutObjectRequest putObjectRequest = new PutObjectRequest("<yourBucketName>", "<yourObjectN
ame>", 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>", "<yourObjectNam
e>", new File("<yourLocalFile>"));

// Upload the local file.
ossClient.putObject(putObjectRequest);

// Shut down the OSSClient instance.
ossClient.shutdown();
```

2.1.6.3. 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("{\\\"mimeType\\\":${mimeType},\\\"size\\\":${size}}");
        // Configure Content-Type for the callback request.
        callback.setCalbackBodyType(CalbackBodyType.JSON);
        // Configure the custom parameters used to send the callback request. Each custom parameter co
nsists 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-ra
nge\", 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)
```

```
            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();
    }
}
```

2.1.6.4. 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 t
he account has permissions on all API operations. We recommend that you use a RAM user's credential
s to call API operations or perform routine operations and maintenance. To create a RAM user, log on t
o 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>", "<yo
urObjectName>", 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();
```

2.1.6.5. 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();
```

2.1.6.6. 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**
   >
   > - 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.
   > - OSS includes the MD5 hash of part data in the ETag header and returns the MD5 hash to the user.
   > - 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. Example: 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 range, 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 all 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>", "<yourObjectNam
e>", "<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 part
s 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>", "<yourObjectNam
e>", "<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>", "<yourObjectNam
e>", "<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.	ListMultipartUploadsRequest.setPrefix(String prefix)
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.	ListMultipartUploadsRequest.setDelimiter(String delimiter)
maxUploads	Specifies the maximum number of multipart upload tasks to return. The default value is the same as the maximum value, which is 1000.	ListMultipartUploadsRequest.setMaxUploads(Integer maxUploads)
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 uploadIdMarker parameter to specify the start point to list the returned results.	ListMultipartUploadsRequest.setKeyMarker(String keyMarker)

Parameter	Description	Configuration method
uploadIdMarker	UploadIdMarker is used with the keyMarker parameter to specify the start point to list the returned results. If the keyMarker parameter is not configured, the uploadIdMarker parameter is invalid. If the keyMarker parameter is configured, the query result includes: • All objects whose names come after the value of keyMarker. • All multipart upload tasks in which the object names come after the keyMarker value and the upload IDs are greater than the uploadIdMarker value.	ListMultipartUploadsRequest.setUploadIdMarker(String uploadIdMarker)

- 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(buck
etName);
MultipartUploadListing multipartUploadListing = ossClient.listMultipartUploads(listMultipartUploads
Request);

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, listMultipartUploads 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(buck
etName);

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(buck
etName);
// 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();
```

2.1.6.7. 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: " + perce
nt + "%(" + 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 endpoi
nt.
        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 us
er 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.

2.1.6.8. Upload callback

This topic describes how to use upload callback.

The following code provides an example on how to use upload callback:

```
// 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>";
// Specify the IP address of the server to which you want send the callback request. Example: http://oss-demo.aliyuncs.com:23450 or http://127.0.0.1:9090.
String callbackUrl = "<yourCallbackServerUrl>";

// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);

String content = "Hello OSS";
PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, objectName,new ByteArrayInputStream(content.getBytes()));

// Configure upload callback parameters.
Callback callback = new Callback();
callback.setCallbackUrl(callbackUrl);
// Optional. Configure the value of the host field included in the callback request header. This value is the same as the value of the host field specified for your server.
// callback.setCallbackHost("yourCallbackHost");
// Configure the value of the body field included in the callback request.
callback.setCallbackBody("{\\\"mimeType\\\":${mimeType},\\\"size\\\":${size}}");
// Configure Content-Type for the callback request.
callback.setCalbackBodyType(CalbackBodyType.JSON);
// Configure custom parameters for the callback request. Each custom parameter consists of a key and a value. The key must start with x:.
callback.addCallbackVar("x:var1", "value1");
callback.addCallbackVar("x:var2", "value2");
putObjectRequest.setCallback(callback);
```

```
PutObjectResult putObjectResult = ossClient.putObject(putObjectRequest);

// Read the response for the upload callback request.
byte[] buffer = new byte[1024];
putObjectResult.getResponse().getContent().read(buffer);
// You must close the stream after the object is read. Otherwise, connection leaks may occur. Consequ
ently, no connections are available and an exception occurs.
putObjectResult.getResponse().getContent().close();

// Shut down the OSSClient instance.
ossClient.shutdown();
```

2.1.7. Download objects

2.1.7.1. 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.

2.1.7.2. 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 p
ath must include the file extension of the object. For example, set <yourObjectName> to abc/efg/123.j
pg.
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. Ot
herwise, connection leaks may occur. As a result, no connections are available and an exception occur
s.
reader.close();

// Shut down the OSSClient instance.
ossClient.shutdown();
```

2.1.7.3. 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();
```

2.1.7.4. 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 p
ath 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 s
pecified 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. Ot
herwise, connection leaks may occur. As a result, no connections are available and an exception occur
s.
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 fro
m byte 500 to byte 999. The 206 HTTP status code is also returned.
getObjectRequest = new GetObjectRequest(bucketName, objectName);
getObjectRequest.setRange(500, 2000);
getObjectRequest.addHeader("x-oss-range-behavior", "standard");
OSSObject ossObject = ossClient.getObject(getObjectRequest);
ossObject.close();

System.out.println("standard get " + "500~2000 "+ "statusCode:"  + ossObject.getResponse().getStatus
Code());
System.out.println("standard get " + "500~2000 " + "contentLength:" + ossObject.getResponse().getCo
ntentLength());

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.
    getObjectRequest = new GetObjectRequest(bucketName, objectName);
    getObjectRequest.setRange(1000, 2000);
    getObjectRequest.addHeader("x-oss-range-behavior", "standard");
    ossClient.getObject(getObjectRequest);
} catch (OSSException e) {
    System.out.println("standard get " + "1000~2000:" + "error code:" + e.getErrorCode());
}

// Shut down the OSSClient instance.
ossClient.shutdown();
```

2.1.7.5. 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 setUploadFile
partSize	The size of each part. Valid values: 1 byte to 5 GB.	No	Object size/10000	setPartSize
taskNum	The number of parts you need to download simultaneously.	No	1	setTaskNum
enableCheckpoint	Specifies whether to enable resumable upload.	No	false	setEnableCheckpoint

Parameter	Description	Required	Default value	Configuration method
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	downloadFile.ucp (share the same directory with DownloadFile)	setCheckpointFile

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();
```

2.1.7.6. 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.

Parameter	Description
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.

2.1.7.7. 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.

2.1.8. Manage objects

2.1.8.1. 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

2.1.8.2. 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 val
ue 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();
```

2.1.8.3. 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

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 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();
```

2.1.8.4. 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 an
d compares this MD5 hash wih 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 encodin
```

```
g type used to read the object based on the content type of the object. If the content type is not sp
ecified, a content type is generated based on the object name extension. If no extension is availabl
e, 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 t
he 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 OS
S because the account has permissions on all API operations. We recommend that you use a RAM us
er 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 metadat
a.
CopyObjectRequest request = new CopyObjectRequest(sourceBucketName, sourceObjectName, destin
ationBucketName, destinationObjectName);

ObjectMetadata meta = new ObjectMetadata();
// Specify the type of content you want to upload. The browser determines the format and encoding t
ype used to read the object based on the content type of the object. If the content type is not specifie
d, a content type is generated based on the object name extension. If no extension is available, the d
efault 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();
```

2.1.8.5. List objects

Objects in a bucket are sorted alphabetically. You can use ossClient.listObjects to list objects in a bucket.

listObjects supports the following types of parameter formats:

- ObjectListing listObjects(String bucketName): lists objects in a bucket. A maximum of 100 objects can be listed.
- 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(ListObjectsRequest listObjectsRequest): provides multiple filtering features to implement flexible queries.

The following table describes parameters you can configure for ObjectListing.

Parameter	Description	Configuration method
objectSummaries	Specifies the object metadata you want to return.	List<OSSObjectSummary> getObjectSummaries()
prefix	Specifies the prefix that the returned object names must contain.	String getPrefix()
delimiter	Specifies the character used to group objects by name.	String getDelimiter()
marker	Specifies the name of the object after which the list begins.	String getMarker()
maxKeys	Specifies the maximum number of objects that can be listed each time.	int getMaxKeys()
nextMarker	Specifies the name of the object after which the next list begins.	String getNextMarker()
isTruncated	Indicates whether the objects list is truncated. • If all objects are listed, the returned value of this parameter is false. • If only part of objects are listed, the returned value of this parameter is true.	boolean isTruncated()
commonPrefixes	Specifies a set of substrings of 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 in commonPrefixes.	List<String> getCommonPrefixes()

Parameter	Description	Configuration method
encodingType	Specifies the encoding type of the object names in the response.	String getEncodingType()

Simple list

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.

```
// 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 KeyPrefix = "<yourKeyPrefix>";

// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);

// List objects. If you do not set KeyPrifex, all objects in the bucket are listed. If you set KeyPrifex, 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	Configuration method
prefix	Specifies the prefix that returned object names must contain.	setPrefix(String prefix)

Parameter	Description	Configuration method
delimiter	Specifies 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 commonPrefixes.	setDelimiter(String delimiter)
marker	Specifies the name of the object after which the list begins.	setMarker(String marker)
maxKeys	Specifies the maximum number of objects that can be listed each time. The default value is 100. The maximum value is 1000.	setMaxKeys(Integer maxKeys)
encodingType	Specifies the encoding type of the object names in the response. Only URL encoding is supported.	setEncodingType(String encodingType)

- List a specified number of objects

  The following code provides an example on how to list a specified number of objects:

```
// 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);

// Specifies the maximum number of objects that can be listed each time.
final int maxKeys = 200;
// List objects.
ObjectListing objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withMaxKe
ys(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

  The following code provides an example on how to list 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>";

// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);

// Specify a prefix.
final String keyPrefix = "<yourkeyPrefix>";

// List the objects whose names contain the specified prefix. By default, a maximum of 100 objects c
an be 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 come after a specified marker

  The marker parameter specifies the name of the object after which the list begins. The following code provides an example on how to list objects whose names come after a specified marker:

```
// 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);
final String marker = "<yourMarker>";

// List the objects whose names come after the specified marker. By default, a maximum of 100 obje
cts can be listed.
ObjectListing objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withMarke
r(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 by page

  The following code provides an example on how to list all objects in a specified bucket by page. The maximum number of objects that can be listed on each page is specified by maxKeys.

```
// 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);

final int maxKeys = 200;
String nextMarker = null;
ObjectListing objectListing;

do {
    objectListing = ossClient.listObjects(new ListObjectsRequest(bucketName).withMarker(nextMark
er).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:

```
// 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);

// Specify that the maximum number of objects that can be listed on each page is 200.
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 and specify the encoding type of the object names in the response

  An object name that contains any of the following special characters needs to be encoded before it is transmitted. Only URL encoding is supported.

  - Single quotation mark (')
  - Double quotation mark (")
  - Ampersands (&)
  - Angle brackets (<>)

- Pause marker (、)
- Chinese characters

The following code provides an example on how to list objects and specify the encoding type of the object names in the response:

```
// 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);

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 in the response.
    listObjectsRequest.setEncodingType("url");

    objectListing = ossClient.listObjects(listObjectsRequest);

    // Decode the object names.
    for (OSSObjectSummary objectSummary: objectListing.getObjectSummaries()) {
        System.out.println("Key:" + URLDecoder.decode(objectSummary.getKey(), "UTF-8"));
    }

    // Decode the value of commonPrefixes.
    for (String commonPrefixes: objectListing.getCommonPrefixes()) {
        System.out.println("CommonPrefixes:" + URLDecoder.decode(commonPrefixes, "UTF-8"));
    }
```

```
    // Decode the value of nextMarker.
    if (objectListing.getNextMarker() ! = null) {
        nextMarker = URLDecoder.decode(objectListing.getNextMarker(), "UTF-8");
    }
} while (objectListing.isTruncated());

// Shut down the OSSClient instance.
ossClient.shutdown();
```

List objects by folder

OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. A folder is an object whose size is 0 and whose name ends with a forward slash. You can upload and download this object. By default, the OSS console displays objects whose names end with a forward slash (/) as a folder.

You can use the delimiter and prefix parameters to list objects by folder.

- If prefix is set to the name of a folder, objects and subfolders whose names contain the prefix are listed.
- If you also set delimiter to a forward slash (/) in the request, the objects and subfolders whose names start with the specified prefix in the folder are listed. Each subfolder is listed as a single result element in commonPrefixes. The objects and folders in these subfolders are not listed.

For more information about folders, see Folder simulation. For the complete code used to create a folder, visit GitHub.

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 examples show you how to list objects by folder in a bucket.

- List all objects in a bucket

  The following code provides an example on how to list all objects in 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 ListObjectsRequest.
ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);

// List all objects in the bucket.
ObjectListing listing = ossClient.listObjects(listObjectsRequest);

// Traverse all the objects to display them.
System.out.println("Objects:");
for (OSSObjectSummary objectSummary : listing.getObjectSummaries()) {
   System.out.println(objectSummary.getKey());
}

// Traverse all the commonPrefix elements to display them.
System.out.println("CommonPrefixes:");
for (String commonPrefix : listing.getCommonPrefixes()) {
   System.out.println(commonPrefix);
}

// Shut down the OSSClient instance.
ossClient.shutdown();
```

Example output:

```
Objects:
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
oss.jpg

CommonPrefixes:
```

- List all objects in a specified folder

  The following code provides an example on how to list all 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>";

// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);

// Create ListObjectsRequest.
ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);
// Set prefix to fun/ to list all objects in the fun/ folder.
listObjectsRequest.setPrefix("fun/");

// Lists all objects in the fun/ folder.
ObjectListing listing = ossClient.listObjects(listObjectsRequest);

// Traverse the listed objects to display them.
System.out.println("Objects:");
for (OSSObjectSummary objectSummary : listing.getObjectSummaries()) {
   System.out.println(objectSummary.getKey());
}

// Traverse the listed commonPrefix elements to display them.
System.out.println("\nCommonPrefixes:");
for (String commonPrefix : listing.getCommonPrefixes()) {
   System.out.println(commonPrefix);
}

// Shut down the OSSClient instance.
ossClient.shutdown();
```

Example output:

```
Objects:
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg

CommonPrefixes:
```

- List the objects and subfolders in a folder

  The following code provides an example on how to list the objects and subfolders in a specified directory:

```
// 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 ListObjectsRequest.
ListObjectsRequest listObjectsRequest = new ListObjectsRequest(bucketName);

// Set the delimiter to a forward slash (/).
listObjectsRequest.setDelimiter("/");

// List all objects and subfolders in the fun folder.
listObjectsRequest.setPrefix("fun/");

ObjectListing listing = ossClient.listObjects(listObjectsRequest);

// Traverse all objects to display them.
System.out.println("Objects:");
// In the response, objectSummaries lists the objects in the fun folder.
for (OSSObjectSummary objectSummary : listing.getObjectSummaries()) {
    System.out.println(objectSummary.getKey());
}

// Traverse all commonPrefix elements to display them.
System.out.println("\nCommonPrefixes:");
// commonPrefixs lists the subfolders in the fun folder. The fun/movie/001.avi and fun/movie/007.a
vi objects are not listed because they are in the movie subfolder of the fun folder.
for (String commonPrefix : listing.getCommonPrefixes()) {
    System.out.println(commonPrefix);
}

// Shut down the OSSClient instance.
ossClient.shutdown();
```

Example output:

```
Objects:
fun/test.jpg

CommonPrefixes:
fun/movie/
```

- Query 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:

```
// Query the total size of objects in the specified folder.
    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).withM
axKeys(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() {
        // The endpoint of the China (Hangzhou) region is used in this example. Specify the actual end
point.
        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 folder, leave this p
```

```
arameter empty.
        final String keyPrefix = "<YourPrefix>";
        ObjectListing objectListing = null;
        do {
            // By default, a maximum of 100 objects can be listed each time.
            ListObjectsRequest request = new ListObjectsRequest(bucketName).withDelimiter("/").withP
refix(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();
    }
```

2.1.8.6. 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 CSVForm
at()));
        selectObjectRequest.setExpression("select * from ossobject where _4 > 40");
```

```
        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.getObjectCont
ent()));
        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.getObjectCont
ent()));
        while (true) {
            String line = reader.readLine();
            if (line == null) {
                break;
            }
            System.out.println(line);
        }
        reader.close();

        ossClient.deleteObject(bucketName, key);
    }
}
```

2.1.8.7. 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

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 objec
t you want to delete from OSS, and must include the extension of the object name. Example: abc/efg/1
23.jpg.
List<String> keys = new ArrayList<String>();
keys.add("key0");
keys.add("key1");
keys.add("key2");

DeleteObjectsResult deleteObjectsResult = ossClient.deleteObjects(new DeleteObjectsRequest(bucke
tName).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).withKey
s(keys);
      ossClient.deleteObjects(deleteObjectsRequest);
   }

   nextMarker = objectListing.getNextMarker();
} while (objectListing.isTruncated());

// Shut down the OSSClient instance.
ossClient.shutdown();
```

2.1.8.8. 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
CopyObjectResult copyObject(String sourceBucketName, String sourceKey, String destinationBucketName, String destinationKey)	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.
CopyObjectResult copyObject(CopyObjectRequest copyObjectRequest)	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
sourceBucketName	The name of the source bucket.	setSourceBucketName(String sourceBucketName)
sourceKey	The name of the source object.	setSourceKey(String sourceKey)
destinationBucketName	The name of the destination bucket.	setDestinationBucketName(String destinationBucketName)
destinationKey	The name of the destination object.	setDestinationKey(String destinationKey)
newObjectMetadata	The metadata of the destination object.	setNewObjectMetadata(ObjectMetadata newObjectMetadata)
matchingETagConstraints	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.	setMatchingETagConstraints(List<String> matchingETagConstraints)

Parameter	Description	Configuration method
nonmatchingEtagConstraints	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.	setNonmatchingETagConstraints(List<String> nonmatchingEtagConstraints)
unmodifiedSinceConstraint	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.	setUnmodifiedSinceConstraint(Date unmodifiedSinceConstraint)
modifiedSinceConstraint	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.	setModifiedSinceConstraint(Date modifiedSinceConstraint)

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, destination
BucketName, 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, sourceObjec
tName, 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, sourceObjectNam
e);
// 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 t
he metadata of the target object.
InitiateMultipartUploadRequest initiateMultipartUploadRequest = new InitiateMultipartUploadRequest
(destinationBucketName, destinationObjectName);
InitiateMultipartUploadResult initiateMultipartUploadResult = ossClient.initiateMultipartUpload(initiat
eMultipartUploadRequest);
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, de
stinationObjectName);
    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 CompleteMultipartUploadR
equest(
                destinationBucketName, destinationObjectName, uploadId, partETags);
ossClient.completeMultipartUpload(completeMultipartUploadRequest);

// Shut down the OSSClient instance.
ossClient.shutdown();
```

2.1.9. 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 ba
sed on the security token and temporary AccessKey pair (AccessKey ID and AccessKey secret) contain
ed 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) co
ntained in the credential.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToke
n);

GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(bucketName, objectNa
me, 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, requestHead
ers, 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 endpoi
nt.
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, object
Name, 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 endpoi
nt.
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, object
Name, 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();
```

2.1.10. IMG

Image Processing (IMG) provided by OSS is a secure, cost-effective, and highly reliable image processing service that processes 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 create your own image processing service.

Basic features

OSS provides the following IMG 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. Choose **Data Processing** > **Image Processing (IMG)**. Click Create Rule to customize an image processing style.
- Call multiple IMG features by using cascade processing.

Usage of IMG

You can use the standard `HTTP GET` request 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 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,parame_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.

Parameter	Description
action	Operations on images, such as scaling, cropping, and rotating.
param	Parameters that are corresponding to the operations on images.

- Basic operations

  Example: Scale an image to a width of 100 pixels 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
  ```

- Custom styles

  Access IMG anonymously by using a third-level domain in the following format:

  ```
  http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
  ```

  - style: the identifier reserved by the system for the custom style.
  - yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when the custom style is added in the Apsara Stack Cloud Management (ASCM) 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 over HTTPS

  IMG also supports access over 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 add a signature to a URL to access IMG:

```
String endpoint = "<endpoint, such as 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, HttpM
ethod.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 `Date expiration = DateUtil.parseRfc822Date("Wed, 21 Dec 2022 14:20:00 GMT");` to specify the expiration time.

- Access by using SDK

  You can directly use SDK to access and process image objects configured with any ACLs.

  > **Note**
  > - For the complete IMG code, visit GitHub.
  > - You can use the SDK to operate image objects by setting Create Style, HTTPS, and Cascading Processing.

  - Basic operations

Basic IMG operations include query image information, convert image formats, resize, crop, rotate, apply effects, and add watermarks.

```
String endpoint = "<endpoint, such as 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.
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 = new GetObjectRequest(bucketName, key);
request.setProcess(style);
ossClient.getObject(request, new File("example-info.txt"));
ossClient.shutdown();
```

- Custom styles

```
String endpoint = "<endpoint, such as 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 a 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 = "<endpoint, such as 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();
```

2.1.11. 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: OSSException and ClientException. Both are subclasses of RuntimeException.

Examples

```
try {
    // Do some operations with the instance here, such as put object...
    client.putObject(...);
} catch (OSSException oe) {
    System.out.println("Caught an OSSException, which means your request made it to OSS, "
            + "but was rejected with an error response for some reason.") ;
    System.out.println("Error Message: " + oe.getErrorCode());
    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.

OSSException

OSSException 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.

OSSException 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
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

Error code	Error message	HTTP status code
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

Error code	Error message	HTTP status code
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

2.1.12. 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.j
ava:52)
 at org.apache.http.impl.io.DefaultHttpRequestWriterFactory.<init>(DefaultHttpRequestWriterFactory.j
ava:56)
 at org.apache.http.impl.io.DefaultHttpRequestWriterFactory.<clinit>(DefaultHttpRequestWriterFactory
.java:46)
 at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<init>(ManagedHttpClientConnect
ionFactory.java:82)
 at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<init>(ManagedHttpClientConnect
ionFactory.java:95)
 at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<init>(ManagedHttpClientConnect
ionFactory.java:104)
 at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<clinit>(ManagedHttpClientConne
ctionFactory.java:62)
 at org.apache.http.impl.conn.PoolingHttpClientConnectionManager$InternalConnectionFactory.<init>(
PoolingHttpClientConnectionManager.java:572)
 at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init>(PoolingHttpClientConnectio
nManager.java:174)
 at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init>(PoolingHttpClientConnectio
nManager.java:158)
 at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init>(PoolingHttpClientConnectio
nManager.java:149)
 at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.<init>(PoolingHttpClientConnectio
nManager.java:125)
 at com.aliyun.oss.common.comm.DefaultServiceClient.createHttpClientConnectionManager(DefaultSer
viceClient.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.TaskExcutorManager$UploadTask.run(TaskExcutorManager.java:114) ~[agoo-dump-biz-2.0.0-SNAPSHOT.jar:na]
        at java.util.concurrent.Executors$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:1145) [na:1.7.0_51]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(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=0x000000000291e000 nid=0xc40 waiting on condition [0x0000000002dae000]
java.lang.Thread.State: WAITING (parking)
   at sun.misc.Unsafe.park(Native Method)
   - parking to wait for  <0x00000007d85697f8> (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-Le
ngth 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:12
8)
    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.

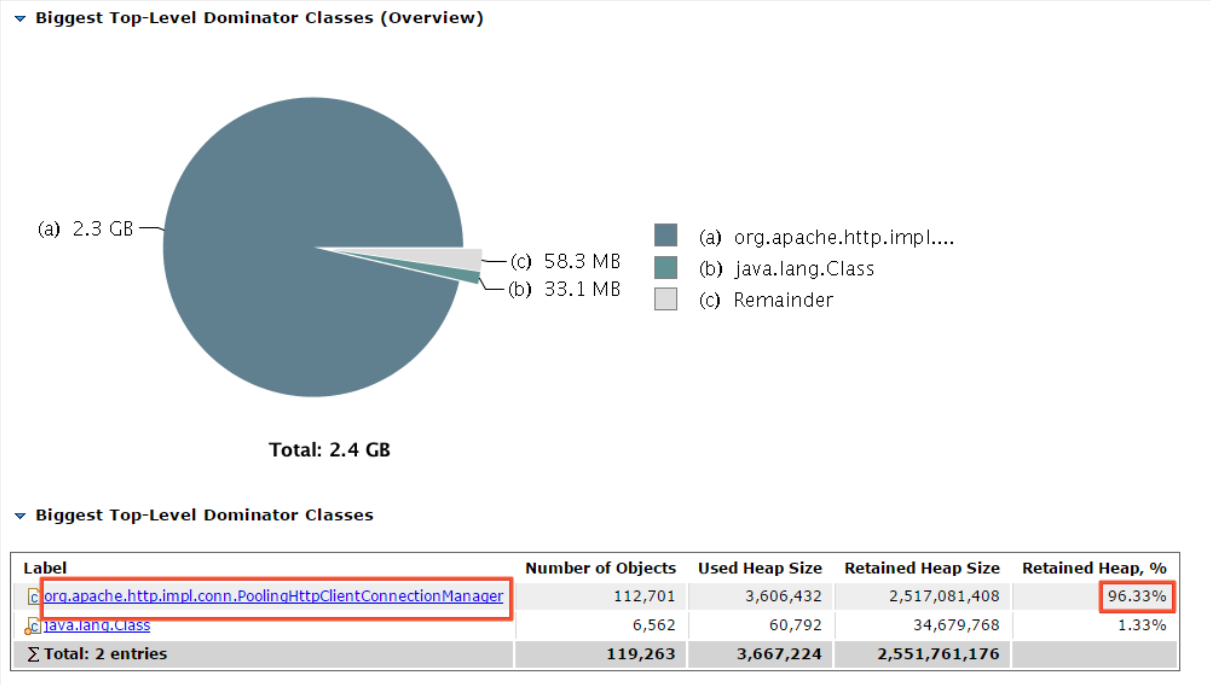
Solution: If you need to read part of the data during unspecified time periods, you can use range download to avoid connection closure during data reading.

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) {
}
```

2.2. Python-SDK

2.2.1. 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.
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).

Sample file	Description
bucket.py	Shows you how to manage buckets such as bucket creation, deletion, and listing.

2.2.2. 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
  i. Enter `python` in the command line and press Enter to go to the Python environment.
  ii. 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:

  i. Enter `python` in the command line and press the Enter key to go the Python environment.
  ii. 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:

a. Run the following command to uninstall the crcmod library:

```
pip uninstall crcmod
```

b. Install python-devel.

c. 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
```

2.2.3. Quick start

This topic describes how to use OSS SDK for Python to perform routine operations such as create buckets, upload objects, and download objects.

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:

```
# -*- 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 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 an object to OSS:

```
# -*- 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>')

# <yourObjectName> indicates the complete path of the object you want to upload to OSS, and must in
clude the extension of the object name. Example: abc/efg/123.jpg.
# <yourLocalFile> consists of a local file path and an object name with extension. Example: /users/loc
al/myfile.txt.
bucket.put_object_from_file('<yourObjectName>', '<yourLocalFile>')
```

Download objects

The following code provides an example on how to download a specified object to a local directory:

```
# -*- 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>')
# <yourObjectName> indicates the complete path of the object you want to download from OSS, and m
ust include the extension of the object name. Example: abc/efg/123.jpg.
# <yourLocalFile> consists of a local file path and an object name with extension. Example: /users/loc
al/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>')
# 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.ObjectIteratorr 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>')
# 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>')

# <yourObjectName> 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.
bucket.delete_object('<yourObjectName>')
```

2.2.4. Initialization

This topic describes how to initialize OSS SDK for Python.

Most operations of OSS SDK for Python are performed based on oss2.Service and oss2.Bucket.

- oss2.Service is used to list buckets.
- oss2.Bucket is used to upload, download, and delete objects as well as configure buckets.

You need to specify the endpoint when you initialize the two classes. oss2.Service does not support access from custom domain names. For more information about endpoints, see Obtain an endpoint.

Initialize oss2.Service

For more information, see List buckets.

Initialize oss2.Bucket

- Use OSS domain names to initialize oss2.Bucket

  The following code provides an example on how to initialize oss2.Bucket by uisng an OSS domain name:

  ```
  # -*- 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.
  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>')
# The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
endpoint = 'https://oss-cn-hangzhou.aliyuncs.com'

# Set the timeout period to 30 seconds.
bucket = oss2.Bucket(auth, endpoint, '<yourBucketName>', connect_timeout=30)
```

- 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:

```
# -*- 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.
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 during upload and download may be affected.

2.2.5. Buckets

2.2.5.1. 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:

```
# -*- 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. Th
e 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>')

# Create the bucket. By default, the ACL of the created bucket is private.
bucket.create_bucket()
```

2.2.5.2. 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.

The following code provides an example on how to list all 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.
service = oss2.Service(auth, 'http://oss-cn-hangzhou.aliyuncs.com')

print([b.name for b in oss2.BucketIterator(service)])
```

2.2.5.3. 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:

```
# -*- 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>')

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')
```

2.2.5.4. 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.	oss2.BUCKET_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.	oss2.BUCKET_ACL_PUBLIC_READ

ACL	Description	Value
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	oss2.BUCKET_ACL_PUBLIC_READ_WRITE

The following code provides an example on how to configure the ACL for a bucket:

```
# -*- 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 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 obtain the ACL of a bucket:

```
# -*- 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>')

print(bucket.get_bucket_acl().acl)
```

2.2.5.5. 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 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.

2.2.5.6. Manage lifecycle rules

OSS allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to IA or Archive. This way, storage costs are minimized. 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 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:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import LifecycleExpiration, LifecycleRule, BucketLifecycle,AbortMultipartUpload
import datetime

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>')

# 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 last modified before this date expired.
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 data. Parts that are last modified before this date expired.
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)
```

View lifecycle rules

The following code provides an example on how to view the lifecycle rules configured for a bucket:

```
# -*- 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>')

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))
```

Delete lifecycle rules

The following code provides an example on how to delete the lifecycle rules configured for a bucket:

```
# -*- 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>')

bucket.delete_bucket_lifecycle()

# An exception occurs when you view lifecycle rules again.
try:
    lifecycle = bucket.get_bucket_lifecycle()
except oss2.exceptions.NoSuchLifecycle:
    print('lifecycle is not configured')
```

2.2.5.7. Static website hosting

This topic describes how to set the static website hosting mode for your bucket. 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

Run the following code to configure static website hosting:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketWebsite

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>')

# 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'))
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
# -*- 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:
    # If static website hosting is disabled, the system returns NoSuchWebsite when get_bucket_website is used.
    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 static website hosting configurations

Run the following code to delete static website hosting configurations:

```
# -*- 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>')

bucket.delete_bucket_website()
```

2.2.5.8. 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

Run the following code to configure CORS rules for the specified bucket:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketCors, CorsRule

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>')

rule = CorsRule(allowed_origins=['*'],
                allowed_methods=['GET', 'HEAD'],
                allowed_headers=['*'],
                max_age_seconds=1000)

# The existing rules are replaced.
bucket.put_bucket_cors(BucketCors([rule]))
```

Query CORS rules

Run the following code to obtain CORS rules:

```
# -*- 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:
    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

Run the following code to delete all CORS rules for the specified bucket:

```
# -*- 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>')

bucket.delete_bucket_cors()
```

2.2.5.9. 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:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketReferer

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 Allow Empty Referer to True. True indicates that an empty Referer field is allowed and False indi
cates that an empty Referer field is not allowed. Configure the Referer whitelist.
bucket.put_bucket_referer(BucketReferer(True, ['http://aliyun.com', 'http://*.aliyuncs.com']))
```

Query hotlink protection configurations

Run the following code to obtain a referer whiltelist:

```
# -*- 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>')

config = bucket.get_bucket_referer()
print('allow empty referer={0}, referers={1}'.format(config.allow_empty_referer, config.referers))
```

Delete hotlink protection configurations

The following code provides an example on how to delete hotlink protection configurations of a bucket:

```
# -*- 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>')

# You cannot delete a Referer whitelist directly. To delete a Referer whitelist, you must create a rule t
hat allows an empty Referer field to replace the existing rule.
bucket.put_bucket_referer(BucketReferer(True, []))
```

2.2.5.10. 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

Run the following code to enable bucket access logging:

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import BucketLogging

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>')

# Enable logging. Save the log objects to the current bucket. Set the folder that stores the log files 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)
```

View logging configurations

Run the following code to view the access logging configurations for a bucket:

```
# -*- 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>')

logging = bucket.get_bucket_logging()
print('TargetBucket={0}, TargetPrefix={1}'.format(logging.target_bucket, logging.target_prefix))
```

Disable logging

Run the following code to disable access logging for a bucket:

```
# -*- 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>')

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)
```

2.2.6. Upload objects

2.2.6.1. 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 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: 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. After the object is uploaded, you can perform upload callback. For more information, see Upload callback.

2.2.6.2. Simple upload

You can use simple upload to upload objects of the following types: strings, bytes, Unicode, 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	Upload directly.

Type	Upload method
Bytes	Upload directly.
Unicode	Convert to bytes encoded in UTF-8 to upload.
Local files	File object. The file object must be opened in binary mode such as the rb mode.
Network streams	Iterable object. The iterable object is uploaded 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 a 200 OK is returned.

Upload strings

The following code provides an example on how to upload a string:

```
# -*- 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>')

# Obtain the return result.
result = bucket.put_object('<yourObjectName>', 'content of object')
# Obtain the returned HTTP status code.
print('http status: {0}'.format(result.status))
# Obtain the unique request ID. We recommend that you add this parameter to the program logs.
print('request_id: {0}'.format(result.request_id))
# Obtain the ETag value returned by the put_object method.
print('ETag: {0}'.format(result.etag))
# Obtain the HTTP response headers.
print('date: {0}'.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>')
# 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>')

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>')
# 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>')

bucket.put_object('<yourObjectName>', u'content of object')
```

Upload local files

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>')
# 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>')

# The object must be opened in binary mode because the number of bytes in the object must be availa
ble before the upload.
with open('<yourLocalFile>', 'rb') as fileobj:
    # Use the seek method to read data from the specified position of the 1000th byte. The data is uplo
aded from the specified 1000th byte until the entire object is uploaded.
    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 network streams

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>')
# 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>')

# The requests.get method returns an iterable object. In this case, OSS SDK for Python uploads the ob
ject by using chunked encoding.
input = requests.get('http://www.aliyun.com')
bucket.put_object('<yourObjectName>', input)
```

2.2.6.3. Append upload

OSS allows you to use append upload to append content directly to the end of an object. You can apply this method only to append objects.

Run the following code for append 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>')

# Set the position for the first upload (the Position parameter) 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 st
arts this time from the next_position field in the response returned by the last operation or by using t
he 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 two 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.

2.2.6.4. 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

Parameter	Description	Required	Default value
filename	The local file path from which to upload the object.	Yes	None
store	The directory where the checkpoint information is stored.	No	The *.py-oss-upload* 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
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 valu
e. The multipart_threshold parameter is optional, and the default value of multipart_threshold is 10 M
B. If you do not specify a directory by using the store parameter, the .py-oss-upload directory is create
d 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 s
pecified 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 uploa
d 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.

2.2.6.5. 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:

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

   - 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.
   - OSS includes the MD5 hash of part data in the ETag header and returns the MD5 hash to the user.
   - 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:

```
# -*- 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 posi
tion.
        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 para
meter 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)
```

2.2.6.6. 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)
```

2.2.6.7. Upload callback

This topic describes how to use upload callback.

For the complete code used to implement upload callback, visit GitHub.

You can use upload callback when you use put_object, put_object_from_file, and complete_multipart_upload.

The following code provides an example on how to use upload callback:

```
# -*- coding: utf-8 -*-
import json
import base64
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>')

# Configure the upload callback parameter.
callback_dict = {}
# Configure the IP address of the server to which you want to send the callback request. Example: htt
p://oss-demo.aliyuncs.com:23450 or http://127.0.0.1:9090.
callback_dict['callbackUrl'] = 'http://oss-demo.aliyuncs.com:23450'
# Optional. Configure the value of the host field carried in the callback request header.
# callback_dict['callbackHost'] = 'yourCallbackHost'
# Configure the value of the body field included in the callback request.
callback_dict['callbackBody'] = 'filename=${object}&size=${size}&mimeType=${mimeType}'
# Configure Content-Type for the callback request.
callback_dict['callbackBodyType'] = 'application/x-www-form-urlencoded'
# The callback parameter is in the JSON format and encoded in Base64.
callback_param = json.dumps(callback_dict).strip()
base64_callback_body = base64.b64encode(callback_param)
# Configure the header with the encoded parameter and send the header to OSS.
headers = {'x-oss-callback': base64_callback_body}

# Upload the object and perform upload callback.
result = bucket.put_object('<yourObjectName>', 'a'*1024*1024, headers)
```

2.2.7. Download objects

2.2.7.1. 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.

2.2.7.2. Streaming download

If you need to download a large object or it takes a long time to download an object at a time, you can use streaming download to download the object incrementally until the entire object is downloaded.

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>')
# 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>')

# The return value of bucket.get_object is a file-like and iterable object.
object_stream = bucket.get_object('<yourObjectName>')
print(object_stream.read())

# You must call read() to read the object from the stream returned by get_object before you can calcul
ate 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 t
o 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 a iterable object. You can copy object_stream to another object in streaming mode.
object_stream = bucket.get_object('<yourObjectName>')
bucket.put_object('<yourBackupObjectName>', object_stream)
```

2.2.7.3. 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 re
places 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/l
ocal/myfile.txt.
bucket.get_object_to_file('<yourObjectName>', '<yourLocalFile>')
```

2.2.7.4. 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:

```
# -*- 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>')

# If an object is 1,000 bytes in size, the valid range 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 sp
ecified value is larger than the object size, the entire 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.

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 specify Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns the HTTP status code 206 and the data that is within the range from byte 500 to byte 999.
- If you specify Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns the HTTP status code 416 and the error code InvalidRange.

The following code provides an example on how to perform compatible 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>')

# 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 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.
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 contnet_length:', object_stream.content_length)

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.
    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)
```

2.2.7.5. 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 *.py-oss-upload* 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. T
hen 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.

2.2.7.6. 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)
```

2.2.8. Manage objects

2.2.8.1. 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

2.2.8.2. Determine whether an object exists

This topic describes how to determine whether an object exists.

Run the following code to determine whether a specified object exists:

```
# -*- 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>')

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')
```

2.2.8.3. Manage object ACLs

This topic describes how to manage the access control list (ACL) of an object.

The following table describes the ACLs that 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 to which the object belongs.	oss2.OBJECT_ACL_DEFAULT
Private	Only the bucket owner and authorized users have the read and write permissions on the object.	oss2.OBJECT_ACL_PRIVATE
Public Read	Only the bucket owner and authorized users have read and write permissions on the object. Other users have only read permissions on the object. Exercise caution when you use this ACL.	oss2.OBJECT_ACL_PUBLIC_READ
Public	All users have read and write permissions on the object. Exercise caution when you use this ACL.	oss2.OBJECT_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.

Configure ACL for an object

You can run the following code to configure an ACL for 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>')

# Configure ACL for the object.
bucket.put_object_acl('<yourObjectName>', oss2.OBJECT_ACL_PUBLIC_READ)
```

Query the ACL of an object

You can run the following code to obtain the ACL for 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>')

print(bucket.get_object_acl('<yourObjectName>').acl)
```

2.2.8.4. Manage object metadata

Object metadata includes HTTP headers and user metadata.

Configure HTTP headers

Run the following code to configure HTTP headers:

```
# -*- 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>')

bucket.put_object('<yourObjectName>', '{"age": 1}', headers={'Content-Type': 'application/json; charset
=utf-8'})
```

For more information about HTTP headers, see RFC2616.

Configure user metadata

You can define user-defined Object Meta for object descriptions.

Run the following code to configure user-defined Object Meta:

```
# -*- 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>')

bucket.put_object('<yourObjectName>', 'a novel', headers={'x-oss-meta-author': 'O. Henry', 'Content-Ty
pe': 'application/json; charset=utf-8'})
```

Modify object metadata

Run the following code to modify Object Meta:

```
# -*- 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>')

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 operations provided by OSS SDK for Python to query object metadata.

Configuration method	Description	Benefits
get_object_meta	Queries parts of the object metadata including the ETag, size, and last modified time of the object.	More lightweight and faster
head_object	Queries all object metadata.	None

Run the following code to obtain Object Meta:

```
# -*- 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>')

// Query parts 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'])
```

2.2.8.5. 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>')
# 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>')

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>')
# 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 objects in the 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>')
# 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 objects in the fun folder, including objects from the subfolders.
for obj in oss2.ObjectIterator(bucket, prefix='fun/'):
    print(obj.key)
```

Example output:

```
fun/movie/001.avi
fun/movie/007.avi
fun/test.jpg
```

List objects and subfolders in a specified folder

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.

- If prefix is set to the name of a folder, objects and subfolders whose names contain the prefix are listed.
- If delimiter is set to a forward slash (/), only the objects and subfolders in the folder are listed. The objects and folders in subfolders are not listed.

Run the following code to list all objects in a specified folder:

```
# -*- 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>')

# List objects and subfolders in the fun folder. Objects in the subfolders are not listed.
for obj in oss2.ObjectIterator(bucket, prefix = 'fun/', delimiter = '/'):
    # Use is_prefix to determine whether obj is a folder.
    if obj.is_prefix():  # folder
        print('directory: ' + obj.key)
    else:               # object
        print('file: ' + obj.key)
```

Example output:

```
directory: fun/movie/
file: fun/test.jpg
```

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:

```
# -*- 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>')
# 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>')

for obj in oss2.ObjectIterator(bucket, delimiter='/'):
    if obj.is_prefix():  # folder
        length = CalculateFolderLength(bucket, obj.key)
        print('directory: ' + obj.key + '  length:' + str(length) + "Byte.")
    else: # object
        print('file:' + obj.key + '  length:' + str(obj.size) + "Byte.")
```

2.2.8.6. 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 = '<yourtBucketName>'
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: *None*, *Ignore*, and *Use*. ■ *None*: This object has no header information configured. ■ *Ignore*: This object has header information configured, which is not used when the SQL statement is executed. ■ *Use*: 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.

Parameter	Description
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 firstname, lastname, and age. The SQL statement is `select firstname, age from ossobject`. ■ If KeepAllColumns is set to *true*, the output result is firstname,,age, in which a comma (,) is added to indicate the position of the excluded lastname column. ■ If KeepAllColumns is set to *false*, the output result is firstname,age. Note By using this parameter, you can directly use the code that is used to process GetObject to process SelectObject without modifications.
OutputRawData	■ If this parameter is set to *True*, SelectObject 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 *False*, the output data is encapsulated in frames. Default value: False.
EnablePayloadCrc	Specifies whether a cyclic redundancy check (CRC) value is calculated for each frame. Default value: False.
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 True for a CSV object, the current record is skipped when a column of this record has no data. If this parameter is set to True for a JSON object, the current record is skipped when a key of this record does not exist. If this parameter is set to False, a column without data is left empty in the output result. Example: A row includes the columns firstname, lastname, and age. The SQL statement is `select _1, _4 from ossobject`. ■ If this parameter is set to True, this row is skipped. ■ If this parameter is set to False, firstname,\n 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.

Parameter	Description
ParseJsonNumberAsString	■ If this parameter is set to True, all numbers in the JSON object are parsed as strings. ■ If this parameter is set to False, all numbers in the JSON object are parsed as integers or floating-point numbers. Default value: False. 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 True and use the CAST 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
                          ):
```

  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):
```

    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
Json_Type	■ By default, the object is a CSV object if this parameter is not specified. ■ If this parameter is specified, the parameter value must be LINES, which indicates that the object is a JSON Lines object. Note This operation does not apply to JSON Document objects.
RecordDelimiter	The row delimiter in the CSV object.
FieldDelimiter	The column delimiter in the CSV object.
QuoteCharacter	The quote character in the CSV object. Row and column delimiters enclosed in quote characters are processed as normal characters.
CompressionType	The format in which the object is compressed. No compression formats are supported. The default value is None.
OverwriteIfExists	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.

2.2.8.7. Delete objects

This topic describes how to delete objects.

Warning Deleted objects cannot be restored. Exercise caution when you delete objects.

Delete a single object

The following code provides an example on how to delete 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>')

# Delete the object. <yourObjectName> indicates the full path of the object that you want to delete fr
om OSS. The path must include the extension of the object name. Example: abc/efg/123.jpg.
# To delete a folder, set <yourObjectName> to the corresponding folder name. If the folder is not emp
ty, you must delete all objects in the folder before you can delete the folder.
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. A maximum of 1,000 objects can be deleted at a time.
result = bucket.batch_delete_objects(['<yourObjectName-a>', '<yourObjectName-b>', '<yourObjectNam
e-c>'])
# Display the names of 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>')
# 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>')
prefix = "<yourKeyPrefix>"

# Delete the objects whose names contain the specified prefix.
for obj in oss2.ObjectIterator(bucket, prefix=prefix):
    bucket.delete_object(obj.key)
```

2.2.8.8. Copy an object

You can copy an object from a source bucket to a destination bucket within the same region.

Object copy has the following limits:

- You must have read permissions on the source object and write permissions on the destination bucket.
- You cannot copy an object across regions.

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>')
# 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', '<yourDestinationBucketName>')

bucket.copy_object('<yourSourceBucketName>', '<yourSourceObjectName>', '<yourDestinationObjectName>')
```

Copy a large object

To copy an object larger than 1 GB, you must split the object into smaller parts and copy them sequentially by using UploadPartCopy. To implement multipart copy, perform the following operations:

1. Use bucket.init_multipart_upload to initiate a multipart copy task.
2. Use bucket.upload_part_copy to copy parts. All parts except for the last part must be greater than 100 KB.
3. Use bucket.complete_multipart_upload to complete the multipart copy task.

```
# -*- coding: utf-8 -*-
import oss2
from oss2.models import PartInfo
from oss2 import determine_part_size

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>')

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)
```

2.2.9. 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 it to a visitor to grant temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of access from visitors.

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))
```

2.2.10. 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 `u'中文'` 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 `b'中文'` 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 yourLocalFile parameter value used in bucket.get_object_to_file.
Input data stream	bytes	Example: data in bucket.put_object.

  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 bucket.list_object.

Output data	Type	Description
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
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.

2.2.11. IMG

OSS Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service that processes 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 OSS upload API operation. You can use OSS IMG to build your own image processing service.

Basic features

OSS provides the following IMG 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. Choose **Data Processing** > **Image Processing (IMG)**. Click Create Rule to customize an image processing style.
- Call multiple IMG features by using cascade processing.

Usage of IMG

You can use the standard `HTTP GET` request 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

  Access IMG anonymously by using a third-level domain in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=image/action,parame_value
```

  - bucket: the name of the bucket.
  - endpoint: the domain name used to access the region.
  - object: the image object uploaded to OSS.
  - image: the identifier reserved by IMG.
  - action: the operation performed on the image, such as resizing, cropping, or rotating.
  - parame: 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 an image is processed by using a custom style, you can access IMG anonymously by using 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 the custom style is added in the Apsara Stack Cloud Management (ASCM) 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,parame_value/action,parame_value/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

IMG also supports access over 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:

```
# -*- coding: utf-8 -*-
import oss2
endpoint = '<Your 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 SDK

  You can directly use `SDK` to access and process image objects configured with any ACLs.

  Note

  - For the complete IMG code, visit GitHub.
  - You can use SDK to process image objects by configuring Create Style, HTTPS, and Cascading Processing.

  - Basic operations

    Basic IMG operations include query image information, convert image formats, resize, crop, rotate, apply effects, and add watermarks.

```
# -*- coding: utf-8 -*-
import os
import oss2
endpoint = '<Your 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 the 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.
style = 'image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ'
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 = '<Your 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 the sample image.
bucket.put_object_from_file(key, 'example.jpg')
# Customize a 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.

- Cascade processing

```
# -*- coding: utf-8 -*-
import os
import oss2
endpoint = '<Your 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 the 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.

2.2.12. 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.

Variable	Type	Description
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. For more information, see OSS error response format.

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.
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.

Error	HTTP status code	OSS error code	Description
ObjectNotAppendable	409	ObjectNotAppendable	The error message returned because the object is not appendable.

2.3. PHP-SDK

2.3.1. 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.

Sample file	Description
BucketCors.php	Shows you how to configure, query, and delete the CORS configurations for a bucket.

2.3.2. 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
  i. 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"
  }
  ```

  ii. 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
  i. Select the corresponding version and download the packaged PHAR file from GitHub.
  ii. Import the PHAR file in your code:

```
require_once '/path/to/oss-sdk-php.phar';
```

- Source code
  i. Select the required version and download the ZIP package from GitHub.
  ii. The decompressed root directory contains the `autoload.php` file. Import the file to the code:

```
require_once '/path/to/oss-sdk/autoload.php';
```

2.3.3. 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:

```
<? 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: 51
84000.
    $ossClient->setTimeout(3600);
    // Set the timeout period for connection establishment. Unit: seconds. Default value: 10.
    $ossClient->setConnectTimeout(10);
} catch (OssException $e) {
    print $e->getMessage();
}
```

2.3.4. Quick start

This topic describes how to use OSS SDK for PHP 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 for objects stored in OSS. 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>";
// 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);
    $ossClient->createBucket($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 objects

The following code provides an example on how to upload an object 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>";
// 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>";
// <yourObjectName> indicates the complete path of the object that you want to upload to OSS. The p
ath must include the extension of the object name. 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 objects

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>";
// 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>";
// <yourObjectName> 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.
$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 specified bucket. By default, up to 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>";
// 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);

    $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 objects

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>";
// 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>";
// <yourObjectName> indicates the complete 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.
$object = "<yourObjectName>";

try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
    $ossClient->deleteObject($bucket, $object);
} catch (OssException $e) {
    print $e->getMessage();
}
```

2.3.5. Buckets

2.3.5.1. 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:

```
<? 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);
    // 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 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 used to create a bucket, visit GitHub.

2.3.5.2. 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.

The following code provides an example on how to list all buckets:

```
<? 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);

    $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");
}
```

2.3.5.3. 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:

```
<? 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);

    $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");
}
```

2.3.5.4. 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.	OssClient::OSS_ACL_TYPE_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.	OssClient::OSS_ACL_TYPE_PUBLIC_READ
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this ACL.	OssClient::OSS_ACL_TYPE_PUBLIC_READ_WRITE

The following code provides an example on how to configure the ACL 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>";
// 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>";
// 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 obtain 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>";
// 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);

    $res = $ossClient->getBucketAcl($bucket);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
print('acl: ' . $res);
```

2.3.5.5. 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:

```
<? 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");
```

2.3.5.6. Manage lifecycle rules

OSS allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to IA or Archive. This way, storage costs are minimized. 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 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:

```
<? 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>";
// 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>";

// Set 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_TI
MING_DAYS, 3);
$lifecycleRule = new LifecycleRule($ruleId0, $matchPrefix0, "Enabled", $actions);
$lifecycleConfig->addRule($lifecycleRule);
$actions = array();
// Specify that the objects that are last modified before the specified date expire.
$actions[] = new LifecycleAction(OssClient::OSS_LIFECYCLE_EXPIRATION, OssClient::OSS_LIFECYCLE_TI
MING_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");
```

View lifecycle rules

The following code provides an example on how to view the lifecycle rules configured 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>";
// 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>";

$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 the lifecycle rules configured 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>";
// 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>";

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");
```

2.3.5.7. 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:

```
<? 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>";
// 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>";

$refererConfig = new RefererConfig();
// Allow empty Referers.
$refererConfig->setAllowEmptyReferer(true);
// Add a Referer whitelist. You can use an asterisk (*) or a question mark (?) as a wildcard to set the R
eferer parameter.
$refererConfig->addReferer("www.aliiyun.com");
$refererConfig->addReferer("www.aliiyuncs.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 hotlink protection information

Run the following code to obtain a referer whiltelist:

```
<? 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>";
// 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>";

$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");
```

Delete hotlink protection configurations

Run the following code to clear a referer whitelist:

```
<? 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>";
// 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>";

$refererConfig = new RefererConfig();
try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);

    // You cannot delete hotlink protection configurations directly. To delete hotlink protection configur
ations, you must create a rule that allows an empty Referer field to replace the existing rule.
    $ossClient->putBucketReferer($bucket, $refererConfig);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

2.3.5.8. 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

Run the following code to enable bucket access logging:

```
<? 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>";

$option = array();
// Specify the bucket used to store log objects.
$targetBucket = "<yourBucketName>";
$targetPrefix = "access.log";

try {
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);

    // Enable logging.
    $ossClient->putBucketLogging($bucket, $targetBucket, $targetPrefix, $option);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

View logging configurations

Run the following code to view the access logging configurations 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>";
// 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>";

$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

Run the following code to disable access 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>";
// 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>";

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");
```

2.3.5.9. Configure 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 name to access this static website and be redirected to a specified index page or error page.

For the complete static website hosting code, visit GitHub.

Configure static website hosting

Run the following code 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>";
// 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>";

// 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");
```

View static website hosting configurations

Run the following code to view 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>";
// 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>";

$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

Run the following code 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>";
// 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>";

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");
```

2.3.5.10. 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.

For the complete CORS rule code, visit GitHub.

Configure CORS rules

Run the following code to configure CORS rules for the 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;
use OSS\Model\CorsConfig;
use OSS\Model\CorsRule;

$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>";

$corsConfig = new CorsConfig();
$rule = new CorsRule();
// AllowedHeaders and ExposeHeaders do not support wildcards.
$rule->addAllowedHeader("x-oss-header");
// AllowedOlowedMethods allows only one wildcard asterisk (*). Asterisks (*) indicate that all domain
names or operations are allowed.
$rule->addAllowedOrigin("http://www.b.com");
$rule->addAllowedMethod("POST");
$rule->setMaxAgeSeconds(10);
// A maximum of 10 rules can be configured for each bucket.
$corsConfig->addRule($rule);

try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);

    // The existing rules are replaced.
    $ossClient->putBucketCors($bucket, $corsConfig);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

Query CORS rules

Run the following code to obtain CORS rules:

```
<? 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>";

$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

Run the following code to delete all CORS rules for 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>";

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");
```

2.3.6. Upload objects

2.3.6.1. 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.

2.3.6.2. Simple upload

In simple upload, you can upload strings or local files. This topic describes how to upload strings and local files.

String upload

The following code provides an example on how to upload a string:

```
<? 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>";
// 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 to specify the information of the uploaded object. Fo
r example, you can set the object ACL to private and configure the user metadata of the object
```

```
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");
```

Local file upload

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>";
// 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>";
// <yourLocalFile> consists of a local file path and a file name with an extension. Example: /users/loca
l/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");
```

2.3.6.3. 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 len
gth 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");
```

2.3.6.4. 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:

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
   > - 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.
   > - OSS includes the MD5 hash of part data in the ETag header and returns the MD5 hash to the user.
   > - 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.

For the complete multipart upload code, visit GitHub.

The following code provides a complete example that describes the process of 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;

$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>";
$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 rel
ated operations such as cancel or query 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;
    $upOptions = 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.
```

```
    // Perform the MD5 verification.
    if ($isCheckMd5) {
        $contentMd5 = OssUtil::getMd5SumForFile($uploadFile, $fromPos, $toPos);
        $upOptions[$ossClient::OSS_CONTENT_MD5] = $contentMd5;
    }
    try {
        // Upload the parts.
        $responseUploadPart[] = $ossClient->uploadPart($bucket, $object, $uploadId, $upOptions);
    } 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 ETags and the PartNumbers 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. OSS verifies the validit
y of each part after it receives the submitted $uploadParts. 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 from the preceding sample code.

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	The size of each part.
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 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:

```
<? 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>";
$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");
```

Multipart upload of local files

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>";
// 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>";
$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");
```

Multipart upload of local folders

The following code provides an example on how to upload a local folder including all files in the folder 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>";
// 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>";
$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>";
// 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>";
$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>";
// 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>";

$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	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.
key-marker	All multipart upload tasks with objects whose names start with the letter that comes after the key-marker value in the alphabetical order. 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 list this time. The default value is 1000. The maximum value is 1000.
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.
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 the key-marker parameter is configured, the query result includes: • All objects whose names start with a letter that comes after the key-marker parameter value in alphabetical order. • 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.

2.3.6.5. Upload callback

This topic describes how to use upload callback in simple upload and multipart upload.

Configure callback for simple upload

The following code provides an example on how to configure callback for 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;
```

```
$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);

// Configure callback for upload.
// callbackUrl is the IP address of the callback server. Examples: http://oss-demo.aliyuncs.com:23450 and http://127.0.0.1:9090.
// Optional. callbackHost is the value of the host field carried in the callback request.
$url =
    '{
        "callbackUrl":"<yourCallbackServerUrl>",
        "callbackHost":"<yourCallbackServerHost>",
        "callbackBody":"bucket=${bucket}&object=${object}&etag=${etag}&size=${size}&mimeType=${mimeType}&imageInfo.height=${imageInfo.height}&imageInfo.width=${imageInfo.width}&imageInfo.format=${imageInfo.format}&my_var1=${x:var1}&my_var2=${x:var2}",
        "callbackBodyType":"application/x-www-form-urlencoded"
    }';

// Configure custom parameters for the callback request. Each custom parameter consists of a key and a value. The key must start with x:.
$var =
    '{
        "x:var1":"value1",
        "x:var2":"value2"
    }';
$options = array(OssClient::OSS_CALLBACK => $url,
    OssClient::OSS_CALLBACK_VAR => $var
);
$result = $ossClient->putObject($bucket, $object, file_get_contents(__FILE__), $options);
print_r($result['body']);
print_r($result['info']['http_code']);
```

Configure callback for multipart upload

The following code provides an example on how to configure callback for 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;

$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>";
$uploadFile = "<yourLocalFile>";

/**
 * Step 1. Initiate a multipart upload task and obtain the uploadId.
 */

$ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
// An upload ID is returned. It is the unique identifier for a multipart upload task. You can initiate relat
ed operations such as cancel or query a multipart upload task based on the upload ID.
$uploadId = $ossClient->initiateMultipartUpload($bucket, $object);

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;
    $upOptions = array(
```

```
    $upOptions = 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);
        $upOptions[$ossClient::OSS_CONTENT_MD5] = $contentMd5;
    }
    try {
        // Upload the parts.
        $responseUploadPart[] = $ossClient->uploadPart($bucket, $object, $uploadId, $upOptions);
    } 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 ETags and the PartNumbers of each part.
$uploadParts = array();
foreach ($responseUploadPart as $i => $eTag) {
    $uploadParts[] = array(
        'PartNumber' => ($i + 1),
        'ETag' => $eTag,
    );
}

/**
 * Step 3. Complete the multipart upload task.
 */
// callbackUrl is the IP address of the callback server. Examples: http://oss-demo.aliyuncs.com:23450
and http://127.0.0.1:9090.
// Optional. callbackHost is the value of the host field carried in the callback request.
$json =
    '{
        "callbackUrl":"<yourCallbackServerUrl>",
        "callbackHost":"<yourCallbackServerHost>",
```

```
    "callbackBody":"{\"mimeType\":${mimeType},\"size\":${size},\"x:var1\":${x:var1},\"x:var2\":${x:va
r2}}",
    "callbackBodyType":"application/json"
  }';

// Configure custom parameters for the callback request. Each custom parameter consists of a key an
d a value. The key must start with x:.
$var =
  '{
    "x:var1":"value1",
    "x:var2":"value2"
  }';
$options = array(OssClient::OSS_CALLBACK => $json,
  OssClient::OSS_CALLBACK_VAR => $var);
// You must provide all valid $uploadParts when you perform this operation. OSS verifies the validity o
f each part after it receives the submitted $uploadParts. After all parts are verified, OSS combines the
se parts into a complete object.
$ossClient->completeMultipartUpload($bucket, $object, $uploadId, $uploadParts, $options);

printf(__FUNCTION__ . ": completeMultipartUpload OK\n");
```

2.3.7. Download objects

2.3.7.1. 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

2.3.7.2. 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 p
ath 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/loca
l/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 ar
e 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");
```

2.3.7.3. 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.

2.3.7.4. 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 spe
cified range is invalid, for example, the specified range includes a negative number, or the specified v
alue 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");
```

2.3.7.5. 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");
```

2.3.8. Manage objects

2.3.8.1. 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

2.3.8.2. Determine whether objects exist

This topic describes how to determine whether an object exists.

Use the following code to determine whether a specified 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>";
// 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);

    $exist = $ossClient->doesObjectExist($bucket, $object);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
var_dump($exist);
```

2.3.8.3. 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 an object.

ACL	Description	Value
Inherited from the bucket	The ACL of an object is the same as that of the bucket to which the object belongs.	default

ACL	Description	Value
Private	Only the owner or authorized users of the bucket can read and write the object.	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.	public-read
Public read/write	Any users, including anonymous users can read and write the object. Exercise caution when you use this 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.

Configure ACL for an object

You can run the following code to configure an ACL for a specified 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>";
// 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>";
// Set the ACL for the object to public read. By default, the ACL for the object inherit the ACL for the b
ucket.
$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 for an object

You can run the following code to obtain the ACL for a specified 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>";
// 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);

    $objectAcl = $ossClient->getObjectAcl($bucket, $object);
} catch (OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
var_dump($objectAcl);
```

2.3.8.4. 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>";
$copyOptions = 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, $copyOptions);
} 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");
}
```

2.3.8.5. 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 la
st object previously obtained through the listObjects 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

2.3.8.6. 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 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>";
// 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);

    $ossClient->deleteObject($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 at a time:

```
<? 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>";

$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 you use try{}catch{} and an exception is returned, a single object or multiple objects fail to be deleted. You can use $e->getMessage to obtain the error message and analyze causes.

2.3.8.7. Copy an object

This topic describes how to copy an object.

Object copy has the following limits:

- 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

You can call $ossClient->copyObject to copy an object smaller than or equal to 1 GB 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>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the 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");
```

For more information about how to copy a small object, see CopyObject.

Copy a large object

To copy an object larger than 1 GB, you must split the object into smaller parts and copy them sequentially by using UploadPartCopy. To implement multipart copy, perform the following operations:

1. Call $ossClient->initiateMultipartUpload to initiate a multipart copy task.
2. Call $ossClient->uploadPartCopy to perform multipart copy. All parts except for the last part, must be larger than 100 KB.
3. Call $ossClient->completeMultipartUpload to complete the multipart copy task.

The following code provides an example on how to perform multipart copy:

```
<? 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>";
// The endpoint of the China (Hangzhou) region is used in this example. Specify the 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_part
s);
} catch(OssException $e) {
    printf(__FUNCTION__ . ": FAILED\n");
    printf($e->getMessage() . "\n");
    return;
}
print(__FUNCTION__ . ": OK" . "\n");
```

2.3.9. 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 it to a visitor to grant temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of access from visitors.

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')) {
```

```
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 singed URL.
$request->set_method('GET');
$request->add_header('Content-Type', '');
$request->send_request();
```

```
$res = new ResponseCore($request->get_response_header(), $request->get_response_body(), $re
quest->get_response_code());
if ($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");
};
```

2.3.10. IMG

Image Processing (IMG) provided by OSS is a secure, cost-effective, and highly reliable image processing service that processes 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 provides the following IMG 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. Choose **Data Processing** > **Image Processing (IMG)**. Click Create Rule to customize an image processing style.
- Call multiple IMG features by using cascade processing.

Usage of IMG

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

You can access IMG anonymously by using a third-level domain in the following format:

```
http://bucket.<endpoint>/object?x-oss-process=image/action,parame_value
```

Parameter	Description
bucket	The name of the bucket.
endpoint	The domain name used to access the region.
object	The image object uploaded to OSS.
image	The identifier reserved by IMG.
action	Operations performed on images, such as scaling, cropping, and rotating.
param	The parameters corresponding to the operations performed on the images.

- Basic operations

  Example: Scale an image to a width of 100 pixels 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
  ```

- Custom styles

  Access IMG anonymously by using a third-level domain in the following format:

  ```
  http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleNam
  e>
  ```

  - style: the identifier reserved by the system for the custom style.
  - yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when the custom style is added in the Apsara Stack Cloud Management (ASCM) 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 over HTTPS

  IMG also supports access over HTTPS. Example:

  ```
  https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
  ```

- Authorized access

  Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

  Run the following code to generate a signed URL for 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. Set the validity period to 3,600 seconds and allow access from a browser.
$timeout = 3600;
$options = array(
    OssClient::OSS_PROCESS => "image/resize,m_lfit,h_100,w_100" );
$signedUrl = $ossClient->signUrl($bucket, $object, $timeout, "GET", $options);

print("rtmp url: \n" . $signedUrl);
```

- Access by using SDK

  You can use SDK to access and process an image object.

  SDK allows you to customize image styles, HTTPS access, and cascade operations.

  - Basic operations

    Run the following code to perform basic operations on an image:

```
<? 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 endpoi
nt.
$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");

// Scale 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);
```

```
$ossClient->getObject($bucket, $object, $options);

// Add watermarks.
$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 styles

  Run the following code to customize an image style:

```
<? 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 endpoi
nt.
$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");

// Custom styles.
$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

  Use the following code to perform cascade operations on an image:

```
<? 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 endpoi
nt.
$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");

// 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);
```

Image processing persistence

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 name of the target object.
$targetObject = "<yourTargetObjectName>";



try{
    $ossClient = new OssClient($accessKeyId, $accessKeySecret, $endpoint);
// Convert the image format and save the processed image. To crop, rotate, or add watermarks for the
image, modify the corresponding parameters in $options.
$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);
print(__FUNCTION__ . ": OK" . "\n");
```

2.3.11. 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

Error code	Error message	HTTP status code
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

Error code	Error message	HTTP status code
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

Error code	Error message	HTTP status code
InvalidEncryptionAlgorithmError	The error message returned because the specified encryption algorithm based on entropy encoding is invalid.	400

2.4. C-SDK

2.4.1. 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

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).

2.4.2. 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: cRUL, 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

Parameter	Description	Version
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-util1-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 cRUL, 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 cRUL, 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 cRUL, 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.

2.4.3. 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 cRUL-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;
}
```

2.4.4. Quick start

This topic describes how to use 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 decompress 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. Set OSS_ENDPOINT, ACCESS_KEY_ID, ACCESS_KEY_SECRET, and BUCKET_NAME in the sample code to 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 decompress 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

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:

```
#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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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);
    /* Create the bucket. */
    aos_status_t *resp_status;
    aos_table_t *resp_headers;
    oss_acl_e 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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

Upload objects

The following code provides an example on how to upload an object 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>";
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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_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. */
    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 upload request is successful. */
    if (aos_status_is_ok(resp_status)) {
        printf("put file succeeded\n");
    } else {
        printf("put 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;
}
```

Download objects

The following code provides an example on how to download a specified object to a local directory:

```
#include "oss_api.h"
#include "aos_http_io.h"
const char *endpoint = "<yourEndpoint>";
const char *access_key_id = "<yourAccessKeyId>";
```

```
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 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_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 download request is successful. */
    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, which releases 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 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 t
o create a memory pool is included in the APR library. */
    aos_pool_t *pool = NULL;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options = NULL;
    /* 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;
    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 list request is successful. */
    if (! aos_status_is_ok(resp_status))
    {
        printf("list object failed\n");
    }
    else{
        /* Display the objects. */
        printf("Object\tSize\tLastModified\n");
```

```
    printf("Object\tSize\tLastModified\n");
    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);
    }
    printf("Total %d\n", 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;
}
```

Delete objects

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 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 t
o create a memory pool is included in the APR library. */
    aos_pool_t *pool = NULL;
    /* Create a memory pool. The second parameter is NULL. This parameter indicates that the pool doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t *oss_client_options = NULL;
    /* 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_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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

2.4.5. Buckets

2.4.5.1. 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:

```
#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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, access_key_id, acces_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;
    oss_acl_e oss_acl = OSS_ACL_PRIVATE;
    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);
    /* 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, which releases 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 set the ACL of a bucket when you create the bucket:

```
    /* Create a bucket and set the ACL of the bucket to public read (the default ACL is private). */
    resp_status = oss_create_bucket(oss_client_options, &bucket, OSS_ACL_PUBLIC_READ, &resp_heade
rs);
    if (aos_status_is_ok(resp_status)) {
        printf("create bucket succeeded\n");
    } else {
        printf("create bucket failed\n");
    }
```

2.4.5.2. 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.

The following code provides an example on how to list all buckets:

```
#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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, access_key_id, acces_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;
}
```

2.4.5.3. 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.	OSS_ACL_PRIVATE

ACL	Description	Value
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 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 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 t
o 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 doe
s not inherit any other memory pool. */
```

```
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_heade
rs);
    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 obtain 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information, such as e
ndpoint, access_key_id, acces_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;
}
```

2.4.5.4. 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 in the bucket and parts generated by multipart upload.

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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, access_key_id, acces_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;
}
```

2.4.5.5. Manage lifecycle rules

OSS allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to IA or Archive. This way, storage costs are minimized. 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 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:

```
#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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information, such as e
ndpoint, access_key_id, acces_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);
    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_head
ers);
    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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

View lifecycle rules

The following code provides an example on how to view the lifecycle rules configured 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 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 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_head
ers);
    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, which releases 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 the lifecycle rules configured 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 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

2.4.5.6. 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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. */
```

```
/* 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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, no
de) {
                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, n
ode) {
                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, nod
e) {
                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, nod
e) {
                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();
```

```
    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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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;
}
```

2.4.5.7. Configure logging

You can enable logging to record access to a bucket in log objects, which are stored in a specified bucket.

The name of log objects is in the following format: `<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString`.

Enable logging

Run the following code to enable bucket access logging:

```
#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 *target_bucket_name = "<yourTargetBucketName>";
const char *target_logging_prefix = "<yourTargetPrefix>";
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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 parameters. */
    aos_string_t bucket;
    oss_logging_config_content_t *content;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    content = oss_create_logging_rule_content(pool);
    aos_str_set(&content->target_bucket, target_bucket_name);
    aos_str_set(&content->prefix, target_logging_prefix);
    /* Enable logging for the bucket. */
    resp_status = oss_put_bucket_logging(oss_client_options, &bucket, content, &resp_headers);
```

```
    if (aos_status_is_ok(resp_status)) {
        printf("put symlink succeeded\n");
    } else {
        printf("put symlink 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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

View logging configurations

Run the following code to view the access logging configurations 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 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 t
```

```
    /* Create a memory pool to manage memory. aos_pool_t is equivalent to apr_pool_t. The code used t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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;
    oss_logging_config_content_t *content;
    aos_table_t *resp_headers = NULL;
    aos_status_t *resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    content = oss_create_logging_rule_content(pool);
    /* View logging configurations of the bucket. */
    resp_status = oss_get_bucket_logging(oss_client_options, &bucket, content, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get bucket logging succeeded\n");
    } else {
        printf("get bucket logging 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);
    }
    printf("bucket:%s, prefix:%s\n", content->target_bucket.data, content->prefix.data);
    /* 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;
}
```

Disable logging

Run the following code to disable access logging 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 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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);
    /* Disable logging. */
```

```
    resp_status = oss_delete_bucket_logging(oss_client_options, &bucket, &resp_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("delete bucket logging succeeded\n");
    } else {
        printf("delete bucket logging 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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

2.4.5.8. 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 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 whiltelist:

```
#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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
```

```
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 configura
tions, 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;
}
```

2.4.5.9. 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);
```

```
    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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_header
s);
    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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_header
s);
    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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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;
```

```
}
```

2.4.6. Upload objects

2.4.6.1. Overview

Objects are the basic unit 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 streaming 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, and 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 progress bars. For more information about progress bar, seeUpload progress bars. After the object is uploaded, you can perform upload callback. For more information about upload callback, see Upload callback.

2.4.6.2. 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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) {
```

```
    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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 do
es not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
  /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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);

  /* 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;
}
```

2.4.6.3. 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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, &buff
er, 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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, hea
ders2, &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;
}
```

2.4.6.4. 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.

Current upload progress is recorded into a checkpoint file during the upload. If a part fails to be uploaded during the process, the object is uploaded from the part recorded in the checkpoint file when the upload restarts, that is, the resumable upload. After the upload is complete, the checkpoint is deleted.

You can use `oss_resumable_clt_params_t` to perform resumable upload. The following table describes the parameters you can configure for this method.

Parameter	Description
thread_num	The number of concurrent threads. Default value: 1.
enable_checkpoint	Specifies whether to enable resumable upload. By default, resumable upload is disabled.
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.

You can run the following code 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 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 *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, which releases memory resources allocated for the request. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

2.4.6.5. 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:

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**
   > - 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.
   > - OSS includes the MD5 hash of part data in the ETag header and returns the MD5 hash to the user.
   > - 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:

```
#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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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(p, 1);
    complete_headers = aos_table_make(p, 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_n
um++, upload_file, &resp_headers);
```

```
        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.len, 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. */
```

```
    /* 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 method 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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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");
```

```
    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.

2.4.6.6. 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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, p
ercentage, &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.

2.4.6.7. Upload callback

This topic describes how to use upload callback.

For the complete code used to implement upload callback, visit GitHub.

The following code provides an example on how to use upload callback:

```
#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[])
{
    aos_pool_t *p = NULL;
    aos_status_t *s = NULL;
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t *headers = NULL;
```

```
oss_request_options_t *options = NULL;
aos_table_t *resp_headers = NULL;
aos_list_t resp_body;
aos_list_t buffer;
aos_buf_t *content;
char *buf = NULL;
int64_t len = 0;
int64_t size = 0;
int64_t pos = 0;
char b64_buf[1024];
int b64_len;
/* Use the JSON format. */
/* Optional. Configure the value of the host field carried in the callback request header. */
char *callback =  "{"
    "\"callbackUrl\":\"http://oss-demo.aliyuncs.com:23450\","
     "\"callbackHost\":\"yourCallbackHost\","
    "\"callbackBody\":\"bucket=${bucket}&object=${object}&size=${size}&mimeType=${mimeType}\","
    "\"callbackBodyType\":\"application/x-www-form-urlencoded\""
    "}";
/* 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);
}
/* Initialize the parameters. */
aos_pool_create(&p, NULL);
options = oss_request_options_create(p);
init_options(options);
aos_str_set(&bucket, bucket_name);
aos_str_set(&object, object_name);
aos_list_init(&resp_body);
aos_list_init(&buffer);
content = aos_buf_pack(options->pool, object_content, strlen(object_content));
aos_list_add_tail(&content->node, &buffer);
/* Add callback to the header. */
b64_len = aos_base64_encode((unsigned char*)callback, strlen(callback), b64_buf);
b64_buf[b64_len] = '\0';
headers = aos_table_make(p, 1);
apr_table_set(headers, OSS_CALLBACK, b64_buf);
/* Configure upload callback. */
s = oss_do_put_object_from_buffer(options, &bucket, &object, &buffer,
```

```
        headers, NULL, NULL, &resp_headers, &resp_body);
    if (aos_status_is_ok(s)) {
        printf("put object from buffer succeeded\n");
    } else {
        printf("put object from buffer failed\n");
    }
    /* Obtain the buffer length. */
    len = aos_buf_list_len(&resp_body);
    buf = (char *)aos_pcalloc(p, (apr_size_t)(len + 1));
    buf[len] = '\0';
    /* Copy the content from the buffer to the memory. */
    aos_list_for_each_entry(aos_buf_t, content, &resp_body, node) {
        size = aos_buf_size(content);
        memcpy(buf + pos, content->pos, (size_t)size);
        pos += size;
    }
    /* 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;
}
```

2.4.7. Download objects

2.4.7.1. 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.

2.4.7.2. 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 fil
e, 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.

2.4.7.3. 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_pcalloc(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;
}
```

2.4.7.4. 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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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, &bu
ffer, &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_pcalloc(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;
}
```

2.4.7.5. 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.

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.

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
thread_num	The number of concurrent threads. Default value: 1.
enable_checkpoint	Specifies whether to enable resumable upload. By default, resumable upload is disabled.
partSize	The size of each part. Valid values: 1 byte to 5 GB.
checkpoint_path	The path of the checkpoint file. Default value: {upload_file_path}.cp.

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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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(&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, N
ULL, 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;
}
```

2.4.7.6. 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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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, p
ercentage, &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, per
centage, 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.

2.4.8. Manage objects

2.4.8.1. 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

2.4.8.2. 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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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_head
ers);
```

```
    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;
}
```

2.4.8.3. 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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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", obje
ct_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.

2.4.8.4. 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

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 t
```

```
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, 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);
```

```
            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 rele
ased. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

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 t
he 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, 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");
```

```
        printf("list object failed\n");
    }
    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);
    }
    printf("Total %d\n", size);
    /* Release the memory pool after the request is complete. In this case, the memory allocated to vari
ous 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 resourc
es. */
    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) ! = AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. The code used to create a memory pool is included in t
he 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, 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 vari
ous 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 resourc
es. */
    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) ! = AOSE_OK) {
        exit(1);
    }
    /* Create a memory pool to manage memory. The code used to create a memory pool is included in t
he 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, 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.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 vari
ous 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 resourc
es. */
    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 use
d 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 d
oes 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_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 re
leased. */
    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) ! = 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;
    /* 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_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 v
arious 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 reso
urces.
    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. */
```

```
    /* 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 d
oes 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 poo
l. */
    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_pr
efix_list, node) {
            line = apr_psprintf(pool, "%.*s", commonPrefix->prefix.len, commonPrefix->prefix.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 v
arious 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 reso
urces. */
    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;
```

```
    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_psprintf(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 v
arious 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) ! = 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 d
oes 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 poo
l. */
    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_pr
efix_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 v
arious 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 reso
urces. */
  aos_http_io_deinitialize();
  return 0;
}
```

2.4.8.5. Delete objects

This topic describes how to delete objects.

> **Warning** You cannot recover deleted objects. Exercise caution when you delete objects.

For the complete code used to delete objects, visit GitHub.

Delete a single 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 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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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;
    /* Assign char* data to the aos_string_t bucket. */
    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, which releases 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 at a time:

```
#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 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);
```

```
    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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 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 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 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, which releases 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 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. */
```

```
    /* 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 an
d 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 t
o 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter specifies global configuration information such as e
ndpoint, 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 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, which releases 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.

2.4.8.6. 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 an
d 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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, access_key_id, acces_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 rele
```

```
ased. */
    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 an
d 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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, access_key_id, acces_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;
```

```
    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, &upl
oad_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 rele
ased. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

2.4.9. Authorized access

This topic describes how to authorize access to OSS by using Security Token Service (STS) or a signed URL.

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 use STS to authorize temporary access 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 *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 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 an
d 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 cre
ate 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 doe
s not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information such as en
dpoint, access_key_id, acces_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;
    /* 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->node, &buffer);

    /* Start to upload the object. */
    resp_status = oss_put_object_from_buffer(oss_client_options, &bucket, &object, &buffer, headers, &
```

```
    resp_status = oss_put_object_from_buffer(oss_client_options, &bucket, &object, &buffer, headers, &
resp_headers);
    /* Determine whether the upload is successful. */
    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. The memory allocated to various resources used for the request is rele
ased. */
    aos_pool_destroy(pool);

    /* Release the allocated global resources. */
    aos_http_io_deinitialize();

    return 0;
}
```

Use a signed URL to authorize temporary access

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 limit the period of access from visitors.

- 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:

```
#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 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 a
nd 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 c
reate 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 d
oes 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, acces_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_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: %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_hea
ders);
    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. The memory allocated to various resources used for the request is re
leased. */
    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 download a specified object by using a signed URL:

```
#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 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 a
nd 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 c
reate 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 d
oes 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, acces_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 *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, &res
p_headers);
    if (aos_status_is_ok(resp_status)) {
        printf("get object succeeded\n");
    } else {
        printf("get object failed\n");
    }
    /* Release the memory pool. The memory allocated to various resources used for the request is re
leased. */
    aos_pool_destroy(pool);
    /* Release the allocated global resources. */
    aos_http_io_deinitialize();
    return 0;
}
```

2.4.10. IMG

Image Processing (IMG) provided by OSS is a secure, cost-effective, and highly reliable image processing service that processes 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 provides the following IMG 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. Choose **Data Processing > Image Processing (IMG)**. Click Create Rule to customize an image processing style.
- Call multiple IMG features by using cascade processing.

Usage of IMG

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- Anonymous access

  You can use a third-level domain in the following format to anonymously access processed images:

  ```
  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	Operations on images, such as scaling, cropping, and rotating.
param	Parameters that are corresponding to the operations on images.

  - Basic operations

    Example: Scale an image to a width of 100 pixels 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
    ```

- Custom styles

  Access IMG anonymously by using a third-level domain in the following format:

  ```
  http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
  ```

  - style: the identifier reserved by the system for the custom style.
  - yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when the custom style is added in the Apsara Stack Cloud Management (ASCM) 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 over HTTPS

  IMG also supports access over HTTPS. Example:

  ```
  https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
  ```

- Authorized access

  Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

  Use the following code to generate a signed URL for 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 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 use
d 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 d
oes 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 *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 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 SDK

  You can use SDK to access and process an image object.

  SDK allows you to customize image styles, HTTPS access, and cascade operations.

  - Basic operations

    Run the following code to perform basic operations on an image:

```
#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 memo
ry. */
    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 u
sed 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 *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, &fi
```

```
le, &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, &fi
le, &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, &fi
le, &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, &fi
le, &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 the watermark. */
    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, &fi
le, &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, &fi
le, &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, &fi
le, &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 styles

  Run the following code to customize an image style:

```
#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 memo
ry. */
    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 u
sed 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 *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 an image processing 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, &fi
le, &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

  Run the following code to perform cascade operations on an image:

```
#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 memo
ry. */
    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 u
sed 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 *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, &fi
le, &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;
}
```

2.4.11. 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

Error code	Description	Type
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

Error code	Error message	HTTP status code
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

Error code	Error message	HTTP status code
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
TooManyBuckets	The error message returned because the number of the buckets managed by a user exceeds the limit.	400

2.5. Go-SDK

2.5.1. 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.

2.5.2. 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)
}
```

2.5.3. 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: `http://8.8.8.8:3128`.	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 c
an be bound to a bucket.
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.UseCna
me(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.Security
Token("<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)
}
```

2.5.4. 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. A bucket is a container for objects stored in OSS. 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() {
    // The endpoint of the China (Hangzhou) region is used in this example. Specify the 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 a 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 an object to OSS:

```
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() {
    // The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // <yourObjectName> indicates the complete path of the object that you want to upload to OSS. The path must include the extension of the object name. Example: abc/efg/123.jpg.
    objectName := "<yourObjectName>"
    // <yourLocalFileName> consists of a local file path and an object 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)
    }
    // Specify the bucket name.
    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 a local file:

```
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() {
    // The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // <yourObjectName> indicates the complete path of the object that you want to download from OS
S. The path 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)
    }
    // Specify the bucket name.
    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)
    }
    // Specify the bucket name.
    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, 100 records are returned each 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() {
    // The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // <yourObjectName> indicates the complete path of the object that you want to delete from OSS. T
he 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)
    }
    // Specify the bucket name.
    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 in Manage objects.

2.5.5. Buckets

2.5.5.1. 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.

For the complete code 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 a bucket. By default, the storage class of the bucket is Standard, and the ACL of the bucke
t is private.
    err = client.CreateBucket("<yourBucketName>")
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

2.5.5.2. 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.

For the complete code used to list buckets, visit GitHub.

For more information about how to list buckets, see GetBucket (ListObjects).

List all buckets

Buckets are listed in alphabetical order. You can list all buckets or the buckets that meet specified conditions.

The following code provides an example on how to list all 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)
    }

    // List all 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 each time.
        for _, bucket := range lsRes.Buckets {
            fmt.Println("Bucket: ", bucket.Name)
        }

        if lsRes.IsTruncated {
            marker = lsRes.NextMarker
        } else {
            break
        }
    }
}
```

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:

```
// List the buckets whose names contain a 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 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 that follow a specified marker:

```
    // List buckets whose names come after the specified 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 a specified number (MaxKeys) of buckets:

```
    // Set the maximum number of buckets to list each time to 500. The default value is 100. The maximu
m value is 1000.
    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)
    }
```

2.5.5.3. 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.

For the complete code used to manage the ACL of a bucket, visit GitHub.

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.	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 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)
    }

    // 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 obtain 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)
}
```

2.5.5.4. 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)
    }
}
```

2.5.5.5. Manage lifecycle rules

OSS allows you to configure lifecycle rules to delete expired objects and parts or convert the storage class of expired objects to IA or Archive. This way, storage costs are minimized. 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 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:

```
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 co
ntain 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)
    }
}
```

View lifecycle rules

The following code provides an example on how to view the lifecycle rules configured 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 the lifecycle rules.
    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 the lifecycle rules configured 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>"

    // Delete the lifecycle rules.
    err = client.DeleteBucketLifecycle(bucketName)
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
```

2.5.5.6. 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 asteris
k (*) 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 whiltelist:

```
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 di
rectly. To delete hotlink protection configurations, you must create a rule that allows an empty Refere
r field to replace the existing rule.
    err = client.SetBucketReferer(bucketName, []string{}, true)
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

2.5.5.7. 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. tar
getPrefix specifies the prefix of the stored log objects. The log objects are stored in the targetPrefix f
older.
    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)
    }
}
```

2.5.5.8. 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)
    }
}
```

2.5.5.9. 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)
    }
}
```

2.5.6. Upload objects

2.5.6.1. 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.

2.5.6.2. Simple upload

This topic describes how to use simple upload.

For the complete code used to perform simple upload, visit GitHub.

Upload a string

The following code provides an example on how to upload a string:

```
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 to which to upload the object.
    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, o
bjectAcl)
    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 to which to upload the object.
    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("yourObjectValueByteArrary")
))
    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 to which to upload the object.
    bucket, err := client.Bucket("<yourBucketName>")
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Read the local 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 a local file

The following code provides an example on how to upload a local 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 to which to upload the object.
    bucket, err := client.Bucket("<yourBucketName>")
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Upload the local file.
    err = bucket.PutObjectFromFile("<yourObjectName>", "<yourLocalFile>")
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

2.5.6.3. Append upload

This topic describes how to perform append upload.

You cannot perform the copyObject operation on append objects. If the object to append does not exist, you can call `AppendObject` to create an append object. If the object to append exists, you can call `AppendObject` to append content to the end of the object.

Run the following code to upload a file with 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)
    }

    // 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)
    }

    var nextPos int64 = 0
    // If the object is appended for the first time, the append position is 0. The returned value is the pos
ition for the next append. The position to start the next append is the total length of appended bytes
and the append object.
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectAppendV
alue1"), nextPos)
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Start the second append.
```

```
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectAppendV
alue2"), 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 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.
    nextPos, err = bucket.AppendObject("<yourObjectName>", strings.NewReader("YourObjectValue"), 0
, oss.Meta("MyProp", "MyPropVal"))
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
```

2.5.6.4. 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 complete.

> **Note**
>
> - The SDK 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 object being uploaded is modified, you must upload the entire object again.

You can use `Bucket.UploadFile` to perform resumable upload. The following table describes the parameters of this operation.

Parameter	Description
objectKey	The name of the object to upload to OSS.
filePath	The local file path from which to upload the object.
partSize	The size of each part. Valid values: 100 KB to 5 GB.
options	The optional parameters, including: • Routines: the number of concurrent upload threads. The default value is 1, indicating that only one thread is used to upload the parts. • Checkpoint: specifies whether to enable resumable upload and the path of the checkpoint file. By default, resumable upload is disabled. For example, `oss.Checkpoint(true, "")` indicates that resumable upload is enabled and the checkpoint file is `file.cp` contained in the same directory as the source local file. `file` is a variable that indicates the local file name. You can also use `oss.Checkpoint(true, "your-cp-file.cp")` 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 by using 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)
    }

    // 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)
    }

    // The part size is 100 KB. Start resumable upload. Upload the parts by using three concurrent threa
ds.
    // <yourObjectName> is equivalent to objectKey and indicates the complete path of the object that
you want to upload to 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.UploadFile("<yourObjectName>", "LocalFile", 100*1024, oss.Routines(3), oss.Checkpoint(
true, ""))
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

2.5.6.5. 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:

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
   > - 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.
   > - OSS includes the MD5 hash of part data in the ETag header and returns the MD5 hash to the user.
   > - 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:

```
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>"
locaFilename := "<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(locaFilename, 3)
fd, err := os.Open(locaFilename)
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"
```

```
    "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>"
    locaFilename := "<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(locaFilename, 3)
    fd, err := os.Open(locaFilename)
    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 o
peration.
    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: ", upl
oad.LastModified)
    }
    // List all the uploaded parts based on the result that is obtained by using the InitiateMultipartUplo
adResult 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: ", upl
oad.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 CommonPrefixes.
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 UploadIDMarker 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 KeyMarker parameter.
	• This parameter is ignored if the KeyMarker parameter is not set.
	• If the KeyMarker 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 KeyMarker value and the upload IDs are greater than the UploadIDMarker 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(uplo
adIdMarker))
        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 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)
```

2.5.6.6. 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)
    }
}
```

2.5.7. Download objects

2.5.7.1. 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.

2.5.7.2. 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 c
onnections 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)
}
```

2.5.7.3. 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 an object:

```
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 s
pecified 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 co
nnections 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 specify Range: bytes to 500-2000, the value at the end of the range is invalid. Then, OSS returns the HTTP status code 206 and the data that is within the range from byte 500 to byte 999.
- If you specify Range: bytes to 1000-2000, the value at the start of the range is invalid. Then, OSS returns the HTTP status code 416 and the error code InvalidRange.

The following code provides an example on how to perform compatible download:

```
package main

import (
    "fmt"
    "io/ioutil"
    "os"
    "strings"
```

```
    "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 2
06 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. OSS returns th
e 416 HTTP status code and the error code InvalidRange.
    body, err := bucket.GetObject("<yourObjectName>", oss.Range(500, 2000), oss.RangeBehavior("stan
dard"))
```

```
    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 c
onnections 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)
    }

    if len(data) ! = 500 {
        fmt.Println("read data error, len:%d", len(data))
        os.Exit(-1)
    }
    fmt.Println("data:", string(data))
}
```

2.5.7.4. 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)
    }
}
```

2.5.7.5. 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, `oss.Checkpoint(true, "")` indicates that resumable download is enabled and the checkpoint file is the `file.cp` file contained in the same directory as the destination local file. `file` is a variable that indicates the local file name. You can also use `oss.Checkpoint(true, "your-cp-file.cp")` to specify the checkpoint file. • Conditions: conditions for multipart download. For more information about the conditions, seeManage 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 yo
u 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.Checkpoi
nt(true, ""))
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

2.5.7.6. 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)
    }
}
```

2.5.7.7. 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)
```

```
    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.")
}
```

2.5.8. Manage objects

2.5.8.1. 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

2.5.8.2. 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)
}
```

2.5.8.3. 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)
}
```

2.5.8.4. 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 `X-Oss-Meta-` .

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 pu
blic read, and MyProp to MyPropVal as user metadata.
    expires := time.Date(2049, time.January, 10, 23, 0, 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...)
```

```
    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)
}
```

2.5.8.5. 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 commonPrefixes.
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

Run the following code to 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

Run the following code to list objects with 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 a
re 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

The marker parameter indicates the name of an object from which the listing begins. Run the following code to specify an object (specified with marker) after which the listing begins:

```
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 obje
cts 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

Use the following code to list all objects on one or more pages. The maximum number of objects listed on each page is specified with 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 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 e
ach 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
        }
    }
}
```

2.5.8.6. Delete objects

This topic describes how to delete objects.

> **Warning** You cannot recover deleted objects. Exercise caution when you delete objects.

For the complete code used to delete objects, visit GitHub.

Delete a single object

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 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)
    }

    // Delete a single object. objectName indicates the full path that includes the extension of the objec
t. Example: abc/efg/123.jpg.
    // To delete a folder, set objectName to the folder name. If the folder is not empty, you must delete
all objects from the folder before you can delete it.
    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.
    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.
    bucket, err := client.Bucket("<yourBucketName>")
    if err ! = nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // List and delete objects whose names contain the specified prefix.
    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)
}
```

2.5.8.7. 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)
```

```
        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 `X-Oss-Meta-`.

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)
    }
}
```

2.5.9. Authorized access

This topic describes how to use 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:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

// After you obtain a temporary STS credential for your OSSClient, an OSSClient instance is created ba
sed on the security token and temporary AccessKey pair (AccessKey ID and AccessKey secret) that are
contained in the credential.
// Create an OSSClient instance.
client, err := oss.New("<yourEndpoint>", "<yourAccessKeyId>", "<yourAccessKeySecret>", oss.Security
Token("<yourSecurityToken>"))
if err ! = nil {
    fmt.Println("Error:", err)
    os.Exit(-1)

// Perform an operation in OSS.
}
```

Use a signed URL to authorize temporary access

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 limit the period of access from visitors.

For the complete code of using a signed URL to authorize temporary access, visit GitHub.

- 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 {
```

```
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, see Manage object metadata.

- 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 to download the object.
    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)
```

```
        HandleError(err)
    }

    body, err := bucket.GetObjectWithURL(signedURL)
    if err ! = nil {
        HandleError(err)
    }
    // Read the object content.
    data, err := ioutil.ReadAll(body)
    body.Close()
    data = data // use data

    // Download the object to a local file by using the signed URL.
    err = bucket.GetObjectToFileWithURL(signedURL, localDownloadedFilename)
    if err ! = nil {
        HandleError(err)
    }
}
```

2.5.10. IMG

Image Processing (IMG) provided by OSS is a secure, cost-effective, and highly reliable image processing service that processes 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 provides the following IMG 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. Choose **Data Processing** > **Image Processing (IMG)**. Click Create Rule to customize an image processing style.
- Call multiple IMG features by using cascade processing.

Usage of IMG

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- Anonymous access

  You can use a third-level domain in the following format to anonymously access processed images:

  ```
  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	Operations on images, such as scaling, cropping, and rotating.
param	The parameter corresponding to the operations on the image.

  - Basic operations

    Example: Scale an image to a width of 100 pixels 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
    ```

  - Custom styles

    Access IMG anonymously by using a third-level domain in the following format:

    ```
    http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
    ```

    - style: the identifier reserved by the system for the custom style.
    - yourStyleName: the name of the custom style. The name is specified by the Rule Name parameter when the custom style is added in the Apsara Stack Cloud Management (ASCM) 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 over HTTPS

  IMG also supports access over HTTPS. Example:

  ```
  https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
  ```

- Authorized access

  Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

  Use the following code to generate a signed URL for 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)
    }

    // Specify the name of 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,p
ng"))
    if err ! = nil {
    HandleError(err)
    } else {
    fmt.Println(signedURL)
    }
}
```

- Access by using SDK

  You can use SDK to access and process an image object.

  SDK allows you to customize image styles, HTTPS access, and cascade operations.

- Basic operations

  Run the following code to perform basic operations on an image:

```
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)
    }

    // Scale 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 styles

  Run the following code to customize an image style:

```
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)
    }

    // Customize image processing styles.
    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

  Run the following code to perform cascade operations on an image:

```
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)
     }

    // Customize image processing styles.
    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)
    }
}
```

Image processing persistence

Run the following code for image processing persistence:

```
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)
    }

    // Specify the name of the bucket.
    bucketName := "<yourBucketName>"
    bucket, err := client.Bucket(bucketName)
    if err ! = nil {
     HandleError(err)
    }


    // Scale the image. By default, the processed image is stored in the same bucket as the source imag
e.
    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(targ
etImageName)))
    result, err := bucket.ProcessObject(sourceImageName, process)
    if err ! = nil {
    HandleError(err)
```

```
    } else {
        fmt.Println(result)
    }

    // Scale the image. Store the processed image in a bucket different from the bucket in which the so
urce image is stored.
    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.
    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(targ
etImageName)))
    result, err = bucket.ProcessObject(sourceImageName, process)
    if err ! = nil {
    HandleError(err)
    } else {
        fmt.Println(result)
    }

    // Rotate the image.
    targetImageName = "example-rotate.png"
    style = "image/rotate,90"
    process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targ
etImageName)))
    result, err = bucket.ProcessObject(sourceImageName, process)
    if err ! = nil {
    HandleError(err)
    } else {
        fmt.Println(result)
    }
```

```
    // Sharpen the image.
    targetImageName = "example-sharpen.png"
    style = "image/sharpen,100"
    process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targ
etImageName)))
    result, err = bucket.ProcessObject(sourceImageName, process)
    if err ! = nil {
    HandleError(err)
    } else {
      fmt.Println(result)
    }

    // Add watermarks to the image.
    targetImageName = "example-watermark.png"
    style = "image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ"
    process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targ
etImageName)))
    result, err = bucket.ProcessObject(sourceImageName, process)
    if err ! = nil {
    HandleError(err)
    } else {
       fmt.Println(result)
    }

    // Convert the image format.
    targetImageName = "example-formatconvert.jpg"
    style = "image/format,jpg"
    process = fmt.Sprintf("%s|sys/saveas,o_%v", style, base64.URLEncoding.EncodeToString([]byte(targ
etImageName)))
    result, err = bucket.ProcessObject(sourceImageName, process)
    if err ! = nil {
    HandleError(err)
    } else {
       fmt.Println(result)
    }
}
```

2.5.11. 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

Error code	Error message	HTTP status code
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
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
FilePartInterity	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

Error code	Error message	HTTP status code
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

Error code	Error message	HTTP status code
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
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

Error code	Error message	HTTP status code
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.

2.6. 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.

Term	Description
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.

2.7. List of operations by function

This topic describes API operations that are provided by OSS.

Service operations

API parameters for service operations

API	Description
GetService	Obtains all the buckets of a specified account.

Bucket operations

API parameters for bucket operations

API	Description
PutBucket	Creates a bucket.
PutBucketACL	Configures bucket ACL.
PutBucketLogging	Enables logging for a bucket.
PutBucketWebsite	Enables static website hosting for a bucket.
PutBucketReferer	Allows you to configure hotlink protection for a bucket.
PutBucketLifecycle	Allows you to configure lifecycle rules for objects in a bucket.
GetBucketAcl	Allows you to query the ACL of a bucket.

API	Description
GetBucketLocation	Allows you to query the location information of the data center to which a specified bucket belongs.
GetBucketLogging	Allows you to query the logging configuration of a bucket.
GetBucketWebsite	Allows you to query the static website hosting status of the bucket.
GetBucketReferer	Allows you to query hotlink protection information of a bucket.
GetBucketLifecycle	Allows you to query lifecycle rules of objects in a bucket.
DeleteBucket	Deletes a bucket.
DeleteBucketLogging	Disables the logging function for a bucket.
DeleteBucketWebsite	Disables static website hosting for a bucket.
DeleteBucketLifecycle	Deletes lifecycle rules of objects in a bucket.
GetBucket(List Object)	Allows you to query information about all the objects in a bucket.
GetBucketInfo	Allows you to query bucket Information.
ListBucketByTagging	Lists buckets that are configured with the specified tag key and tag value.

Object operations

API parameters for object operations

API	Description
PutObject	Uploads an object.
CopyObject	Copies an object.
GetObject	Allows you to query an object.
AppendObject	Uploads an object by appending the object to an existing object.
DeleteObject	Deletes an object.
DeleteMultipleObjects	Deletes multiple objects.
HeadObject	Allows you to query the metadata of an object.
GetObjectMeta	Allows you to query the basic metadata of an object in a bucket.
PostObject	Uploads an object in the POST mode.

API	Description
PutObjectACL	Allows you to configure ACL for an object.
GetObjectACL	Allows you to query the ACL information of an object.

Multipart upload operations

API parameters for multipart upload operations

API	Description
InitiateMultipartUploade	Initializes a multipart upload task.
UploadPart	Uploads an object in multiple parts.
UploadPartCopy	Copies and uploads an object in multiple parts.
CompleteMultipartUpload	Completes multipart upload of the entire object.
AbortMultipartUpload	Cancels a multipart upload task.
ListMultipartUploads	Lists all the ongoing multipart upload tasks.
ListParts	Lists all uploaded parts that are mapped to a specific upload ID.

CORS

API parameters for CORS

API	Description
PutBucketCORS	Allows you to configure cross-origin resource sharing (CORS) rules for a specified bucket.
GetBucketCORS	Allows you to query the CORS rules of a specified bucket.
DeleteBucketCORS	Disables CORS for a specified bucket and deletes all CORS rules.
OptionObject	The preflight request for cross-region access.

2.8. Access control

2.8.1. 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.

2.8.2. 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-sha1(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: eB5eJF1ptWaXm4bijSPyxw==. 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:

1. 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`.
2. 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.
3. Sort all obtained HTTP request headers in alphabetical order.
4. 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`.
5. 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:

1. Set CanonicalizedResource to empty character string ("").

2. 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.
3. 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`
4. 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

1. 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.
2. The HMAC-SHA1 method defined in RFC 2104 is used to calculate the final signature. In this method, the Key is `AccessKeySecret` .
3. 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`).
4. 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.
5. 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 `O***V` .

Signature examples

Request	Signature string calculation formula	Signature string
`PUT /nelson HTTP/1.0 Content-MD5: eB5eJF1ptWaXm4bijSPyxw== 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`	`Signature = base64(hmac-sha1(AccessKeySecret,VERB + "\n" + Content-MD5 + "\n"+ Content-Type + "\n" + Date + "\n" + CanonicalizedOSSHeaders+ CanonicalizedResource))`	`"PUT\n ***w==\n text/html\n Thu, 17 Nov 2005 18:49:58 GMT\n x-oss-magic:a***a\nx-oss-meta-author:***@bar.com\n/oss-example/nelson"`

An example of the signature calculation method:

```
import base64
import hmac
import sha
h = hmac.new("O***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: InvalidAccessKeyId.
- 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 63 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

SDK	Signature implementation
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()
'x\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(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

2.8.3. 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\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.)

2.8.4. 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.

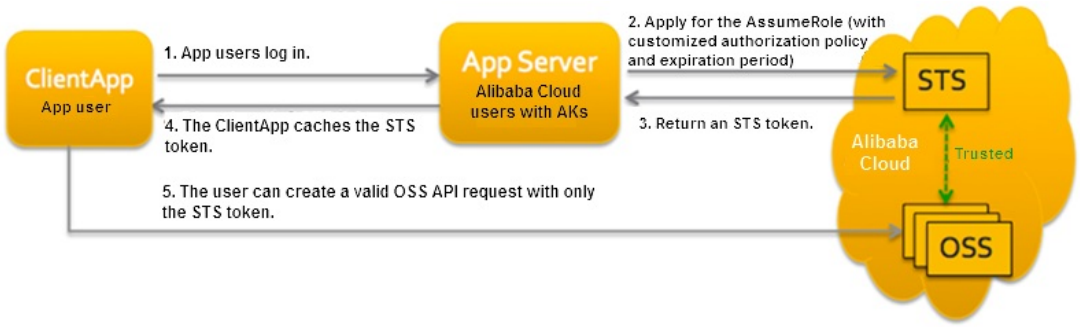
Introduction to STS

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.

2.8.5. 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.

2.9. 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: `<bucketname>.regionid.example.com`. 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
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

2.10. Service-relevant operations

2.10.1. 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.

Request structure

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

Request parameters

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

Request parameters

Parameter	Description
prefix	The prefix the names of returned buckets must contain. If this parameter is not specified, prefix information is not be used as a filter. Type: string Default value: null
marker	The name of the bucket after which the list begins. Buckets are listed in alphabetical order of their names. If this parameter is not specified, all buckets are returned. Type: string Default value: null
max-keys	The maximum number of buckets returned for one request. If this parameter is not specified, the default value is 100. The maximum value cannot exceed 1000. Type: string Default value: 100

Response elements

Response elements

Element	Description
ListAllMyBucketsResult	The container used to store results of the GetService request. Type: container Child nodes: Owner and Buckets Parent nodes: none

Element	Description
Prefix	The prefix of the returned bucket names for one request. This node is available only when not all buckets are returned. Type: string Parent nodes: ListAllMyBucketsResult
Marker	The name of the bucket after which the results are returned for the GetService(ListBucket) request. This node is available only when not all buckets are returned. Type: string Parent nodes: ListAllMyBucketsResult
MaxKeys	The maximum number of buckets returned for one request. This node is available only when not all buckets are returned. Type: string Parent nodes: ListAllMyBucketsResult
IsTruncated	Indicates whether all results are returned. A value of true indicates that not all results are returned this time. A value of false indicates that all results are returned this time. This node is available only when not all buckets are returned. Type: Boolean Valid values: true and false Parent nodes: ListAllMyBucketsResult
NextMarker	The marker for the next GetService (ListBuckets) request to return the remaining results. This node is available only when not all buckets are returned. Type: string Parent nodes: ListAllMyBucketsResult
Owner	The container that stores the information about the bucket owner. Type: container Parent nodes: ListAllMyBucketsResult
ID	The user ID of the bucket owner. Type: string Parent nodes: ListAllMyBucketsResult and Owner

Element	Description
DisplayName	The name of the bucket owner, which is the same as the user ID. Type: string Parent nodes: ListAllMyBucketsResult and Owner
Buckets	The container that stores the information about multiple buckets. Type: container Child nodes: Bucket Parent nodes: ListAllMyBucketsResult
Bucket	The container that stores the bucket information. Type: container Child nodes: Name, CreationDate, and Location Parent nodes: ListAllMyBucketsResult and Buckets
Name	The name of the bucket. Type: string Parent nodes: ListAllMyBucketsResult, Buckets, and Bucket
CreateDate	The time when the bucket was created. The time is in the YYYY-MM-DDTHH:mm:ss.sssZ format. Example: 2011-12-01T12:27:13.000Z. Parent nodes: ListAllMyBucketsResult, Buckets, and Bucket
Location	The region to which the bucket belongs. Type: string Parent nodes: ListAllMyBucketsResult, Buckets, and Bucket
ExtranetEndpoint	The public endpoint to access the bucket. Type: string Parent nodes: ListAllMyBucketsResult, Buckets, and Bucket
IntranetEndpoint	The internal endpoint to access the bucket from ECS within the same region. Type: string Parent nodes: ListAllMyBucketsResult, Buckets, and Bucket

Detail analysis

- 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.

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>51264</ID>
 <DisplayName>51264</DisplayName>
 </Owner>
 <Buckets>
 <Bucket>
 <CreationDate>2015-12-17T18:12:43.000Z</CreationDate>
 <ExtranetEndpoint>regionid.example.com</ExtranetEndpoint>
 <IntranetEndpoint>regionid-internal.example.com</IntranetEndpoint>
 <Location>oss-cn-shanghai</Location>
 <Name>app-base-oss</Name>
 </Bucket>
 <Bucket>
 <CreationDate>2014-12-25T11:21:04.000Z</CreationDate>
 <ExtranetEndpoint>regionid.example.com</ExtranetEndpoint>
 <IntranetEndpoint>regionid-internal.example.com</IntranetEndpoint>
 <Location>regionid</Location>
 <Name>a***3</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>ut_test_put_bucket</ID>
 <DisplayName>ut_test_put_bucket</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>x***0</Name>
 </Bucket>
 </Buckets>
</ListAllMyBucketsResult>
```

2.11. Bucket-relevant operations

2.11.1. 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
```

Detail analysis

- You can use the `x-oss-acl` header in the PUT request to configure bucket access control list (ACL). You can set bucket ACL to public read/write, public read, or private.
- If the requested bucket exists and is owned by the requester, OSS returns 200 OK.
- If the requested bucket exists but is not owned by the requester, OSS returns 409 Conflict and error code BucketAlreadyExists.
- If the bucket you want to create does not conform to naming rules, OSS returns 400 Bad Request and error code InvalidBucketName.
- If the information for user authentication is not imported when you initiate a PutBucket request, OSS returns 403 Forbidden and error code AccessDenied.
- If the number of buckets exceeds the upper limit when you initiate a PutBucket request, OSS returns 400 Bad Request and error code TooManyBuckets.
- If you do not configure ACL when you create the bucket, default ACL private applies.

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 q***c:7***=
```

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
```

2.11.2. PutBucketACL

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

Three ACLs are available for buckets: public read/write, public read, and private. The x-oss-acl header in a PutBucketACL request is used to set the ACL. Only the bucket creator has permissions to set the bucket ACL. If the operation succeeds, 200 is returned. Otherwise, the corresponding error code and message are returned.

Request structure

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

Detail analysis

- If the bucket already exists, the specified ACL in the request is different from the existing ACL, and the requester is the bucket owner, the specified ACL overwrites the existing ACL.
- If the information for user authentication is not imported when you initiate a PutBucket request, OSS returns 403 Forbidden and error code AccessDenied.
- If the x-oss-acl header is not specified in the request, and the bucket already exists and belongs to the requester, the existing ACL of the bucket remains unchanged.

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 q***c: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>
```

2.11.3. 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.

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

Request elements

Element	Description	Required
BucketLoggingStatus	The container that stores the logging status information. Type: container Child nodes: LoggingEnabled Parent nodes: none	Yes
LoggingEnabled	The container that stores the access log information. This element is required only when logging is enabled. Type: container Child nodes: TargetBucket and TargetPrefix Parent nodes: BucketLoggingStatus	No
TargetBucket	The bucket that stores access logs. Type: string Child nodes: none Parent nodes: BucketLoggingStatus and LoggingEnabled	This element is required when logging is enabled.

Element	Description	Required
TargetPrefix	The name prefix of the object that stores access logs. Type: string Child nodes: none Parent nodes: BucketLoggingStatus and LoggingEnabled	No

Naming conventions of access log objects

```
<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

Log object format

Field	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	-	A reserved field.
Reserved	-	A reserved field.
Time	[02/May/2012:00:00:04 +0800]	The time when OSS receives 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 returns.
SentBytes	5576	The traffic consumed when a user downloads data from OSS.
RequestTime (ms)	71	The length of time in milliseconds used to complete the request.

Field	Example	Description
Referer	http://www.aliyun.com/product/oss	The HTTP header that identifies the address of the web page 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.
Request ID	505B01695037C2AF032593A4	The UUID used to uniquely identify this request.
LoggingFlag	true	Indicates whether the access logging feature has been 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 the request.
Bucket	oss-example	The name of the bucket that is requested.
Key	/aliyun-logo.png	The of the object that is requested.
ObjectSize	5576	The size of the 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 returns.
Request Length	302	The length of the request. Unit: bytes.
UserID	1657136103983691	The ID of the 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.

Detail analysis

- The source bucket to which an object belongs and the destination bucket where the processed object is stored must belong to the same user.

- In the preceding request structure, BucketName specifies the bucket for which logging is enabled. TargetBucket specifies the bucket to which access log objects are saved. TargetPrefix specifies the name prefix of the access log objects and can be null.
- The source bucket and destination bucket can be the same or different buckets. You can save logs from multiple source buckets to the same destination bucket. In this case, we recommend that you assign different values to TargetPrefix.
- 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 including signatures. Anonymous access is not supported.
- If the initiator of a PutBucketLogging request is not the owner of the source bucket (BucketName in the sample request), OSS returns the 403 HTTP status code.
- If the source bucket does not exist, OSS returns error code NoSuchBucket.
- If the initiator of a PutBucketLogging request is not the owner of the destination bucket (TargetBucket in the sample request), OSS returns the 403 HTTP status code. If the destination bucket does not exist, OSS returns error code InvalidTargetBucketForLogging.
- The source bucket and destination bucket must belong to the same region. Otherwise, OSS returns HTTP status code 400 and error code InvalidTargetBucketForLogging.
- If the XML format in a PutBucketLogging request is invalid, OSS returns error code MalformedXML.
- The source bucket and destination bucket can be the same bucket. You can save the logs of different source buckets to the same destination bucket. Note that you must set TargetPrefix to different values.
- 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. Examples:

  `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 developers develop log processing tools.

- If the Content-MD5 request header is specified, OSS calculates the MD5 hash of the received content and compares the calculated MD5 hash with the Content-MD5 header value in the request. If the two MD5 hashes are different, OSS returns InvalidDigest.

Examples

Sample requests for enabling bucket logging:

```
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 for disabling bucket logging:

```
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
```

2.11.4. PutBucketWebsite

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

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 elements

Request elements

Element	Description	Required
ErrorDocument	The parent element of the Key child element. Type: container Parent nodes: WebsiteConfiguration	No

Element	Description	Required
IndexDocument	The parent element of the Suffix child element. Type: container Parent nodes: WebsiteConfiguration	Yes
Key	The file name used when the 404 HTTP status code is returned. Type: string Parent nodes: WebsiteConfiguration and ErrorDocument Conditional: This element is required only when ErrorDocument is set.	Conditional
Suffix	The index object name added when a directory URL is returned. This element cannot be empty or contain a forward slash (/). For example, if the index document is set to index.html, regionid.example.com/mybucket/mydir/ contained in a request is converted to regionid.example.com/mybucket/index.html by default. Type: string Parent nodes: WebsiteConfiguration and IndexDocument	Yes
WebsiteConfiguration	The container for the request. Type: container Parent nodes: none	Yes

Detail analysis

- 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.
- To use a custom domain name to access bucket-based static websites, you can use CNAME.
- When you set a bucket to the static website hosting mode, you must specify the index page. The error page is optional.
- When you set a bucket to the static website hosting mode, the specified index page and error page are objects in the bucket.
- 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.
- If the Content-MD5 request header is specified, OSS calculates the MD5 hash of the received content and compares the calculated MD5 hash with the Content-MD5 header value in the request. If the two MD5 hashes are different, OSS returns InvalidDigest.

Examples

Sample requests

```
PUT /? website HTTP/1.1
Host: oss-example.regionid.example.com
Content-Length: 209
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS q***c:K***=

<? xml version="1.0" encoding="UTF-8"? >
<WebsiteConfiguration>
<IndexDocument>
<Suffix>index.html</Suffix>
</IndexDocument>
<ErrorDocument>
<Key>error.html</Key>
</ErrorDocument>
</WebsiteConfiguration>
```

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
```

2.11.5. PutBucketReferer

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

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

Request elements

Element	Description	Required
RefererConfiguration	The container that contains the Referer configurations. Type: container Child nodes: AllowEmptyReferer and RefererList Parent nodes: none	Yes
AllowEmptyReferer	Specifies whether to allow a request whose Refer field is empty. Type: enumerated string Valid values: true and false Default value: true Parent nodes: RefererConfiguration	Yes
RefererList	The container that contains the Referer whitelist. Type: container Parent nodes: RefererConfiguration Child nodes: Referer	Yes

Element	Description	Required
Referer	The Referer whitelist. Type: string Parent nodes: RefererList	No

Detail analysis

- Only the bucket owner can initiate a PutBucketReferer request. Otherwise, OSS returns 403 Forbidden and error code AccessDenied.
- The configurations specified in AllowEmptyReferer replace the previous AllowEmptyReferer configurations. This field is required. By default, AllowEmptyReferer is set to true.
- 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.
- If the Content-MD5 request header is specified, OSS calculates the MD5 hash of the received content and compares the calculated MD5 hash with the Content-MD5 header value in the request. If the two MD5 hashes are different, OSS returns InvalidDigest.

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 q***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 q***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
```

2.11.6. PutBucketLifecycle

The 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.

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 elements

Request elements

Field	Description	Required
CreatedBeforeDate	The last modified date of objects. Specify the time in the ISO 8601 standard. The time must be at 00:00:00 UTC. For example, 2002-10-11T00:00:00.000Z indicates that objects last modified before October 11, 2002 00:00:00.000 UTC are deleted. Objects last modified at or after this time are not deleted. Type: string Parent nodes: Expiration or AbortMultipartUpload	Either Days or CreatedBeforeDate required
Days	The number of days after which the rule takes effect following the last modified time of the object. Type: positive integer Parent nodes: Expiration	Either Days or CreatedBeforeDate required

Field	Description	Required
Expiration	The delete operation on the objects. Type: container Child nodes: Days or CreatedBeforeDate Parent nodes: Rule	No
AbortMultipartUpload	The delete operation on the parts generated from incomplete multipart upload tasks. Type: container Child nodes: Days or CreatedBeforeDate Parent nodes: Rule	No
ID	The unique ID of the rule. The ID can be a maximum of 255 bytes in length. When this parameter is unspecified or null, OSS generates a unique rule ID. Type: string Child nodes: none Parent nodes: Rule	No
LifecycleConfiguration	The container that stores lifecycle configurations. The container can contain up to 1,000 rules. Type: container Chile nodes: Rule Parent nodes: none	Yes
Prefix	The name prefix of the objects to which the rule applies. This rule applies only to objects whose names contain this prefix. The prefixes specified by different rules cannot overlap. Type: string Child nodes: none Parent nodes: Rule	Yes
Rule	The container that contains rule configurations. Type: container Child nodes: ID, Prefix, Status, and Expiration Parent nodes: LifecycleConfiguration	Yes

Field	Description	Required
Status	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. Type: string Parent nodes: Rule Valid values: Enabled or Disabled	Yes

Detail analysis

- Only the bucket owner has permissions to initiate PutBucketLifecycle requests for the bucket. Otherwise, OSS returns 403 Forbidden and error code AccessDenied.
- 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 overwrites the existing one.
- You can call the PutBucketLifecycle operation to set a validity period or expiration date for objects or parts. The parts are generated from incomplete multipart upload tasks.

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>delete objects and parts after one day</ID>
 <Prefix>logs/</Prefix>
 <Status>Enabled</Status>
 <Expiration>
 <Days>1</Days>
 </Expiration>
 <AbortMultipartUpload>
 <Days>1</Days>
 </AbortMultipartUpload>
 </Rule>
 <Rule>
 <ID>delete created before date</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
```

2.11.7. GetBucket

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

Request structure

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

Request parameters

When you initiate a GetBucket (ListObject) request, you can use prefix, marker, delimiter, and max-keys to return specified results. In addition, you can use the encoding-type parameter to encode the following elements in the response: Delimiter, Marker, Prefix, NextMarker, and Key.

Request parameters

Parameter	Description
delimiter	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 and the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Type: string Default value: null
marker	The name of the object after which the list begins. Type: string Default value: null
max-keys	The maximum number of objects that can be returned. If this parameter is not specified, default value 100 is used. The maximum value is 1000. Type: string Default value: 100
prefix	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. Type: string Default value: null

Parameter	Description
encoding-type	The encoding type of the object names in the response. Values of Delimiter, Marker, Prefix, NextMarker, and Key can be 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 0 to 10. You can set the encoding-type parameter to encode values of Delimiter, Marker, Prefix, NextMarker, and Key in the response. Type: string Default value: null Valid value: url

Response elements

Response elements

Element	Description
Contents	The container that stores the returned object metadata. Type: container Parent nodes: ListBucketResult
CommonPrefixes	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 and the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Type: string Parent nodes: ListBucketResult
Delimiter	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 and the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Type: string Parent nodes: ListBucketResult
EncodingType	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. Type: string Parent nodes: ListBucketResult

Element	Description
DisplayName	The name of the object owner. Type: string Parent nodes: ListBucketResult, Contents, and Owner
ETag	The 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 using another method, the ETag value of the content is the universally unique identifier (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 check data integrity. Type: string Parent nodes: ListBucketResult and Contents
ID	The user ID of the bucket owner. Type: string Parent nodes: ListBucketResult, Contents, and Owner
IsTruncated	Indicates whether all results were returned. A value of true indicates that not all results are returned this time. A value of false indicates that all results are returned this time. Type: Boolean Valid values: true and false Parent nodes: ListBucketResult
Key	The name of the object. Type: string Parent nodes: ListBucketResult and Contents
LastModified	The last modified time of the object. Type: time Parent nodes: ListBucketResult and Contents
ListBucketResult	The container that stores the response to the GetBucket request. Type: container Child nodes: Name, Prefix, Marker, MaxKeys, Delimiter, IsTruncated, NextMarker, and Contents Parent nodes: none

Element	Description
Marker	The name of the object after which the list begins. Type: string Parent nodes: ListBucketResult
MaxKeys	The maximum number of objects to return for one request. Type: string Parent nodes: ListBucketResult
Name	The name of the bucket. Type: string Parent nodes: ListBucketResult
Owner	The container that stores the information about the bucket owner. Type: container Child nodes: DisplayName and ID Parent nodes: ListBucketResult
Prefix	The prefix contained in the names of returned objects. Type: string Parent nodes: ListBucketResult
Size	The size of the returned object content. Unit: bytes. Type: string Parent nodes: ListBucketResult and Contents
StorageClass	The storage class of the returned objects. Only Standard is available Type: string Parent nodes: ListBucketResult and Contents

Detail analysis

- User metadata of objects is not returned for the GetBucket request.
- If the bucket you want to access does not exist, or if you access a bucket that cannot be created because the bucket name is invalid, OSS returns 404 Not Found and error code NoSuchBucket.
- If you have no permissions to access the bucket, OSS returns 403 Forbidden and error code AccessDenied.
- If listing cannot be completed at a time because of the max-keys setting, `<NextMarker>` is added to the returned result, indicating 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.
- If the prefix, marker, or delimiter parameter does not meet the length requirement, 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.

Examples

Sample requests

```
GET / HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS q***c:B***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
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>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>00220120222</ID>
      <DisplayName>user-example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/movie/007.avi</Key>
    <LastModified>2012-02-24T08:43:27.000Z</LastModified>
    <ETag>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>00220120222</ID>
      <DisplayName>user-example</DisplayName>
    </Owner>
  </Contents>
<Contents>
    <Key>fun/test.jpg</Key>
    <LastModified>2012-02-24T08:42:32.000Z</LastModified>
    <ETag>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0***2</ID>
      <DisplayName>***</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>oss.jpg</Key>
    <LastModified>2012-02-24T06:07:48.000Z</LastModified>
    <ETag>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
```

```
        <ID>0***2</ID>
        <DisplayName>***</DisplayName>
      </Owner>
    </Contents>
</ListBucketResult>
```

Sample requests that contain the prefix parameter:

```
GET /? prefix=fun HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS q***c:B***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
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>***</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>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0***2</ID>
      <DisplayName>***</DisplayName>
```

```
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/movie/007.avi</Key>
    <LastModified>2012-02-24T08:43:27.000Z</LastModified>
    <ETag>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0***2</ID>
      <DisplayName>***</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/test.jpg</Key>
    <LastModified>2012-02-24T08:42:32.000Z</LastModified>
    <ETag>&amp;quot;5***E&amp;quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0***2</ID>
      <DisplayName>***</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

Sample requests that contain the prefix and delimiter parameters:

```
GET /? prefix=fun/&delimiter=/ HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS q***c:D***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***B
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>&quot;5***E&quot;</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0***2</ID>
      <DisplayName>***</DisplayName>
    </Owner>
  </Contents>
 <CommonPrefixes>
    <Prefix>fun/movie/</Prefix>
 </CommonPrefixes>
</ListBucketResult>
```

2.11.8. 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
```

Response elements

Response elements

Element	Description
AccessControlList	The container that contains the ACL information. Type: container Parent nodes: AccessControlPolicy
AccessControlPolicy	The container that contains the result of the GetBucketACL request. Type: container Parent nodes: none
DisplayName	The name of the bucket owner, which is the same as the user ID. Type: string Parent nodes: AccessControlPolicy and Owner
Grant	The ACL of the bucket. Type: enumerated string Valid values: private, public-read, and public-read-write Parent nodes: AccessControlPolicy and AccessControlList
ID	The user ID of the bucket owner. Type: string Parent nodes: AccessControlPolicy and Owner
Owner	The container that stores the information about the bucket owner. Type: container Parent nodes: AccessControlPolicy

Detail analysis

Only the bucket owner can call the GetBucketACL operation on the bucket.

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-Tupe: application/xml
Connection: keep-alive
Server: AliyunOSS

<? xml version="1.0" ? >
<AccessControlPolicy>
  <Owner>
    <ID>0***2</ID>
    <DisplayName>***</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read</Grant>
  </AccessControlList>
</AccessControlPolicy>
```

2.11.9. 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
```

Response elements

Response elements

Element	Description
LocationConstraint	The region to which the bucket belongs. Type: string

Detail analysis

Only the bucket owner of a bucket can view the location information of the bucket. If other users query the location information, OSS returns 403 Forbidden and error code AccessDenied.

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 for buckets that have logging rules configured

```
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</LocationC
onstraint >
```

2.11.10. GetBucketInfo

You can call this operation to view the information about a bucket.

You can view the following information when you call GetBucketInfo:

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

Request structure

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

Response elements

Response elements

Element	Description
BucketInfo	The container that stores the bucket information content. Type: container Child nodes: Bucket Parent nodes: none
Bucket	The container that stores the specific information about the bucket. Type: container Parent nodes: BucketInfo
CreationDate	The time when the bucket was created. Time format: 2013-07-31T10:56:21.000Z Type: time Parent nodes: BucketInfo and Bucket
ExtranetEndpoint	The public endpoint to access the bucket. Type: string Parent nodes: BucketInfo and Bucket
IntranetEndpoint	The internal endpoint to access the bucket from ECS within the same region. Type: string Parent nodes: BucketInfo and Bucket
Location	The region of the data center to which the bucket belongs. Type: string Parent nodes: BucketInfo and Bucket

Element	Description
Name	The name of the bucket. Type: string Parent nodes: BucketInfo and Bucket
Owner	The container that stores the information about the bucket owner. Type: container Parent nodes: BucketInfo and Bucket
ID	The user ID of the bucket owner. Type: string Parent nodes: BucketInfo, Bucket, and Owner
DisplayName	The name of the bucket owner, which is the same as the user ID. Type: string Parent nodes: BucketInfo, Bucket, and Owner
AccessControlList	The container that contains the ACL information. Type: container Parent nodes: BucketInfo and Bucket
Grant	The ACL of the bucket. Type: enumerated string Valid values: private, public-read, and public-read-write Parent nodes: BucketInfo, Bucket, and AccessControlList

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- Only the bucket owner can view the bucket information. If other uses query the bucket information, OSS returns 403 Forbidden and error code AccessDenied.
- The request can be initiated from any OSS endpoint.

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 success responses when information about the bucket is obtained

```
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>regionid.example.com</ExtranetEndpoint>
  <IntranetEndpoint>regionid-internal.example.com</IntranetEndpoint>
  <Location>regionid</Location>
  <Name>oss-example</Name>
  <Owner>
   <DisplayName>***</DisplayName>
   <ID>2***3</ID>
  </Owner>
  <AccessControlList>
   <Grant>private</Grant>
  </AccessControlList>
 </Bucket>
</BucketInfo>
```

Sample responses when the specified 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 the 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>
```

2.11.11. GetBucketLogging

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

Request structure

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

Response elements

Response elements

Element	Description
BucketLoggingStatus	The container that stores the logging status information. Type: container Child nodes: LoggingEnabled Parent nodes: none
LoggingEnabled	The container that stores the access log information. This element is required only when logging is enabled. Type: container Child nodes: TargetBucket and TargetPrefix Parent nodes: BucketLoggingStatus
TargetBucket	The bucket that stores access logs. Type: string Child nodes: none Parent nodes: BucketLoggingStatus and LoggingEnabled
TargetPrefix	The prefix of the access log object. Type: string Child nodes: none Parent nodes: BucketLoggingStatus and LoggingEnabled

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- Only the bucket owner can view the logging configurations of the bucket. If other users query the configurations, OSS returns 403 Forbidden and error code AccessDenied.
- If no logging rules are set for the source bucket, OSS returns an XML message body where the BucketLoggingStatus element is empty.

Examples

Sample requests

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

2.11.12. 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
```

Response elements

Response elements

Element	Description
ErrorDocument	The parent element of the Key child element. Type: container Parent nodes: WebsiteConfiguration
IndexDocument	The parent element of the Suffix child element. Type: container Parent nodes: WebsiteConfiguration
Key	The file name used when the 404 HTTP status code is returned. Type: string Parent nodes: WebsiteConfiguration and ErrorDocument Conditional: This element is required only when ErrorDocument is set.
Suffix	The index file name added when a directory 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/ contained in an access request is converted into regionid.example.com/mybucket/index.html by default. Type: string Parent nodes: WebsiteConfiguration and IndexDocument
WebsiteConfiguration	The container for the request. Type: container Parent nodes: none

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- Only the bucket owner can view the static website hosting status of the bucket. If other users query the status information, OSS returns 403 Forbidden and error code AccessDenied.

- If the source bucket is not configured with static website hosting, OSS returns 404 and error code NoSuchWebsiteConfiguration.

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 responses for buckets that have logging rules configured

```
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 responses for buckets that have no logging rules configured

```
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>
```

2.11.13. 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
```

Response elements

Response elements

Element	Description
RefererConfiguration	The container that contains the Referer configurations. Type: container Child nodes: AllowEmptyReferer and RefererList Parent nodes: none

Element	Description
AllowEmptyReferer	Indicates whether to allow the request whose Referer field is empty. Type: enumerated string Valid values: true and false Default value: true Parent nodes: RefererConfiguration
RefererList	The container that contains the Referer whitelist. Type: container Parent nodes: RefererConfiguration Child nodes: Referer
Referer	The Referer whitelist. Type: string Parent nodes: RefererList

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- Only the bucket owner can view the Referer configurations of the bucket. If other users query the configurations, OSS returns 403 Forbidden and error code AccessDenied.
- If the bucket has no Referer-related configurations, OSS returns the default AllowEmptyReferer value and an empty RefererList.

Examples

Sample requests

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

2.11.14. GetBucketLifecycle

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

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

Detail analysis

- Only the bucket owner can view the lifecycle configurations of the bucket. If other users query the configurations, OSS returns 403 Forbidden and error code AccessDenied.
- If the bucket or lifecycle configurations do not exist, OSS returns 404 Not Found and error code NoSuchBucket or NoSuchLifecycle.

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>delete after one day</ID>
  <Prefix>logs/</Prefix>
  <Status>Enabled</Status>
  <Expiration>
   <Days>1</Days>
  </Expiration>
 </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>
```

2.11.15. 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
```

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- 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.

Examples

Sample requests

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

2.11.16. 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
```

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- Only the bucket owner can disable logging for the bucket. If you call the DeleteBucketLogging operation on a bucket when you are not authorized, OSS returns 403 Forbidden and error code AccessDenied.
- If logging is disabled for the target bucket, OSS returns the 204 HTTP status code.

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
```

2.11.17. 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
```

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- Only the bucket owner can disable the static website hosting mode for a bucket. If you call the DeleteBucketWebsite operation on a bucket when you are not authorized, OSS returns 403 Forbidden and error code AccessDenied.

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
```

2.11.18. DeleteBucketLifecycle

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

Request structure

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

Detail analysis

- This operation is used to delete all lifecycle rules of a specified bucket. After these rules are deleted, no objects are automatically deleted from this bucket.
- If the bucket or lifecycle configurations do not exist, OSS returns 404 Not Found and error code NoSuchBucket or NoSuchLifecycle.
- Only the bucket owner can delete the lifecycle configurations for a bucket. If you call the DeleteBucketLifecycle operation on a bucket when you are not authorized, OSS returns 403 Forbidden and error code AccessDenied.

Examples

Sample requests

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

2.12. Object-relevant operations

2.12.1. PutObject

You can call this operation to upload objects.

Request structure

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

Request headers

Request headers

Header	Description
Cache-Control	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	The presentational information of the object when the object is downloaded. For more information, visit RFC 2616. Type: string Default value: null

Header	Description
Content-Encoding	The encoding type of the object content when the object is downloaded. For more information, visit RFC 2616. Type: string Default value: null
Content-MD5	The Content-MD5 value of the message content (excluding headers). The value is obtained by calculating the MD5 hash to obtain a 128-bit number and then encoding the number 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. Type: string Default value: null Limits: none
Expires	The time when the cache expires. For more information, visit RFC 2616. Type: string Default value: null Note OSS does not limit or verify this value.
x-oss-server-side-encryption	The server-side encryption algorithm that is used when OSS creates the object. Type: string Valid values: AES256 and SM4
x-oss-object-acl	The access control list (ACL) configured when OSS creates the object. Type: string Valid values: public-read, private, and public-read-write

Detail analysis

- If the Content-MD5 request header is specified, OSS calculates the Content-MD5 header value in the request body and checks whether the two values are the same. If the two values are different, OSS returns InvalidDigest.
- If the value of the Content-Length request header is smaller than the actual length of data transmitted in the request body, OSS still creates an object. However, the object size is equal to the size defined by Content-Length, and the remaining data is dropped.
- If an object with the same name already exists, and you are authorized to access this object, the new object overwrites the existing object, and OSS returns 200 OK.
- If a PutObject request includes a parameter that contains the x-oss-meta- prefix, the

parameter is considered to be user metadata. Example: x-oss-meta-location. An object can have multiple similar parameters, but the total size of all user metadata cannot exceed 8 KB.

- If the Content length parameter is not added to the request, OSS returns 411 Length Required and error code MissingContentLength.
- If the length is specified, but the message body is not sent, or the size of the sent body is smaller than the specified size, the server waits until the request times out and returns 400 Bad Request and error code RequestTimeout.
- If the bucket to which the object is uploaded does not exist, OSS returns 404 Not Found and error code NoSuchBucket.
- If you have no permissions to access the bucket to which the object is uploaded, OSS returns 403 Forbidden error and error code AccessDenied.
- If the length of the added object exceeds 5 GB, OSS returns 400 Bad Request and error code InvalidArgument.
- If the length of the name of the uploaded object exceeds 1,023 bytes, OSS returns 400 Bad Request and error code InvalidObjectName.
- When you upload an Object, OSS supports the following HTTP headers defined in RFC 2616: Cache-Control, Expires, Content-Encoding, Content-Disposition, and Content-Type. If these headers are set when you upload an object, the corresponding header values are automatically set to the values during the upload next time this object is downloaded.
- If the x-oss-server-side-encryption request header is specified when you upload an object, the header value must be set to AES256. If other values are set, OSS returns 400 and error code InvalidEncryptionAlgorithmError. If this request header is specified, the response also contains this header, and OSS stores the uploaded object after encryption. When this object is downloaded, the response contains x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this object.

Examples

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/jpg
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"
```

2.12.2. GetObject

You can call this operation to query an object. Before you call this operation, you must have read permissions on the object.

Request structure

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

Request parameters

OSS allows you to customize parameters in the GET request for OSS to return corresponding headers in the response if the GET request includes a signature. The following table describes the parameters.

Request parameters

Parameter	Description
response-content-type	Specifies the content-type header of the response. Type: string Default value: null
response-content-language	Specifies the content-language header of the response. Type: string Default value: null
response-expires	Specifies the expires header of the response. Type: string Default value: null

Parameter	Description
response-cache-control	Specifies the cache-control header of the response. Type: string Default value: null
response-content-disposition	Specifies the content-disposition header of the response. Type: string Default value: null
response-content-encoding	Specifies the content-encoding header of the response. Type: string Default value: null

Request headers

Request headers

Header	Description
Range	The range of data you want to return. For example, if you specify bytes=0-9, OSS transfers byte 0 to byte 9, a total of 10 bytes. Type: string Default value: null
If-Modified-Since	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. Type: string Default value: null Time format: ddd, dd MMM yyyy HH:mm:ss GMT. Example: Fri, 13 Nov 2015 14:47:53 GMT.
If-Unmodified-Since	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. Type: string Default value: null Time format: ddd, dd MMM yyyy HH:mm:ss GMT. Example: Fri, 13 Nov 2015 14:47:53 GMT.

Header	Description
If-Match	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. Type: string Default value: null
If-None-Match	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 Default value: null

Detail analysis

- You can set the Range parameter in the GetObject request to implement resumable data transfer. We recommend that you use this parameter when you upload large objects.
- If you set Range in the request, the response contains the length of the entire object and the range of bytes copied for this operation. For example, Content-Range: bytes 0-9/44 indicates that the length of the entire object is 44 bytes. The range of copied bytes 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.
- If the format of the time specified by If-Modified-Since is invalid, OSS returns the object and 200 OK.
- 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.
- If the value specified by If-Unmodified-Since or If-Match meets the actual conditions, OSS returns 412 Precondition Failed.
- If the value specified by If-Modified-Since or If-None-Match does not meet the actual conditions, OSS returns 304 Not Modified.
- If the object does not exist, OSS returns 404 Not Found and error code NoSuchKey.
- Parameters in the GetObject request cannot be customized for OSS to return corresponding headers in the response if the access is anonymous.
- When you customize request parameters for OSS to return corresponding headers in the response, OSS sets these headers to the values specified by the request parameters only if the request is successful (when the system returns 200 OK).
- 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.

Examples

Sample requests

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

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 3***1
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: "5***E "
Content-Type: image/jpg
Content-Length: 344606
Server: AliyunOSS
[344606 bytes of object data]
```

Sample requests that include the Range parameter

```
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 q***:q***=
```

Sample responses

```
HTTP/1.1 206 Partial Content
x-oss-request-id: 2***9
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: "5***E "
Accept-Ranges: bytes
Content-Range: bytes 100-900/344606
Content-Type: image/jpg
Content-Length: 801
Server: AliyunOSS
[801 bytes of object data]
```

Sample requests that include parameters to return custom response headers

```
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%25
3B%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: 5***1
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: "5***E "
Content-Length: 344606
Connection: keep-alive
Content-disposition: attachment; filename:testing.txt
Content-language: Chinese
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]
```

2.12.3. CopyObject

You can call this operation to copy an object.

To create a copy for an existing object in OSS, you can send a PUT request to OSS, and add the x-oss-copy-source node to the PUT request to specify the source object you want to copy. OSS automatically determines that this is a copy operation and performs this operation on the server side. If the copy operation is successful, OSS returns the information of the new object to you. You can call this operation to copy objects smaller than 1 GB. To copy an object larger than 1 GB, you must use multipart upload. For more information, see UploadPartCopy.

Request structure

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

Request headers

Request headers

Header	Description
x-oss-copy-source	The source object address. You must have permissions to read the source object. Type: string Default value: null
x-oss-copy-source-if-match	The object copy condition. 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. Type: string Default value: null
x-oss-copy-source-if-none-match	The object copy condition. 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. Type: string Default value: null

Header	Description
x-oss-copy-source-if-unmodified-since	The object copy 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. Type: string Default value: null
x-oss-copy-source-if-modified-since	The object copy condition. If the source object is modified since the time specified in the request, OSS copies the object. Otherwise, OSS returns 304 Not Modified. Type: string Default value: null
x-oss-metadata-directive	Specifies whether to copy the metadata of the source object. If this parameter is set to COPY, OSS copies metadata for the destination object from the source object. If this parameter is set to REPLACE, OSS ignores all metadata values of the source object, and uses the metadata values specified in this request. If this parameter is set to a value other than COPY and REPLACE, OSS returns the 400 HTTP status code. Note that when the value is COPY, the x-oss-server-side-encryption value is not copied. Type: string Default value: COPY Valid values: COPY and REPLACE
x-oss-server-side-encryption	The server-side encryption algorithm based on entropy encoding to encrypt the destination object. Type: string Valid values: AES256 and SM4
x-oss-object-acl	The access control list (ACL) configured when OSS creates the object. Type: string Valid values: public-read, private, and public-read-write

Response elements

Response elements

Element	Description
CopyObjectResult	The container that contains the copy operation result. Type: string Default value: null

Element	Description
ETag	The ETag value of the destination object. Type: string Parent nodes: CopyObjectResult
LastModified	The last modified time of the destination object. Type: string Parent nodes: CopyObjectResult

Detail analysis

- You can call the CopyObject operation to modify the metadata information of an existing object.
- If the source object URL is the same as the destination object URL during the CopyObject operation, the system replaces the metadata of the source object regardless of the value of x-oss-metadata-directive.
- OSS allows the CopyObject request to contain any number of the four headers that define preconditions. For more information about the related logic, see detail analysis of GetObject.
- To complete a CopyObject operation, the requester must have permissions to read the source object.
- The source and destination objects must belong to a region. Otherwise, OSS returns error code 403 Access Denied and error message "Target object does not reside in the same data center as source object".
- OSS charges a CopyObject operation for one GET request to the source bucket, one PUT request to the destination bucket, and the created bucket.
- Request headers for the CopyObject operation start with x-oss- and must be added to the string-to-sign.
- 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 CopyObject response header contains the x-oss-server-side-encryption header, the value of which is set to the encryption algorithm of the destination object. When this destination object is downloaded, the response also contains the x-oss-server-side-encryption header, the value of which is set to the encryption algorithm of this destination object. If the x-oss-server-side-encryption request header is not specified for the CopyObject operation, the destination object is the data that is not encrypted on the server even if the source object is encrypted on the server.
- When OSS receives a CopyObject request in which the x-oss-metadata-directive header is set to COPY (default value), OSS does not copy the x-oss-server-side-encryption value of the source object. Instead, OSS checks whether the x-oss-server-side-encryption header is specified in the request and determines whether to encrypt the new object accordingly.
- When OSS receives a CopyObject request in which the x-oss-server-side-encryption header is set to a value other than AES256, OSS returns HTTP status code 400 and error code InvalidEncryptionAlgorithmError.
- If the size of the source object is larger than 1 GB, OSS returns HTTP status code 400 and error

code EntityTooLarge.

- Objects created by using the AppendObject operation cannot be copied.

Examples

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>
```

2.12.4. AppendObject

You can call this operation to upload an object by appending the object to an existing object. Objects created by using the AppendObject operation are append objects, whereas objects uploaded by using the PutObject operation are normal objects.

Request structure

```
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

Request headers

Header	Description
Cache-Control	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	The name of the object when the object is downloaded. For more information, visit RFC 2616. Type: string Default value: null
Content-Encoding	The encoding type of the object content when the object is downloaded. For more information, visit RFC 2616. Type: string Default value: null
Content-MD5	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 for the message validity check. 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. Type: string Default value: null Limits: none
Expires	The time when the cache expires. For more information, visit RFC 2616. Type: integer Default value: null
x-oss-server-side-encryption	The server-side encryption algorithm that is used when OSS creates the object. Type: string Valid values: AES256 and SM4
x-oss-object-acl	The ACL configured when OSS creates the object. Type: string Valid values: public-read, private, and public-read-write

Response headers

Response headers

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

Association with other operations

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

Detail analysis

- The append and position parameters belong to CanonicalizedResource that must be included in the signature.
- Parameters in a URL must also include append, which specifies that the operation is AppendObject.
- Query parameters in a URL must include position, which specifies the position from which the append of the object starts. The value of position in the first AppendObject operation must be 0. The value of position in the subsequent operation is the current object length. For example, if the position specified in the first AppendObject request is 0, the content-length is 65536. In this case, the position of the second AppendObject request must be set to 65536. Each time an AppendObject operation succeeds, the x-oss-next-append-position response header indicates the position for the next AppendObject request.
- If the position value is different from the current object length, OSS returns HTTP status code 409 and the PositionNotEqualToLength error code. If the preceding error occurs, you can obtain the position for the next AppendObject request from the x-oss-next-append-position

response header, and send an AppendObject request again.

- The request is successful when the value of position is 0 and the length of an append object with the same name is 0 or the append object with the same name does not exist. In other cases, this error occurs because the position and the length of the object do not match.
- If the value of position is 0 and an object with the same name does not exist, you can set headers such as x-oss-server-side-encryption in the AppendObject request in the same way you set in a PutObject request. This is the same as InitiateMultipartUpload. If you add the correct x-oss-server-side-encryption header in an AppendObject request in which the value of position is 0, the x-oss-server-side-encryption header is also included in the response. If meta must be modified, you can use the CopyObject request.
- Although multiple requests may be sent concurrently, even if you set the value of x-oss-next-append-position in one request, the request may still fail because the value is not updated immediately.
- The length limit of an object generated by using AppendObject is the same as that of an object generated by using PutObject.
- Each time an AppendObject operation is performed, the last modified time of this object is updated.
- If the position value is correct and the content with a length of 0 is appended to an existing append object, the status of the object does not change.

Method to calculate the CRC-64 value

The CRC value of an append object is calculated based on the ECMA-182 standard. The calculation method is the same as that of XZ. Use Boost CRC to calculate the CRC-64 value:

```
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 Python crcmod to calculate the CRC-64 value:

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

print do_crc64("123456789")
```

Examples

Sample requests

```
POST /oss.jpg? append&position=0 HTTP/1.1
Host: BucketName.oss.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/jpg
Content-Length: 1717
Authorization: OSS q***:k***=

[1717 bytes of object data]
```

Sample responses

```
HTTP/1.1 200 OK
Date: Wed, 08 Jul 2015 06:57:01 GMT
ETag: "0***0"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-hash-crc64ecma: 14741617095266562575
x-oss-next-append-position: 1717
x-oss-request-id: 5***1
```

2.12.5. DeleteObject

You can call this operation to delete an object.

Request structure

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

Detail analysis

- To delete an object by using DeleteObject, you must have the delete permissions on this object.
- If the object you want to delete does not exist, OSS returns 204 (No Content).

- If the bucket does not exist, OSS returns 404 Not Found.

Examples

Sample requests

```
DELETE /copy_oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 07:45:28 GMT
Authorization: OSS q***:z***=
```

Sample responses

```
HTTP/1.1 204 NoContent
x-oss-request-id: 5***1
Date: Fri, 24 Feb 2012 07:45:28 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

2.12.6. DeleteMultipleObjects

You can call this operation to delete multiple objects in the same bucket by using one 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 returned by OSS contains the result of each deleted object.
- Quiet mode: The message body returned by OSS contains only the results for objects that fail to be deleted. If all objects are deleted, no message body is returned.

Request structure

```
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 parameters

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

Request parameters

Parameter	Description
encoding-type	The encoding type of the object names in the response. The object names can contain any 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. Type: string Default value: null Valid value: url

Request elements

Request elements

Element	Description
Delete	The container that stores the DeleteMultipleObjects request. Type: container Child nodes: Quite (optional) and Object Parent nodes: none

Element	Description
Key	The name of the object you want to delete. Type: string Parent nodes: Object
Object	The container that stores the information about the object. Type: container Child nodes: Key Parent nodes: Delete
Quiet	Specifies whether to enable the Quite mode. Type: Boolean Valid values: true and false Default value: false Parent nodes: Delete

Response elements

Response elements

Element	Description
Deleted	The container that stores the successfully deleted objects. Type: container Child nodes: Key Parent nodes: DeleteResult
DeleteResult	The container that stores the response to the DeleteMultipleObjects request. Type: container Child nodes: Deleted Parent nodes: none
Key	The name of the object on which OSS performs the delete operation. Type: string Parent nodes: Delete

Element	Description
EncodingType	The encoding type of the object names in the response. If the encoding-type parameter is specified in the request, the object names are encoded in the response. Type: string Parent nodes: container

Detail analysis

- You must specify Content-Length and Content-MD5 for the DeleteMultipleObjects request. OSS verifies that the received message body is correct based on the two header values before the delete operation is performed.
- 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.
- By default, the Verbose mode is used for DeleteMultipleObjects.
- If DeleteMultipleObjects is used to delete an object that does not exist, the operation is considered to be successful.
- The message body for DeleteMultipleObjects can be up to 2 MB. 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 request header is specified, OSS calculates the MD5 hash of the received content and compares the calculated MD5 hash with the Content-MD5 header value in the request. If the two MD5 hashes are different, OSS returns InvalidDigest.

Examples

Sample Request 1

```
POST /? delete HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length:151
Content-MD5: o***==
Authorization: OSS q***:+z***=

<? 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 response

```
HTTP/1.1 200 OK
x-oss-request-id: 7***7
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 Request 2

```
POST /? delete HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length:151
Content-MD5: o***A==
Authorization: OSS q***c:W***=

<? 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 response

```
HTTP/1.1 200 OK
x-oss-request-id: 5***1
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

2.12.7. HeadObject

You can call this operation to return only the metadata of a specified object without returning the object content.

Request structure

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

Request headers

Request headers

Header	Description
If-Modified-Since	The information return condition. If the specified time is earlier than the actual object modified time, OSS returns 200 OK and object metadata. Otherwise, OSS returns 304 not modified. Type: string Default value: null
If-Unmodified-Since	The information return condition. If the specified time is the same as or later than the actual object modified time, OSS returns 200 OK and object metadata. Otherwise, OSS returns 412 precondition failed. Type: string Default value: null
If-Match	The information return condition. If the input expected ETag value matches the ETag value of the object, OSS returns 200 OK and object metadata. Otherwise, OSS returns 412 precondition failed. Type: string Default value: null
If-None-Match	The information return condition. If the input ETag value does not match the ETag value of the object, OSS returns 200 OK and object metadata. Otherwise, OSS returns 304 Not Modified. Type: string Default value: null

Detail analysis

- After the HeadObject request is sent, no message body is returned even if the system returns the 200 OK message or an error message.
- When you call HeadObject, you can set query conditions such as If-Modified-Since, If-Unmodified-Since, If-Match, and If-None-Match. For the detailed setting rules, see the related fields in the GetObject request. If the object is not modified, OSS returns 304 Not Modified.
- 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 does not exist, the system returns 404 Not Found.

- If server-side encryption is configured for the object, OSS automatically 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.

Examples

Sample requests

```
HEAD /oss.jpg HTTP/1.1
Host: ***.regionid.example.com
Date: Fri, 24 Feb 2012 07:32:52 GMT
Authorization: OSS q***:J***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***1
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 07:32:52 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "f***8"
Content-Length: 344606
Content-Type: image/jpg
Connection: keep-alive
Server: AliyunOSS
```

2.12.8. GetObjectMeta

You can call this operation to query the basic metadata of an object in a bucket. The basic metadata includes the ETag, Size (object size), and LastModified. The object content is not returned.

Request structure

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

Detail analysis

- After the GetObjectMeta request is sent, no message body is contained in the response.
- You must include the Object meta parameter in the request when you call the GetObjectMeta operation. Otherwise, the request is used for GetObject.
- If the object does not exist, OSS returns 404 Not Found.

- Only some basic metadata of an object in a bucket is returned. The basic metadata includes the ETag, Size (object size), and LastModified. The value of the Content-Length response header indicates the object size.

Examples

Sample requests

```
GET /oss.jpg? objectMeta HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS q***c:C***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***1
Date: Wed, 29 Apr 2015 05:21:12 GMT
ETag: "5***E"
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 344606
Connection: keep-alive
Server: AliyunOSS
```

2.12.9. PutObjectACL

You can call this operation to modify the ACL of an object.

When you call PutObjectACL, you can set the x-oss-object-acl header in the PUT request to configure ACL for an object. Only the bucket owner have permissions to configure object ACL. If the operation succeeds, 200 is returned. Otherwise, the corresponding error code and message are returned. The following object ACLs are available: private, public-read, public-read-write, and default.

Request structure

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

Object ACLs

Object ACLs

ACL	Description
private	The object is a private resource. Only the owner or authorized users of the bucket can read and write the object.
public-read	The object is a public read resource. Only the owner or authorized users of the bucket can write the object. Other users, including anonymous users can only read the object.
public-read-write	Any users, including anonymous users can read and write the object.
default	The ACL of an object is the same as that of the bucket in which the object is stored.

Detail analysis

- The x-oss-object-acl header value must be set to one of the preceding ACLs. If a value other than the preceding ACLs is set, OSS returns 400 Bad Request and InvalidArgument.
- You can call PutObjectACL to configure ACL for an object. In addition, when you write an object, you can also include the x-oss-object-acl header in the request to configure ACL for the object. For example, you can include the x-oss-object-acl header in the request to configure the ACL for an object when you call PutObject.
- When an unauthorized user accesses an object, OSS returns 403 Forbidden, AccessDenied, and You do not have read permission on this object.
- When an unauthorized user accesses an object, OSS returns 403 Forbidden, AccessDenied, and You do not have write permission on this object.
- Only the bucket owner has permissions to call PutObjectACL to modify the ACL of an object in this bucket. When other users call PutObjectACL, OSS returns 403 Forbidden, AccessDenied, and You do not have write acl permission on this object.
- The object ACL takes precedence over its bucket ACL. For example, the bucket ACL is private, but the object ACL is public read/write. When the object is accessed, the object ACL takes precedence over the bucket ACL. If an object has no ACL configured, its bucket ACL applies.

Examples

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 q***c:K***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***1
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

2.12.10. GetObjectACL

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

Request structure

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

Response elements

Response elements

Element	Description
AccessControlList	The container that contains the ACL information. Type: container Parent nodes: AccessControlPolicy
AccessControlPolicy	The container used to store results of the GetObjectACL request. Type: container Parent nodes: none
DisplayName	The name of the bucket owner, which is the same as the user ID. Type: string Parent nodes: AccessControlPolicy and Owner
Grant	The ACL of the object. Type: enumerated string Valid values: private, public-read, and public-read-write Parent nodes: AccessControlPolicy and AccessControlList

Element	Description
ID	The user ID of the bucket owner. Type: string Parent nodes: AccessControlPolicy and Owner
Owner	The container that stores the information about the bucket owner. Type: container Parent nodes: AccessControlPolicy

Detail analysis

- Only the bucket owner can call GetObjectACL to query the ACL of an object in the bucket. If you are not the bucket owner and send a GetObjectACL request, OSS returns 403 Forbidden, AccessDenied, and You do not have read acl permission on this object.
- If a GetObjectACL request is sent but no ACL is set for the object, OSS returns the default ACL for this object, which indicates that the ACL of this object is the same as that of the bucket.

Examples

Sample requests

```
GET /test-object? acl HTTP/1.1
Host: ***.regionid.example.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS q***c:C***=
```

Sample responses

```
HTTP/1.1 200 OK
x-oss-request-id: 5***1
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>0***2</ID>
    <DisplayName>0***2</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

2.12.11. PostObject

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

POST is an alternative of PUT. POST that implements browser-based object upload to a bucket. 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.

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--
```

Form fields

Form fields

Field	Description	Required
OSSAccessKeyId	The AccessKey ID of the bucket owner. Type: string Default value: null Limit: This form field is required when the bucket ACL is not public read/write or when the policy (or Signature) form field is provided.	Conditional
policy	The security policy that 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 to access only buckets whose ACLs are public read/write. Type: string Default value: null Limit: This form field is required when the bucket ACL is not public read/write or when the OSS AccessKey ID (or Signature) form field is provided.	Conditional
Signature	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. Type: string Default value: null Limit: This form field is required when the bucket ACL is not public read/write, or when the OSSAccessKeyId (or Policy) form field is provided.	Conditional
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	The REST-specific headers. For more information, see PutObject. Type: string Default value: null	No
file	The object or text content. It must be the last field in the form. The browser automatically sets Content-Type based on the object type and overwrites the user configuration. Only one object can be uploaded to OSS each time. Type: string Default value: null	Required

Field	Description	Required
key	The name of the object after the upload. To customize the object name after the upload, use the ${filename} variable. For example, if the user uploads the b.jpg object and the key field is set to /user/a/${filename}, the object name after the upload is /user/a/b.jpg. If the object name before the upload contains the path, remove the path from the object name. For example, if the user uploads the a/b/c/b.jpg object, set the key field to b.jpg. If the Key field is set to /user/a/${filename}, the object name after the upload is /user/a/b.jpg. Type: string Default value: null	Required
success_action_redirect	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 any URL. Type: string Default value: null	No
success_action_status	The status code returned to the client after the object is uploaded if success_action_redirect is not specified. Valid values: 200, 201, and 204. The default value is 204. If the value of this form field is set to 200 or 204, OSS returns an empty object and the 200 or 204 HTTP status code. If this field is set to 201, the OSS returns an XML object and the 201 HTTP status code. If the value of this form field is not specified or set to an invalid value, OSS returns an empty object and the 204 HTTP status code. Type: string Default value: null	No
x-oss-meta-*	The user metadata value specified by the user. OSS does not check or use this value. Type: string Default value: null	No
x-oss-server-side-encryption	The server-side encryption algorithm that is used when OSS creates the object. Type: string Valid value: AES256	No

Field	Description	Required
x-oss-object-acl	The ACL configured when OSS creates the object. Type: string Valid values: public-read, private, and public-read-write	No
x-oss-security-token	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 common AccessKey ID. Type: string Default value: null	No

Response headers

Response headers

Field	Description
x-oss-server-side-encryption	If x-oss-server-side-encryption is specified in the request, the response contains this header to indicate the encryption algorithm used. Type: string

Response elements

Response elements

Element	Description
PostResponse	The container that stores the response to the PostObject request. Type: container Child nodes: Bucket, ETag, Key, and Location
Bucket	The name of the bucket. Type: string Parent nodes: PostResponse
ETag	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. Type: string Parent nodes: PostResponse

Element	Description
Location	The URL of the created object. Type: string Parent nodes: PostResponse

Detail analysis

- To perform the PostObject operation, you must have permissions to write the bucket. If the bucket ACL is public read/write, signature information is optional. If the bucket ACL is not public read/write, signature verification must be performed. Unlike PutObject, PostObject uses an AccessKey secret to calculate the signature for policy. The calculated signature string is used as the value for the Signature form field. OSS checks this value to verify the validity of the signature.
- If one of the OSSAccessKeyId, policy, and Signature form fields is specified, the other two form fields are required. If one of the two form fields is missing, OSS returns the InvalidArgument error code.
- The form submitted by the PostObject operation must be encoded in the multipart/form-data format. For example, the Content-Type header must be in the `multipart/form-data; boundary=xxxxx` format, where boundary is a boundary string.
- 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` .
- The policy form field specifies the validity of form fields in the PostObject request. OSS checks the validity of the request based on the policy. If the request is invalid, OSS returns the AccessDenied error code. When OSS checks the validity of form fields, OSS does not check form fields that are excluded in the policy.
- The form and policy must be encoded in UTF-8. The policy is a JSON text encoded in UTF-8 and Base64.
- The PostObject request can contain extra form fields. OSS checks validity of form fields based on policy.
- If the Content-MD5 request header is specified, OSS calculates the MD5 hash of the received content and compares the calculated MD5 hash with the Content-MD5 header value in the request. If the two MD5 hashes are different, OSS returns InvalidDigest.
- If the PostObject request includes a signature in the Authorization header or in a URL, OSS does not check these signatures.
- 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. An object may have multiple similar parameters, but the total size of the user metadata cannot exceed 8 KB.
- The error message returned because the total size of the PostObject request body cannot exceed 5 GB. If the object size exceeds 5 GB, OSS returns the EntityTooLarge error code.
- If the x-oss-server-side-encryption header is specified in the request, you must set the value to AES256. Otherwise, OSS returns HTTP status code 400 and error code InvalidEncryptionAlgorithmError. If this request header is specified, the response also contains this header, and OSS stores the uploaded object after encryption. When this object is downloaded, the response contains x-oss-server-side-encryption, the value of which is set to

the encryption algorithm of this object.

- Form fields are case-insensitive, but their values are case-sensitive.

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"
eyJleHBpcmF0aW9uIjoiMjAxMy0xMi0wMVQxMjowMDowMFoiLCJjb25kaXRpb25zIjpbWyJjb250ZW50LWxlbmd
0aC1yYW5nZSIsIDAsIDEwNDg1NzYwXSx7ImJ1Y2tldCI6ImFoYWhhIn0sIHsiQSI6ICJhIn0seyJrZXkiOiAiQUJDIn
1dfQ==
--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. Although the policy form field is optional for uploading objects to a bucket whose ACL is public read/write, we recommend that you use this field to limit PostObject requests.

- policy example

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

  In PostObject requests, policy must contain expiration and conditions.

- Expiration

  expiration specifies the expiration time of the request. The time follows the ISO 8601 standard. The time must be in GMT. For example, 2014-12-01T12:00:00.000Z indicates that the PostObject request must be sent before 12:00 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/\${filename}"] instead of ["eq", "$key","$key", "user/user1/a.txt"]. The following table lists the conditions supported by the policy.

Conditions

Field	Description
content-length-range	The maximum and minimum acceptable sizes of the object you want to upload. This condition supports the content-length-range match mode.

Field	Description
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	The 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 to return 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 user metadata specified 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 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 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 policy in the PostObject request.

Escape characters

Escape character	Description
\/	Forward slash
\	Backslash

Escape character	Description
\"	Double quotation marks
\$	Dollar sign
\b	Space
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\uxxxx	Unicode character

Post Signature

To verify a PostObject request, you must include the policy and Signature form fields in the HTML form. 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.

2.12.12. Callback

To enable OSS to return callback information of an object to an application server after the object is uploaded to OSS, you need only to include a callback parameter in the upload request that is sent to OSS. This topic describes how to implement upload callback in detail.

> **Note** The following operations support callback: PutObject, PostObject, and CompleteMultipartUpload.

Step 1: Create parameters

- Callback parameter

  A callback parameter is a Base64-encoded string (field) in the JSON format. To create a callback parameter, you must specify the URL (callbackUrl) of the server to which the callback information is sent and the content (callbackBody) of the callback information.

  The following table describes the JSON fields that are included in a callback parameter.

Field	Description	Required
callbackUrl	○ After an object is uploaded, OSS sends a callback request by using the POST method to this URL. The body of the request is the content specified in callbackBody. In most cases, the URL returns an `HTTP/1.1 200 OK` response. The response body must be in the JSON format, and the Content-Length response header value is smaller than 3 MB. ○ You can configure up to five URLs that are separated with semicolons (;) for a request. OSS sends requests to each URL until the first success response is returned. ○ If no URLs are configured or the value of this field is null, OSS determines that the callback feature is not configured. ○ HTTPS-based URLs are supported. ○ To ensure that Chinese characters can be correctly processed, the callback URL must be encoded.	Yes
callbackHost	○ The value of the Host header in the callback request. This field is valid only when callbackUrl is specified. ○ If callbackHost is not specified, the host values are resolved from the URLs of the callbackUrl field and are specified as the value of callbackHost.	No
callbackBody	○ The value of the callback request body. Example: key=$(key)&etag=$(etag)&my_var=$(x:my_var). ○ OSS system variables as well as custom variables and constants are supported. The following table lists supported system variables: Custom variables are passed in by using the callback-var parameter in the PutObject and CompleteMultipart operations and by using form fields in the PostObject operation.	Yes

Field	Description	Required
callbackBodyType	○ The Content-Type header in the callback request. Valid values: application/x-www-form-urlencoded and application/json. Default value: application/x-www-form-urlencoded. ○ If callbackBodyType is set to application/x-www-form-urlencoded, the variables in callbackBodyType are replaced with URL-encoded values. If callbackBodyType is set to application/json, the variables in callbackBodyType are replaced with JSON-formatted values.	No

Examples of the JSON fields:

```
{
"callbackUrl":"121.101.166.30/test.php",
"callbackHost":"oss-cn-hangzhou.aliyuncs.com",
"callbackBody":"{\"mimeType\":${mimeType},\"size\":${size}}",
"callbackBodyType":"application/json"
}
```

```
{
"callbackUrl":"121.43.113.8:23456/index.html",
"callbackBody":"bucket=${bucket}&object=${object}&etag=${etag}&size=${size}&mimeType=${mimeType}&imageInfo.height=${imageInfo.height}&imageInfo.width=${imageInfo.width}&imageInfo.format=${imageInfo.format}&my_var=${x:my_var}"
}
```

The following table lists the system parameters you can configure for callbackBody.

System parameter	Description
bucket	The name of the bucket.
object	The name of the object.
etag	The ETag field configured for the object and returned to the requester.
size	The size of the object to request, which is the total size of the entire object in CompleteMultipartUpload operations.
mimeType	The type of the resource. For example, the resource type of JPEG images are image/jpeg.
imageInfo.height	The height of the image.

System parameter	Description
imageInfo.width	The width of the image.
imageInfo.format	The format of the image. Examples: JPG and PNG.

> **Note** This parameter applies only to image objects. If the object is not an image, the values of imageInfo.height, imageInfo.width, and imageInfo.format are null.

- Create custom parameters by using callback-var

  You can configure custom parameters by using the callback-var parameter. Custom parameters are key-value pairs in map. You can add the required parameters to the map. When a POST callback request is initiated, OSS adds these custom parameters and the system parameters described in the preceding section to the body of the POST request. This method allows these parameters to be obtained by the requester.

  You can create a custom parameter the way you create a callback parameter, which must be passed in the JSON format. The JSON string is a map that consists of key-value pairs of all custom parameters.

  > **Note** The key of a custom parameter must start with x: and be in lowercase letters. Otherwise, OSS returns an error.

  Assume that you must configure two custom parameters x:var1 and x:var2. The value of x:var1 is value1. The value of x:var2 is value2. Created JSON string:

  ```
  {
  "x:var1":"value1",
  "x:var2":"value2"
  }
  ```

Note If the input callback parameter or callback-var parameter is invalid, HTTP status code 400 and error code InvalidArgument are returned. This error occurs in the following scenarios:

- URLs and headers are passed in at the same time to the callback parameter (x-oss-callback) or the callback-var parameter (x-oss-callback-var) in PutObject and CompleteMultipartUpload operations.
- The size of the callback or callback-var parameter exceeds 5 KB. This does not occur in PostObject operations because the callback-var parameter is unavailable in PostObject operations.
- The callback or callback-var parameter is not Base64-encoded or is not in a valid JSON format after being decoded.
- The callbackUrl field that is decoded from the callback parameter includes more than five URLs, or the port in the URL is invalid. Example:

```
{"callbackUrl":"10.101.166.30:test",
  "callbackBody":"test"}
```

- The callbackBody field that is decoded from the callback parameter is null.
- The value of the callbackBodyType field decoded from the callback parameter is not `application/x-www-form-urlencoded` or `application/json`.
- The variables in the callbackBody field decoded from the callback parameter are not in the format of ${var}.
- The variables in the callbackBody field decoded from the callback-var parameter are not in the expected JSON format of `{"x:var1":"value1","x:var2":"value2"...}`.

Step 2: Create a callback request

After you create the callback and callback-var parameters, you must add the parameters to the callback request that is sent to OSS.

You can use the following methods to add parameters:

- Add the parameters to a URL.
- Add the parameters to the header.
- Add the parameters to the form fields in the body of a POST request.

Note You can use only this method to specify callback parameters when you upload objects by using POST requests.

You can use only one of the preceding three methods. If you use more than one method, OSS returns the InvalidArgument error code.

To add the parameters to a request that is sent to OSS, you must use Base64 to encode the JSON string created in the preceding section, and then add the parameters:

- To add the parameters to a URL, add `callback=[CallBack]` or `callback-var=[CallBackVar]` to the request as a URL parameter. When the signature for the CanonicalizedResource field is calculated, callback or callback-var is used as a subresource.

- To add the parameters to the header, add `x-oss-callback=[CallBack]` or `x-oss-callback-var=[CallBackVar]` to the request as a header. When the signature for the CanonicalizedOSSHeaders field is calculated, include x-oss-callback-var and x-oss-callback. Example:

```
PUT /test.txt HTTP/1.1
Host: callback-test.oss-test.aliyun-inc.com
Accept-Encoding: identity
Content-Length: 5
x-oss-callback-var: eyJ4Om15X3ZhciI6ImZvci1jYWxsYmFjay10ZXN0In0=
User-Agent: aliyun-sdk-python/0.4.0 (Linux/2.6.32-220.23.2.ali1089.el5.x86_64/x86_64;2.5.4)
x-oss-callback: eyJjYWxsYmFja1VybCI6IjEyMS40My4xMTMuODoyMzQ1Ni9pbmRleC5odG1sIiwgICJjYWxs
YmFja0JvZHkiOiJidWNrZXQ9JHtidWNrZXR9Jm9iamVjdD0ke29iamVjdH0mZXRhZz0ke2V0YWd9JnNpemU9J
HtzaXplfSZtaW1lVHlwZT0ke21pbWVUeXBlfSZpbWFnZUluZm8uaGVpZ2h0PSR7aW1hZ2VJbmZvLmhlaW
dodH0maW1hZ2VJbmZvLndpZHRoPSR7aW1hZ2VJbmZvLndpZHRofSZpbWFnZUluZm8uZm9ybWF0PSR7
aW1hZ2VJbmZvLmZvcm1hdH0mbXlfdmFyPSR7eDpteV92YXJ9In0=
Host: callback-test.oss-test.aliyun-inc.com
Expect: 100-Continue
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: text/plain
Authorization: OSS mlepou3zr4u7b14:5a74vhd4UXpmyuudV14Kaen5****
Test
```

- Add the parameters to the form fields in the body of a POST request.
  - It is complex to add the callback parameter when the POST method is used to upload an object because the callback parameter must be added by using a separate form field. Example:

```
--9431149156168
Content-Disposition: form-data; name="callback"
eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5waHAiLCJjYWxsYmFja0hvc3
QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9J
Ht4OnRhYmxlfSIsImNhbGxiYWNrQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVybGVuY29k
ZWQifQ==
```

- Each custom parameter uses a separate form field. You cannot add the callback-var parameter to existing fields. Examples of JSON fields for customer parameters:

```
{
"x:var1":"value1",
"x:var2":"value2"
}
```

The form fields in the POST request:

```
--9431149156168
Content-Disposition: form-data; name="callback"
eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5waHAiLCJjYWxsYmFja0hvc3
QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9J
Ht4OnRhYmxlfSIsImNhbGxiYWNrQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVybGVuY29k
ZWQifQ==
--9431149156168
Content-Disposition: form-data; name="x:var1"
value1
--9431149156168
Content-Disposition: form-data; name="x:var2"
value2
```

You can also add callback conditions to the policy (if callback parameters are not added, upload verification is not performed on the policy). Example:

```
{ "expiration": "2014-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "johnsmith" },
    {"callback": "eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5waHAiLCJjYW
xsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZ
SkiLCJjYWxsYmFja0JvZHlUeXBlIjoiYXBwbGljYXRpb24veC13d3ctZm9ybS11cmxlbmNvZGVkIn0="},
    ["starts-with", "$key", "user/eric/"],
  ]
}
```

Step 3: Initiate a callback request

If an object is uploaded, OSS sends the content specified by the callback and callback-var parameters in the request to the application server by using the POST method. Example:

```
POST /index.html HTTP/1.0
Host: 121.43.113.8
Connection: close
Content-Length: 181
Content-Type: application/x-www-form-urlencoded
User-Agent: http-client/0.0.1
bucket=callback-test&object=test.txt&etag=D8E8FCA2DC0F896FD7CB4CB0031BA249&size=5&mimeType
=text%2Fplain&imageInfo.height=&imageInfo.width=&imageInfo.format=&x:var1=for-callback-test
```

Step 4: (Optional) Sign the callback request

If the callback parameter is configured in the request, OSS uses POST to send a callback request to the application server based on the specified callback URL. To verify whether the callback request received by the application server is initiated by OSS, you can sign the callback request.

- Generate a signature

  A callback request is signed by OSS by using the RSA asymmetric algorithm.

  To generate a signature, encrypt the callback string by using a private key. Example:

  ```
  authorization = base64_encode(rsa_sign(private_key, url_decode(path) + query_string + '\n' + body,
  md5))
  ```

  > **Note** In the preceding code, private_key is a private key known only by OSS, path is the resource path included in the callback request, query_string is the query string, and body is the message body specified in the callback request.

  A callback request is signed in the following steps:

  i. Obtain the callback string to sign, which consists of the resource path that is obtained by decoding the URL, the original query string, a carriage return, and the callback message body.

  ii. Sign the callback string by using the RSA encryption algorithm. Use the private key to encrypt the signature string. The hash function used for the signature is MD5.

  iii. Use Base64 to encode the signed result to obtain the final signature and add the signature to the Authorization header in the callback request.

  Example:

```
POST /index.php? id=1&index=2 HTTP/1.0
Host: 121.43.113.8
Connection: close
Content-Length: 18
authorization: kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8O
kQDjC3uG33esH2t****
Content-Type: application/x-www-form-urlencoded
User-Agent: http-client/0.0.1
x-oss-pub-key-url: aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9jYWxsYmFja19wdWJfa2V5X3Yx
LnBlbQ==
bucket=yonghu-test
```

In the preceding code, path is set to `/index.php` ,query_string is set to `? id=1&index=2` , and the body is set to `bucket=yonghu-test` . The final signature is `kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==` .

- Verify the signature

  Signature verification is an inverse process of signing a request. The signature is verified by the application server. Process:

  ```
  Result = rsa_verify(public_key, md5(url_decode(path) + query_string + '\n' + body), base64_decode(a
  uthorization))
  ```

  The fields in the preceding code and the field that are used to sign the request have the same meanings. public_key specifies the public key and authorization specifies the signature in the callback request header. Process of verifying the signature:

  i. The x-oss-pub-key-url header in the callback request stores the Base64-encoded URL of the public key. Therefore, you must decode the Base64-encoded URL to obtain the public key.

  ```
  public_key = urlopen(base64_decode(x-oss-pub-key-url header))
  ```

  > **Note** To ensure that the public key is issued by OSS, you must verify whether the value of the `x-oss-pub-key-url` header starts with `http://gosspublic.alicdn.com/` or `https://gosspublic.alicdn.com/` .

  ii. Obtain the signature decoded in Base64.

  ```
  signature = base64_decode(authorization header)
  ```

  iii. Obtain the string to sign the way described in the process of signing the callback request.

  ```
  sign_str = url_decode(path) + query_string + '\n' + body
  ```

  iv. Verify the signature

```
result = rsa_verify(public_key, md5(sign_str), signature)
```

Complete process of verifying the signature:

i. Obtain the URL of the public key by decoding `aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9jYWxsYmFja19wdWJfa2V5X3YxLnBlbQ==` in Base64. The decoded URL is `http://gosspublic.alicdn.com/callback_pub_key_v1.pem`.

ii. Decode `kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==` in Base64. The decoded result cannot be displayed because it is a nonprintable string.

iii. Obtain the string to sign, which is url_decode("index.php") + "? id=1&index=2" + "\n" + "bucket=yonghu-test", and perform MD5 verification on the string.

iv. Verify the signature

- Example for an application server to verify a signature:

  The following Python code provides an example of how an application server verifies a signature. Before you run the code, the M2Crypto library must be installed.

```
import httplib
import base64
import md5
import urllib2
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
from M2Crypto import RSA
from M2Crypto import BIO
def get_local_ip():
    try:
        csock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        csock.connect(('8.8.8.8', 80))
        (addr, port) = csock.getsockname()
        csock.close()
        return addr
    except socket.error:
        return ""
class MyHTTPRequestHandler(BaseHTTPRequestHandler):
    '''
    def log_message(self, format, *args):
        return
    '''
    def do_POST(self):
        #get public key
        pub_key_url = ''
        try:
```

```
            pub_key_url_base64 = self.headers['x-oss-pub-key-url']
            pub_key_url = pub_key_url_base64.decode('base64')
            if not pub_key_url.startswith("http://gosspublic.alicdn.com/") and not pub_key_url.startswith
("https://gosspublic.alicdn.com/"):
                self.send_response(400)
                self.end_headers()
                return
            url_reader = urllib2.urlopen(pub_key_url)
            #you can cache it
            pub_key = url_reader.read()
        except:
            print 'pub_key_url : ' + pub_key_url
            print 'Get pub key failed!'
            self.send_response(400)
            self.end_headers()
            return
        #get authorization
        authorization_base64 = self.headers['authorization']
        authorization = authorization_base64.decode('base64')
        #get callback body
        content_length = self.headers['content-length']
        callback_body = self.rfile.read(int(content_length))
        #compose authorization string
        auth_str = ''
        pos = self.path.find('?')
        if -1 == pos:
            auth_str = urllib2.unquote(self.path) + '\n' + callback_body
        else:
            auth_str = urllib2.unquote(self.path[0:pos]) + self.path[pos:] + '\n' + callback_body
        print auth_str
        #verify authorization
        auth_md5 = md5.new(auth_str).digest()
        bio = BIO.MemoryBuffer(pub_key)
        rsa_pub = RSA.load_pub_key_bio(bio)
        try:
            result = rsa_pub.verify(auth_md5, authorization, 'md5')
        except:
            result = False
        if not result:
            print 'Authorization verify failed!'
            print 'Public key : %s' % (pub_key)
```

```
            print 'Public key : %s' % (pub_key)
            print 'Auth string : %s' % (auth_str)
            self.send_response(400)
            self.end_headers()
            return
        #do something according to callback_body
        #response to OSS
        resp_body = '{"Status":"OK"}'
        self.send_response(200)
        self.send_header('Content-Type', 'application/json')
        self.send_header('Content-Length', str(len(resp_body)))
        self.end_headers()
        self.wfile.write(resp_body)
class MyHTTPServer(HTTPServer):
    def __init__(self, host, port):
        HTTPServer.__init__(self, (host, port), MyHTTPRequestHandler)
if '__main__' == __name__:
    server_ip = get_local_ip()
server_port = 23451
server = MyHTTPServer(server_ip, server_port)
server.serve_forever()
```

The code for the server in other programming languages:

Java:

- Download link: Java
- Running method: Decompress the package, and run `java -jar oss-callback-server-demo.jar 9000` . 9000 is the port number and can be customized.

PHP:

- Download link: PHP
- Running method: Deploy the code to an Apache environment because some headers in the PHP code depend on the environment. You can modify the example code based on the environment.

Python:

- Download link: Python
- Running method: Decompress the package, and run `python callback_app_server.py` . Before you run the code, RSA dependencies must be installed.

.NET:

- Download link: .NET
- Running method: Decompress the package, and follow `README.md` .

Go:

- Download link: Go
- Running method: Decompress the package, and follow `README.md`.

Ruby:

- Download link: Ruby
- Running method: Run the ruby aliyun_oss_callback_server.rb command.

Step 5: Return the callback result

The application server returns the response to OSS.

An example of the response to the callback request:

```
HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: application/json
Content-Length: 9
{"a":"b"}
```

> **Note** The response returned by the application server to OSS must contain the Content-Length header. The size of the response body cannot exceed 1 MB.

Step 6: Return the upload result

OSS returns the information that is returned by the application server to the user.

An example of the response:

```
HTTP/1.1 200 OK
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: application/json
Content-Length: 9
Connection: keep-alive
ETag: "D8E8FCA2DC0F896FD7CB4CB0031BA249"
Server: AliyunOSS
x-oss-bucket-version: 1442231779
x-oss-request-id: 55F6BF87207FB30F2640C548
{"a":"b"}
```

Note

- The body of responses for some requests such as CompleteMultipartUpload contains content such as information in the XML format. If you use the upload callback feature, the original body content such as `{"a":"b"}` is overwritten. Exercise caution when you implement upload callback.
- If the upload callback fails, HTTP status code 203 and error code CallbackFailed are returned. This response indicates that the object is uploaded to OSS, but the callback fails. A callback failure only indicates that OSS does not receive the expected callback response. It does not indicate that the application server does not receive a callback request. For example, a callback failure occurs if the response returned by the application server is not in the JSON format.

2.12.13. SelectObject

You can call this operation to execute SQL statements on an object and return the execution result.

Note

- You must have read permissions on the object.
- The 206 HTTP status code is returned if the operation is successful. If the SQL statement is not valid or cannot match any objects, the 400 HTTP status code is returned.

Request structure

Request structure for CSV or JSON objects.

- Request structure for CSV objects

```
POST /object? x-oss-process=csv/select HTTP/1.1
HOST: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: time GMT
Content-Length: ContentLength
Content-MD5: MD5Value
Authorization: Signature
<? xml  version="1.0"  encoding="UTF-8"? >
<SelectRequest>
  <Expression>Base64-encoded SQL statement. Example: c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2
JqZWN0IHdoZXJlIF80ID4gNDU=</Expression>
  <InputSerialization>
    <CompressionType>None|GZIP</CompressionType>
    <CSV>
      <FileHeaderInfo>
        NONE|IGNORE|USE
```

```
        </FileHeaderInfo>
        <RecordDelimiter>Base64-encoded character</RecordDelimiter>
        <FieldDelimiter>Base64-encoded character</FieldDelimiter>
        <QuoteCharacter>Base64-encoded character</QuoteCharacter>
        <CommentCharacter>Base64-encoded character</CommentCharacter>
        <Range>line-range=start-end|split-range=start-end</Range>
        <AllowQuotedRecordDelimiter>true|false</AllowQuotedRecordDelimiter>
      </CSV>
      </InputSerialization>
      <OutputSerialization>
         <CSV>
         <RecordDelimiter>Base64-encoded character</RecordDelimiter>
         <FieldDelimiter>Base64-encoded character</FieldDelimiter>

        </CSV>
        <KeepAllColumns>false|true</KeepAllColumns>
        <OutputRawData>false|true</OutputRawData>
        <EnablePayloadCrc>true</EnablePayloadCrc>
        <OutputHeader>false</OutputHeader>
     </OutputSerialization>
    <Options>
      <SkipPartialDataRecord>false</SkipPartialDataRecord>
      <MaxSkippedRecordsAllowed>
      max allowed number of records skipped
      </MaxSkippedRecordsAllowed>
   </Options>
</SelectRequest>
```

- Request structure for JSON objects

```
POST /object? x-oss-process=json/select HTTP/1.1
HOST: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: time GMT
Content-Length: ContentLength
Content-MD5: MD5Value
Authorization: Signature
<? xml  version="1.0"  encoding="UTF-8"? >
<SelectRequest>
   <Expression>
      Base64-encoded SQL statement. Example: c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2JqZWN0IHd
oZXJlIF80ID4gNDU=
   </Expression>
```

```
<InputSerialization>
  <CompressionType>None|GZIP</CompressionType>
  <JSON>
    <Type>DOCUMENT|LINES</Type>
    <Range>
    line-range=start-end|split-range=start-end
    </Range>
    <ParseJsonNumberAsString> true|false
    </ParseJsonNumberAsString>
  </JSON>
</InputSerialization>
<OutputSerialization>
  <JSON>
    <RecordDelimiter>
      Base64 of record delimiter
    </RecordDelimiter>
  </JSON>
  <OutputRawData>false|true</OutputRawData>
      <EnablePayloadCrc>true</EnablePayloadCrc>
</OutputSerialization>
<Options>
  <SkipPartialDataRecord>
    false|true
  </SkipPartialDataRecord>
  <MaxSkippedRecordsAllowed>
    max allowed number of records skipped
   </MaxSkippedRecordsAllowed>
  </Options>
</SelectRequest>
```

Request elements

Element	Type	Description
SelectRequest	Container	The container that stores the SelectObject request. Child nodes: Expression, InputSerialization, and OutputSerialization Parent nodes: none

Element	Type	Description
Expression	String	The Base64-encoded SQL statement. Child nodes: none Parent nodes: SelectRequest
InputSerialization	Container	Optional. This element specifies the input serialization parameters. Child nodes: CompressionType, CSV, and JSON Parent nodes: SelectRequest
OutputSerialization	Container	Optional. This element specifies the output serialization parameters. Child nodes: CSV, JSON, and OutputRawData Parent nodes: SelectRequest
CSV(InputSerialization)	Container	Optional. This element specifies the input serialization parameters when the CSV object is queried. Child nodes: FileHeaderInfo, RecordDelimiter, FieldDelimiter, QuoteCharacter, CommentCharacter, and Range Parent nodes: InputSerialization
CSV(OutputSerialization)	Container	Optional. This element specifies the output serialization parameters when the CSV object is queried. Child nodes: RecordDelimiter and FieldDelimiter Parent nodes: OutputSerialization
JSON(InputSerialization)	Container	Optional. This element specifies the input serialization parameters when the JSON object is queried. Child nodes: Type, Range, and ParseJsonNumberAsString
JSON(OutputSerialization)	Container	Optional. This element specifies the output serialization parameters when the JSON object is queried. Child nodes: RecordDelimiter

Element	Type	Description
Type	Enumeration	The type of the input JSON object: DOCUMENT or LINES.
OutputRawData	Boolean (false by default)	Optional. This element specifies whether to export raw data. Child nodes: none Parent nodes: OutputSerialization Note • If you specify OutputRawData in the request, OSS returns data accordingly. • If you do not specify OutputRawData in the request, OSS automatically selects a format and returns it in the response. • If you explicitly set OutputRawData to True and it takes a long time for the sent SQL statement to return data, the HTTP request may time out.
CompressionType	Enumeration	The compression type of the object. Valid value: None or GZIP. Child nodes: none Parent nodes: InputSerialization
FileHeaderInfo	Enumeration	Optional. This element specifies the header information of the CSV object. Valid values: • Use: The CSV object contains header information. You can use the column name in the CSV object as the column name in the SelectObject operation. • Ignore:The CSV object contains header information. However, you cannot use the column name in the CSV object as the column name in the SelectObject operation. • None: The CSV object does not contain header information. This is the default value. Child nodes: none Parent nodes: CSV (input)

Element	Type	Description
RecordDelimiter	String	Optional. This element specifies a Base64-encoded line break. Default value: `\n`. The value of this element before the value is encoded can be an ANSI value of up to two characters in length. For example, `\n` is used to indicate a line break in Java. Child nodes: none Parent nodes: CSV (input and output) and JSON (output)
FieldDelimiter	String	Optional. This element specifies a Base64-encoded delimiter used to separate columns in the CSV object. Default value: `,`. The value of this element before the value is encoded must be a one-character ANSI value. For example, `,` is used to indicate a comma in Java code. Child nodes: none Parent nodes: CSV (input and output)
QuoteCharacter	String	Optional. This element specifies a Base64-encoded quote character used in the CSV object. Default value: `\"`. In a CSV object, line breaks and column delimiters enclosed in quotation marks are processed as normal characters. The value of this element before the value is encoded must be a one-character ANSI value. For example, `\"` is used to indicate a quote character in Java code. Child nodes: none Parent nodes: CSV (input)
CommentCharacter	String	The comment character used in the CSV object. The value of this element must be Base64-encoded. This value is empty by default.

Element	Type	Description
Range	String	Optional. This element specifies the query range. The following two query methods are supported: Note SelectMeta must be created for objects that are queried based on Range. For more information about SelectMeta, see CreateSelectObjectMeta. • Query by row: line-range=start-end. For example, line-range=10-20 indicates that data from row 10 to row 20 is scanned. • Query by split: split-range=start-end. For example, split-range=10-20 indicates that data from split 10 to split 20 is scanned. The start and end parameters are both inclusive. The two preceding parameters have the same format as that of the range parameter in range get. This parameter can be used only when the object is in CSV format or when the JSON type is LINES. Child nodes: none Parent nodes: CSV (input) and JSON (output)
KeepAllColumns	bool	Optional. A value of true indicates that all columns in the CSV object are included in the response. The default value is false. However, only columns included in the SELECT clause have values. The columns in the response are sorted in ascending order of the column numbers. Example: `select _5, _1 from ossobject.` If you set KeepAllColumns to true and six columns are included in the CSV object, the following result is returned for the preceding SELECT clause: Value of 1st column,,,,Value of 5th column,\n Child nodes: none Parent nodes: OutputSerialization (CSV)

Element	Type	Description
EnablePayloadCrc	bool	The CRC-32 value for verification of each frame. The client can calculate the CRC-32 value of each payload and compare it with the included CRC-32 value to verify data integrity. Child nodes: none Parent nodes: OutputSerialization
Options	Container	Other optional parameters. Child nodes: SkipPartialDataRecord and MaxSkippedRecordsAllowed Parent nodes: SelectRequest
OutputHeader	bool	A value of true indicates that the header information of the CSV object is included in the beginning of the response. Default value: false. Child nodes: none Parent nodes: OutputSerialization
SkipPartialDataRecord	bool	Specifies whether to ignore rows where data is missing. If this parameter is set to false, OSS processes the row data as null without reporting errors. If this parameter is set to true, rows without data are skipped. If the number of skipped rows has exceeded the maximum allowed number, OSS reports an error and stops processing the data. Default value: false. Child nodes: none Parent nodes: Options

Element	Type	Description
MaxSkippedRecordsAllowed	Int	The maximum allowed number of skipped rows. If a row does not match the type specified in the SQL statement, or if one or more columns in a row are missing and the value of SkipPartialDataRecord is True, the rows are skipped. If the number of skipped rows has exceeded the value of this parameter, OSS reports an error and stops processing the data. Note If a row in a CSV object is not properly formatted, OSS stops processing the data and reports an error because this format error may result in incorrect resolution to the CSV object. For example, a column in the row includes continual odd numbered quote characters. This parameter can be used to adjust the tolerance for irregular data but cannot be applied to invalid CSV objects. Default value: 0 Child nodes: none Parent nodes: Options
ParseJsonNumberAsString	bool	Parses integers and floating-point numbers in a JSON object into strings. The precision of floating-point numbers in a JSON object degrades when the numbers are parsed. Therefore, we recommend that you set the value of this parameter to true if you want to retain the raw data. To use the parsed numbers in calculations, you can use the CAST function in SQL to convert the parsed data into the required type such as int, double, or decimal. Default value: false Child nodes: none Parent nodes: JSON

Element	Type	Description
AllowQuotedRecordDelimiter	bool	Specifies whether the CSV object contains line breaks in quotation marks ("). For example, the value of a column is "abc\ndef" and \n is a line break, you must set this value to true. If this parameter is set to false, SelectObject allows you to specify a range in the request header for more efficient multipart query. Default value: true Child nodes: none Parent nodes: InputSerialization

Response body

- If the HTTP status code included in the response is 4xx, the request has not passed the SQL syntax check, or the target object-related errors occur. In this case, the body format of the returned error message is the same as that of the error message returned for a GetObject request.
- If the HTTP status code included in the response is 5xx, internal server errors occur. In this case, the body format of the returned error message is the same as that of the error message returned for a GetObject request.
- HTTP status code 206 is returned when the operation is successful. In this case:
  - If the value of header x-oss-select-output-raw is true, the object data (not including frame-based data) was successfully returned. The client can obtain the data in the same manner as the GetObject operation.
  - If the value of x-oss-select-output-raw is false, the result is returned as frames.
- The frames are in the format of `Version|Frame-Type | Payload Length | Header Checksum | Payload | Payload Checksum<1 byte><--3 bytes--><---4 bytes----><-------4 bytes--><variable><----4bytes---------->` .

  > **Note** The Checksum is CRC-32 in frames. All integers in a frame are big-endian. The value of Version is 1.

Frame Type

The following table describes three frame types supported by SelectObject.

Frame type	Value	Payload format	Description
Data Frame	8388609	offset | data<-8 bytes><---variable->	The data returned for the SelectObject request. The offset parameter is an 8-bit integer that indicates the current scanning location (the offset from the file header) and is used to report the progress of the operation.

Frame type	Value	Payload format	Description
Continuous Frame	8388612	offset<----8 bytes-->	The frame used to report the progress of an operation and keep an HTTP connection. If no data is returned for a query request within five seconds, a continuous frame is returned.
End Frame	8388613	offset | total scanned bytes | http status code | error message<--8bytes-><--8bytes--------><----4 bytes--------><-variable------>	The frame used to return the final status of an operation, including the number of scanned bytes and the possible error messages. • The offset parameter indicates the final location offset after scanning. • The total scanned bytes parameter indicates the size of the scanned data. • The http status code parameter indicates the final status of the operation. Note SelectObject is a stream operation. Only the first data block is processed when the response header is sent. If the first data block matches the SQL statement, the HTTP status code in the response header is 206, which indicates that the operation was successful. However, the final status code may not be 206 because the subsequent data blocks may be invalid. The status code in the response header cannot be modified. Therefore, HTTP status code is included in the end frame to indicate the final status of the operation. The client uses the status code included in the end frame to determine whether the operation is successful. • The error message parameter includes error messages, including the number of each skipped row and the total quantity of skipped rows. Note The format of error messages included in an end frame is `ErrorCodes.DetailMessage`. The ErrorCodes section contains one or more error codes separated with commas (,). ErrorCodes and DetailMessage are separated with a period (.).

Sample requests

Sample requests for CSV or JSON objects.

- Sample requests for CSV objects

```
POST /oss-select/bigcsv_normal.csv? x-oss-process=csv%2Fselect HTTP/1.1
Date: Fri, 25 May 2018 22:11:39 GMT
Content-Type:
Authorization: OSS LTAIJPLocA0fD:FC/9JRbBGRw4o2QqdaL246Px****
User-Agent: aliyun-sdk-dotnet/2.8.0.0(windows 16.7/16.7.0.0/x86;4.0.30319.42000)
Content-Length: 748
Expect: 100-continue
Connection: keep-alive
Host: host name
<? xml version="1.0"? >
<SelectRequest>
  <Expression>c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2JqZWN0IHdoZXJlIF80ID4gNDU=
  </Expression>
  <InputSerialization>
    <Compression>None</Compression>
    <CSV>
      <FileHeaderInfo>Ignore</FileHeaderInfo>
      <RecordDelimiter>Cg==</RecordDelimiter>
      <FieldDelimiter>LA==</FieldDelimiter>
      <QuoteCharacter>Ig==</QuoteCharacter>
      <CommentCharacter>Iw==</CommentCharacter/>
    </CSV>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <RecordDelimiter>Cg==</RecordDelimiter>
      <FieldDelimiter>LA==</FieldDelimiter>
      <QuoteCharacter>Ig==</QuoteCharacter>
    </CSV>
    <KeepAllColumns>false</KeepAllColumns>
      <OutputRawData>false</OutputRawData>
  </OutputSerialization>
</SelectRequest>
```

- Sample requests for JSON objects

```
POST /oss-select/sample_json.json? x-oss-process=json%2Fselect HTTP/1.1
Host: host name
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Darwin/16.7.0/x86_64;3.5.4)
Accept: */*
Connection: keep-alive
date: Mon, 10 Dec 2018 18:28:11 GMT
authorization: OSS AccessKeySignature
Content-Length: 317
<SelectRequest>
  <Expression>c2VsZWN0ICogZnJvbSBvc3NvYmplY3Qub2JqZWN0c1sqXSB3aGVyZSBwYXJ0eSA9ICdEZ
W1vY3JhdCc=
  </Expression>
  <InputSerialization>
  <JSON>
    <Type>DOCUMENT</Type>
  </JSON>
  </InputSerialization>
  <OutputSerialization>
  <JSON>
    <RecordDelimiter>LA==</RecordDelimiter>
  </JSON>
  </OutputSerialization>
  <Options />
</SelectRequest>
```

Regular expressions in an SQL statement

The regular expression in an SQL statement is `SELECT select-list from table where_opt limit_opt` .

> **Note** SELECT, OSSOBJECT, and WHERE are keywords that cannot be modified.

```
select_list: column name
        | column index (Example: _1, _2. column index applies only to CSV objects)
        | json path (Example: s.contacts.firstname. json path applies only to JSON objects)
        | function(column index | column name)
        | function(json_path) (Applies only to JSON objects)
        | select_list AS alias
```

Note The following functions are supported: AVG, SUM, MAX, MIN, COUNT, and CAST (type conversion function). You can use only the wildcard (*) after COUNT.

table: OSSOBJECT

| OSSOBJECT json_path (Applies only to JSON objects)

For CSV objects, table must be OSSOBJECT. For JSON objects (including objects of the DOCUMENT and LINES type), you can specify json_path after OSSOBJECT.

json_path: ['string '] (The quotation marks around string can be deleted if the string does not include a space or a wildcard (*). In this case, [string] is equivalent to 'string'.)

| [n] (Used to indicate the nth element in an array. The value of n is counted from 0.)

| [*] (Used to indicate any child element in an array or object.)

| .'string ' (The quotation marks around string can be deleted if the string does not include a space or a wildcard (*).)

| json_path jsonpath (You can concatenate multiple elements in a json path such as [n].property1.attributes[*].)

```
Where_opt:
| WHERE expr
expr:
| literal value
| column name
| column index
| json path (Applies only to JSON objects)
| expr op expr
| expr OR expr
| expr AND expr
| expr IS NULL
| expr IS NOT NULL
| (column name | column index | json path) IN (value1, value2,...)
| (column name | column index | json path) NOT in (value1, value2,...)
| (column name | column index | json path) between value1 and value2
| NOT (expr)
| expr op expr
| (expr)
| cast (column index |column name | json path | literal as INT|DOUBLE|)
```

- op: includes the following operators: `>`, `<`, `>=`, `<=`, `!=`, `=`, `,`, `,`, `LIKE`, `+`, `-`, `*`, `/`, `%`, and `||`.
- cast: You can use the CAST function to convert data in a column from one type into another.
- The combination of an aggregation function and limit: `Select avg(cast(_1 as int)) from ossobject limit 100`. The preceding statement calculates the average value of the first columns in the first 100 rows. This function differs from the statement supported by MySQL because only a single row is returned for an aggregation function in SelectObject operations. This makes it unnecessary to limit the output data size. The operation for limit is performed before the aggregation operation when you call SelectObject.

Limits on SQL statements

Limits on SQL statements:

- Only text objects encoded in UTF-8 and UTF-8 text objects compressed in the GZIP format are supported. The deflate format is not supported for GZIP objects.
- Only a single object can be queried when you use an SQL statement. The following clauses are not supported: JOIN, ORDER BY, GROUP BY, and HAVING clauses.
- A WHERE clause cannot include aggregation conditions. For example, the following clause is not allowed: WHERE max(cast(age as int)) > 100.
- A maximum of 1,000 columns can be specified for an SQL statement. The column name in an SQL statement can be a maximum of 1,024 bytes in length.
- A maximum of five wildcards (%) are supported in a LIKE clause. The percent sign (%) and the asterisk (*) are wildcards that represent zero or more characters. The ESCAPE keyword is

supported in SQL LIKE clauses, and is used to convert special characters such as percent signs (%), asterisks (*), and question marks (?) into normal strings.

- A maximum of 1,024 constants are supported in an IN clause.
- The Projection after SELECT can be a column name, a CSV column index (such as _1 and _2), an aggregation function, or a CAST function. Other expressions are not supported such as select _1 + _2 from ossobject.
- The maximum column size and row size for a CSV object are 256 KB.
- json path after FROM supports a JSON node with a maximum size of 512 KB. The path can contain up to 10 levels, and an array can contain a maximum of 5,000 elements. The fields after SELECT and WHERE must be from the nodes corresponding to json path after FROM.
- In SQL statements for a JSON object, the SELECT or WHERE expressions cannot include array wildcards ([*]). Array wildcards ([*]) can be included only in json path after FROM. For example, select * from ossobject.contacts[*] instead of select s.contacts[*] from ossobject s is supported.
- The maximum size of an SQL statement is 16 KB. A maximum of 20 expressions can be added after WHERE. Each statement supports up to 10 levels and 100 aggregation operations.

Data error handling

The following section lists common methods to handle data errors.

- Some columns are missing in some rows in a CSV object.

  If the value of SkipPartialDataRecord is not specified or is set to False, OSS calculates the expressions in the SQL statement by processing the values of the missing columns as null.

  If the value of SkipPartialDataRecord is set to True, OSS ignores the rows in which some columns are missing. In this case, if the value of MaxSkippedRecordsAllowed is not specified or is set to a value smaller than the number of skipped rows, OSS reports an error by sending HTTP status code 400 or including HTTP status code 400 in the end frame.

  Assume that the SQL statement is `select _1, _3 from ossobject` and that the data in a row of the CSV object is "John, Company A".

  - If the value of SkipPartialDataRecord is set to False, "John,\n" is returned.
  - If the value of SkipPartialDataRecord is set to True, this row is skipped.

- Some keys are missing in a JSON object.

  Some JSON objects may exclude the keys specified in the SQL statement.

  - If the value of SkipPartialDataRecord is not specified or is set to False, OSS calculates the expressions in the SQL statement by processing the missing keys as null.
  - If the value of SkipPartialDataRecord is True, OSS skips the data in the JSON node. In this case, if the value of MaxSkippedRecordsAllowed is not specified or is set to a value smaller than the number of skipped rows, OSS reports an error by sending HTTP status code 400 or including HTTP status code 400 in the end frame.

  Assume that the SQL statement is `select s.firstName, s.lastName , s.age from ossobject.contacts[*] s` , and that the value of a JSON node is {"firstName":"John", "lastName":"Smith"}.

  - If the value of SkipPartialDataRecord is not specified or is set to False, {"firstName":"John", "lastName":"Smith"} is returned.
  - If the value of SkipPartialDataRecord is set to True, this row is skipped.

Note For keys in the returned data of a request for JSON objects, the output JSON objects can only be LINES. The Key value in the output result is determined based on the following rules.

- Assume that the SQL statement is `select * from ossobject...`. If * corresponds to a JSON object ({...}), it is returned. If * corresponds to a string or an array, it is returned as a DummyKey _1.

  When the data is {"Age":5} `select * from ossobject.Age s where s = 5`, {"_1":5} is returned because 5 that corresponds to * is not a JSON object. When the SQL statement is `select * from ossobject s where s.Age = 5`, {"Age":5} corresponding to * is returned.

- If the SQL statement does not use select * but specifies columns, the returned content is in the `{"{Column 1}": Value, "{Column 2}": Value...}` format. {Column n} can be generated in the following ways:
  - If the alias of the column is specified in the SELECT clause, the alias applies.
  - If the column name is the key of a JSON object, this key is used as the output key value.
  - If the column is an element of a JSON array or an aggregate function, prefix the column name with the serial number (starting from 1) and underscore (_) as the output key value.

  Assume that the data is {"contacts":{"Age":35, "Children":["child1", "child2", "child3"]}}:

  - When the SQL statement is `select s.contacts.Age, s.contacts.Children[0] from oss objects`, Age is the key of the input JSON object, and Children[0] indicates the first element of Children of the array type and is the second column in the output content. {"Age":35, "_2":"child1"} is returned.
  - When the SQL statement is `select max(cast(s.Age as int)) from ossobject.contact s s` and the selected column is an aggregate function, the column is prefixed with _ *1* and its serial number in the output. {"_1":35} is returned.
  - When the alias of the column is specified in SQL statement `select s.contacts.A ge, s.contacts.Children[0] as firstChild from ossobject`, {"Age":35, "firstChild":"child1"} is returned.

- Keys that match the JSON objects and SQL statements are case-sensitive. For example, "select s. Age" and "select s. age" are different.

- The data type of some columns in a CSV object does not match the SQL statement.

  If the data type of a row in a CSV object does not match the type specified in the SQL statement, the row is skipped. If the number of skipped rows exceeds the value of MaxSkippedRecordsAllowed, OSS stops processing data and returns the 400 HTTP status code.

  Assume that the SQL statement is `select _1, _3 from ossobject where _3 > 5`. If the value of a row in the CSV object is `John, Company A, To be hired`, this row is ignored because the third column in the row is not of the integer type.

- The data type of some keys in a JSON object does not match the SQL statement.

  Assume that the SQL statement is `select s.name from ossobject s where s.aliren_age > 5`. If the value of a JSON node is `{"Name":"John", "aliren_age":To be hired}`, this node is skipped.

CreateSelectObjectMeta

You can call this operation to obtain information about the requested CSV object such as the total number of rows, the total number of columns in the CSV object, and the number of splits. If the information does not exist in the object, the entire CSV object is scanned to analyze and record the preceding information. The information obtained in the first API operation call is used when the operation is called again. This way, the entire CSV object does not need to be scanned again.

> **Note**
>
> - You must have the write permissions on the target object before you can perform the CreateSelectObjectMeta operation.
> - The 200 HTTP status code is returned if the operation is successful. If the target object is not a valid CSV or JSON LINES object, or the specified delimiter does not match any CSV objects, the 400 HTTP status code 400 is returned.

- Request structure
  - Request structure for CSV objects

```
POST  /samplecsv? x-oss-process=csv/meta
<CsvMetaRequest>
  <InputSerialization>
    <CompressionType>None</CompressionType>
    <CSV>
      <RecordDelimiter>Base64-encoded character</RecordDelimiter>
      <FieldDelimiter>Base64-encoded character</FieldDelimiter>
      <QuoteCharacter>Base64-encoded character</QuoteCharacter>
    </CSV>
  </InputSerialization>
  <OverwriteIfExists>false|true</OverwriteIfExists>
</CsvMetaRequest>
```

- Request structure for JSON objects

```
POST  /samplecsv? x-oss-process=json/meta
<JsonMetaRequest>
  <InputSerialization>
    <CompressionType>None</CompressionType>
    <JSON>
      <Type>LINES</Type>
    </JSON>
  </InputSerialization>
  <OverwriteIfExists>false|true</OverwriteIfExists>
</JsonMetaRequest>
```

- Request elements

Element	Type	Description
CsvMetaRequest	Container	The container that stores the CreateSelectObjectMeta request for CSV objects. Child nodes: InputSerialization Parent nodes: none
JsonMetaRequest	Container	The container that stores the CreateSelectObjectMeta request for JSON objects. Child nodes: InputSerialization Parent nodes: none
InputSerialization	Container	Optional. This element specifies the input serialization parameters. Child nodes: CompressionType, CSV, and JSON Parent nodes: CsvMetaRequest and JsonMetaRequest
OverwriteIfExists	bool	Optional. This element specifies whether to recalculate the SelectMeta and overwrite the existing data. A value of false indicates that the result is directly returned if the SelectMeta already exists. The default value is false. Child nodes: none Parent nodes: CsvMetaRequest and JsonMetaRequest

Element	Type	Description
CompressionType	Enumeration	Optional. This element specifies the compression type of the object. Only None is supported. Child nodes: none Parent nodes: InputSerialization
RecordDelimiter	String	Optional. This element specifies a Base64-encoded line break used to separate rows in the CSV object. The default value is \n. The value of this element before the value is encoded can be an ANSI value of up to two characters in length. For example, *\n* is used to indicate a line break in Java code. Child nodes: none Parent nodes: CSV
FieldDelimiter	String	Optional. This element specifies a Base64-encoded delimiter used to separate columns in the CSV object. Default value: `,`. The value of this element before being encoded must be a one-character ANSI value. For example, `,` is used to indicate a comma in Java code. Child nodes: none Parent nodes: CSV (input and output)
QuoteCharacter	String	Optional. This element specifies a Base64-encoded quote character used in the CSV object. Default value: \". In a CSV object, line breaks and column delimiters enclosed in quotation marks are processed as normal characters. The value of this element before the value is encoded can be an ANSI value of up to two characters in length. For example, \"is used to indicate a quote character in Java code. Child nodes: none Parent nodes: CSV (input)
CSV	Container	The format of the input CSV object. Child nodes: RecordDelimiter, FieldDelimiter, and QuoteCharacter Parent nodes: InputSerialization

Element	Type	Description
JSON	Container	The format of the input JSON object. Child nodes: Type Parent nodes: InputSerialization
Type	Enumeration	The type of the JSON object. Valid values: LINES

- Response body

  Similar to SelectObject, the results for CreateSelectObjectMeta are also returned as frames.

Frame type	Value	Payload format	Description
Meta End Frame (CSV)	8388614	offset | total scanned bytes | status| splits count | rows count | columns count | error message <-8 bytes><------8 bytes----><--4bytes><--4 bytes--><--8 bytes><--4 bytes---><variable size>	Reports the final status of the CreateSelectObjectMeta operation. ◦ offset: an 8-bit integer, which indicates the offset when the scanning is complete. ◦ total scanned bytes: an 8-bit integer, which indicates the size of the scanned data. ◦ status: a 4-bit integer, which indicates the final status of the operation. ◦ splits_count: a 4-bit integer, which indicates the total number of splits. ◦ rows_count: an 8-bit integer, which indicates the total number of rows. ◦ cols_count: a 4-bit integer, which indicates the total number of columns. ◦ error_message: detailed error messages. If no errors occur, the value of this parameter is null.

Frame type	Value	Payload format	Description
Meta End Frame (JSON)	8388615	offset | total scanned bytes | status| splits count | rows count | error message <-8 bytes><------8 bytes--------><--4bytes><--4 bytes--><--8 bytes><variable size>	Reports the final status of the CreateSelectObjectMeta operation. ◦ offset: an 8-bit integer, which indicates the offset when the scanning is complete. ◦ total scanned bytes: an 8-bit integer, which indicates the size of the scanned data. ◦ status: a 4-bit integer, which indicates the final status of the operation. ◦ splits_count: a 4-bit integer, which indicates the total number of splits. ◦ rows_count: an 8-bit integer, which indicates the total number of rows. ◦ error_message: detailed error messages. If no errors occur, the value of this parameter is null.

- Sample requests
  - Sample requests for CSV objects

```
POST /oss-select/bigcsv_normal.csv? x-oss-process=csv%2Fmeta HTTP/1.1
Date: Fri, 25 May 2018 23:06:41 GMT
Content-Type:
Authorization: OSS AccessKeySignature
User-Agent: aliyun-sdk-dotnet/2.8.0.0(windows 16.7/16.7.0.0/x86;4.0.30319.42000)
Content-Length: 309
Expect: 100-continue
Connection: keep-alive
Host: Host
<? xml version="1.0"? >
<CsvMetaRequest>
  <InputSerialization>
    <CSV>
      <RecordDelimiter>Cg==</RecordDelimiter>
      <FieldDelimiter>LA==</FieldDelimiter>
      <QuoteCharacter>Ig==</QuoteCharacter>
    </CSV>
  </InputSerialization>
  <OverwriteIfExisting>false</OverwriteIfExisting>
</CsvMetaRequest>
```

- Sample requests for JSON objects

```
POST /oss-select/sample.json? x-oss-process=json%2Fmeta HTTP/1.1
Date: Fri, 25 May 2018 23:06:41 GMT
Content-Type:
Authorization: OSS AccessKeySignature
User-Agent: aliyun-sdk-dotnet/2.8.0.0(windows 16.7/16.7.0.0/x86;4.0.30319.42000)
Content-Length: 309
Expect: 100-continue
Connection: keep-alive
Host: Host
<? xml version="1.0"? >
<JsonMetaRequest>
  <InputSerialization>
    <JSON>
      <Type>LINES</Type>
    </JSON>
  </InputSerialization>
  <OverwriteIfExisting>false</OverwriteIfExisting>
</JsonMetaRequest>
```

Supported time formats

You can convert any of the following listed formats to a timestamp without specifying the time format. For example, string cast('20121201' as timestamp) is automatically parsed as a timestamp: December 1, 2012.

The following table describes the recognized time formats.

Format	Description
YYYYMMDD	year month day
YYYY/MM/DD	year/month/day
DD/MM/YYYY/	day/month/year
YYYY-MM-DD	year-month-day
DD-MM-YY	day-month-year
DD.MM.YY	day. month.year
HH:MM:SS.mss	hour:minute:second. millisecond
HH:MM:SS	hour:minute:second
HH MM SS mss	hour minute second millisecond
HH.MM.SS.mss	hour. minute. second. millisecond
HHMM	hour minute
HHMMSSmss	hour minute second millisecond
YYYYMMDD HH:MM:SS.mss	year month day hour:minute:second. millisecond
YYYY/MM/DD HH:MM:SS.mss	year/month/day hour:minute:second. millisecond
DD/MM/YYYY HH:MM:SS.mss	day/month/year hour:minute:second. millisecond
YYYYMMDD HH:MM:SS	year month day hour:minute:second
YYYY/MM/DD HH:MM:SS	year/month/day hour:minute:second
DD/MM/YYYY HH:MM:SS	day/month/year hour:minute:second
YYYY-MM-DD HH:MM:SS.mss	year-month-day hour:minute:second. millisecond
DD-MM-YYYY HH:MM:SS.mss	day-month-year hour:minute:second. millisecond
YYYY-MM-DD HH:MM:SS	year-month-day hour:minute:second
YYYYMMDDTHH:MM:SS	year month day T hour:minute:second

Format	Description
YYYYMMDDTHH:MM:SS.mss	year month day T hour:minute:second. millisecond
DD-MM-YYYYTHH:MM:SS.mss	day-month-year T hour:minute:second. millisecond
DD-MM-YYYYTHH:MM:SS	day-month-year T hour:minute:second
YYYYMMDDTHHMM	year month day T hour minute
YYYYMMDDTHHMMSS	year month day T hour minute second
YYYYMMDDTHHMMSSMSS	year month day T hour minute second millisecond
ISO8601-0	year-month-day T hour:minute+hour:minute, or year-month-day T hour:minute-hour:minute "+" indicates that the time in the current time zone is ahead of standard UTC time. "-" indicates that the time in the current time zone is behind the standard UTC time. In this format, ISO 8601-0 can be used.
ISO8601-1	year-month-day T hour:minute+hour:minute, or year-month-day T hour:minute-hour:minute "+" indicates that the time in the current time zone is ahead of standard UTC time. "-" indicates that the time in the current time zone is behind the standard UTC time. In this format, ISO 8601-1 can be used.
CommonLog	Example: 28/Feb/2017:12:30:51 +0700
RFC822	Example: Tue, 28 Feb 2017 12:30:51 GMT
? D/? M/YY	day/month/year, in which the day and month can be one or two digits.
? D/? M/YY ? H:? M	day/month/year/hour:minute, in which the day, month, hour, and minute can be one or two digits.
? D/? M/YY ? H:? M:? S	day/month/year/hour:minute:second, in which the day, month, hour, minute, and second can be one or two digits.

The formats in the following table are ambiguous. You must specify a time format when you use strings in these formats. For example, the cast('20121201' as timestamp format 'YYYYDDMM') statement incorrectly parses string 20121201 as January 12, 2012.

Format	Description
YYYYDDMM	year day month
YYYY/DD/MM	year/day/month
MM/DD/YYYY	month/day/year
YYYY-DD-MM	year-day-month
MM-DD-YYYY	month-day-year
MM.DD.YYYY	month. day.year

ErrorCode

SelectObject returns error codes in the following ways:

- The HTTP status code is included in the response header, and the error code is included in the response body, which is the same as other OSS requests. Error codes returned in this manner indicate that input errors such as an invalid SQL statement or data errors occur.
- The error code is included in the end frame of the response body. Error codes returned in this manner indicate that the data is incorrect or does not match the SQL statement. For example, a string exists in a column for which the type is set to integer in the SQL statement. In this case, part of data is processed and returned to the client, and the HTTP status code is 206.

Error codes such as InvalidCSVLine can be returned an HTTP status code in the response header or as a status code in the end frame based on the location of the error row within the CSV object.

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
InvalidSqlParameter	Invalid SQL parameter. The SQL statement in the request is null, the SQL statement size has exceeded the limit, or the SQL statement is not Base64-encoded.	400	None
InvalidInputFieldDelimiter	The error message returned because the input CSV object contains invalid column delimiters. The parameter is not Base64-encoded, or the size of the parameter is greater than 1 byte after being decoded.	400	None
InvalidInputRecordDelimiter	The error message returned because the input CSV object contains invalid row delimiters. The parameter is not Base64-encoded, or the size of the parameter is greater than 2 bytes after being decoded.	400	None

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
InvalidInputQuote	The error message returned because the input CSV object contains invalid quote characters. The parameter is not Base64-encoded, or the size of the parameter is greater than 1 byte after being decoded.	400	None
InvalidOutputFieldDelimiter	The error message returned because the output CSV object contains Invalid column delimiters. The parameter is not Base64-encoded, or the size of the parameter is greater than 1 byte after being decoded.	400	None
InvalidOutputRecordDelimiter	The error message returned because the output CSV object contains Invalid column delimiters. The parameter is not Base64-encoded, or the size of the parameter is greater than 2 bytes after being decoded.	400	None
UnsupportedCompressionFormat	The error message returned because the value of the Compression parameter is not NONE or GZIP (case-insensitive).	400	None
InvalidCommentCharacter	The error message returned because the CSV object contains invalid comment characters. The parameter is not Base64-encoded, or the size of the parameter is greater than 1 byte after being decoded.	400	None
InvalidRange	The error message returned because the Range parameter is not prefixed with line-range= or split-range=, or the range value does not comply with the HTTP standard for Range.	400	None
DecompressFailure	The error message returned because the value of Compression is GZIP and the object cannot be decompressed.	400	None
InvalidMaxSkippedRecordsAllowed	The error message returned because the value of MaxSkippedRecordsAllowed is not an integer.	400	None
SelectCsvMetaUnavailable	The error message returned because the target object does not include CSV Meta when the Range parameter is specified. Call the CreateSelectObjectMeta operation first.	400	None
InvalidTextEncoding	The error message returned because the object is not UTF-8 encoded.	400	None

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
InvalidOSSSelectParameters	The error message returned because the EnablePayloadCrc and OutputRawData parameters are both set to True, which results in conflicts.	400	None
InternalError	The error message returned because an OSS system error has occurred.	500 or 206	None or 500
SqlSyntaxError	The error message returned because the syntax of the Base64-decoded SQL statement is invalid.	400	None
SqlExceedsMaxInCount	The error message returned because the number of values included in the SQL IN clause has exceeded 1,024.	400	None
SqlExceedsMaxColumnNameLength	The error message returned because the size of the column name has exceeded 1,024 bytes.	400	None
SqlInvalidColumnIndex	The error message returned because the column index in the SQL statement is smaller than 1 byte or greater than 1,000 bytes in length.	400	None
SqlAggregationOnNonNumericType	The error message returned because an aggregation function is used in a non-numeric column.	400	None
SqlInvalidAggregationOnTimestamp	The error message returned because the SUM or AVG aggregation function is used in the timestamp column.	400	None
SqlValueTypeOfInMustBeSame	The error message returned because values of different types are included in the SQL IN clause.	400	None
SqlInvalidEscapeChar	The error message returned because an invalid escape character such as a question mark (?), percent sign (%), or an asterisk (*) was specified in the SQL LIKE clause.	400	None
SqlOnlyOneEscapeCharIsAllowed	The error message returned because the size of the escape character in the SQL LIKE clause is greater than 1 byte in length.	400	None
SqlNoCharAfterEscapeChar	The error message returned because no characters after the escape character in the SQL LIKE clause exist.	400	None
SqlInvalidLimitValue	The error message returned because the number after the SQL Limit clause is smaller than 1.	400	None

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
SqlExceedsMaxWildCardCount	The error message returned because the number of wildcards (*) or (%) in the SQL LIKE clause has exceeded the limit.	400	None
SqlExceedsMaxConditionCount	The error message returned because the number of conditional expressions in the SQL WHERE clause has exceeded the limit.	400	None
SqlExceedsMaxConditionDepth	The error message returned because the depth of the conditional tree in the SQL WHERE clause has exceeded the limit.	400	None
SqlOneColumnCastToDifferentTypes	The error message returned because a column was converted into different types by including the CAST function in the SQL statement.	400	None
SqlOperationAppliedToDifferentTypes	The error message returned because an operator was used for two objects of different types in the SQL statement. For example, this error code is returned if col1 in _col1 > 3 is a string.	400	None
SqlInvalidColumnName	The error message returned because a column name used in the SQL statement was excluded in the header of the CSV object.	400	None
SqlNotSupportedTimestampFormat	The error message returned because the timestamp format specified in the SQL CAST clause is not supported.	400	None
SqlNotMatchTimestampFormat	The error message returned because the timestamp format specified in the SQL CAST clause does not match the timestamp string.	400	None
SqlInvalidTimestampValue	The error message returned because no timestamp formats are specified in the SQL CAST clause and the provided string cannot be converted to a timestamp.	400	None
SqlInvalidLikeOperand	The error message returned because column names or indexes are not specified in the SQL LIKE clause, the specified column is not of the string type, or the right column in the LIKE clause is of the string type.	400	None
SqlInvalidMixOfAggregationAndColumn	The error message returned because the SQL SELECT clause includes column names and indexes for both aggregation functions and non-aggregation functions.	400	None

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
SqlExceedsMaxAggregationCount	The error message returned because the number of aggregation functions included in the SQL SELECT clause has exceeded the limit.	400	None
SqlInvalidMixOfStarAndColumn	The error message returned because an asterisk (*), a column name, and a column index are included in the same SQL statement.	400	None
SqlInvalidKeepAllColumnsWithAggregation	The error message returned because the SQL statement includes aggregation functions while the KeepAllColumns parameter is set to True.	400	None
SqlInvalidKeepAllColumnsWithDuplicateColumn	The error message returned because the SQL statement includes repeated column names or column indexes while the KeepAllColumns parameter is set to True.	400	None
SqlInvalidSqlAfterAnalysis	The error message returned because the SQL statement is not supported because the SQL statement is too complex to parse.	400	None
InvalidArithmeticOperand	The error message returned because the SQL statement contains arithmetic operations performed on non-numeric constants or columns.	400	None
SqlInvalidAndOperand	The error message returned because the expressions connected by the AND operator in the SQL statement are not of the Boolean type.	400	None
SqlInvalidOrOperand	The error message returned because the expressions connected by the OR operator in the SQL statement are not of the Boolean type.	400	None
SqlInvalidNotOperand	The error message returned because the expressions connected by the NOT operator in the SQL statement are not of the Boolean type.	400	None
SqlInvalidIsNullOperand	The error message returned because the SQL statement specifies the IS NULL operator-based operations performed on a constant.	400	None
SqlComparerOperandTypeMismatch	The error message returned because the SQL statement specifies the comparison operator-based operations performed on two objects of different types.	400	None
SqlInvalidConcatOperand	The error message returned because the SQL statement contains two constants connected by the concatenation operator (||).	400	None

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
SqlUnsupportedSql	The error message returned because the SQL statement is too complex so that the size of the generated SQL plan has exceeded the limit.	400	None
HeaderInfoExceedsMaxSize	The error message returned because the size of the header information specified in the SQL statement has exceeded the limit.	400	None
OutputExceedsMaxSize	The error message returned because a row of output results has exceeded the size limit.	400	None
InvalidCsvLine	The error message returned because a row in the CSV object is invalid or has exceeded the size limit, or the number of skipped rows has exceeded the value of MaxSkippedRecordsAllowed.	206 or 400	400 or None
NegativeRowIndex	The error message returned because the value of the array index in the SQL statement is a negative number.	400	None
ExceedsMaxNestedColumnDepth	The error message returned because the number of nested levels of the JSON object in the SQL statement has exceeded the level limit.	400	None
NestedColumnNotSupportInCsv	The error message returned because the SQL statement contains nested columns that include periods (.) or arrays with brackets ([]), which are not supported for SQL statements of CSV objects.	400	None
TableRootNodeOnlySupportInJson	The error message returned because the root node path can be specified only after From ossobject in JSON objects.	400	None
JsonNodeExceedsMaxSize	The error message returned because the size of the root node in the JSON object has exceeded the limit.	400 or 206	None or 400
InvalidJsonData	The error message returned because the JSON data is formatted incorrectly.	400 or 206	None or 400
ExceedsMaxJsonArraySize	The error message returned because the number of elements in an array in the root node of the JSON object has exceeded the limit.	400 or 206	None or 400

ErrorCode	Description	HTTP Status Code	Http Status Code in End Frame
WildCardNotAllowed	The error message returned because the wildcard (*) cannot be used in SQL SELECT clauses or SQL WHERE clauses for the JSON object. For example, the following statement is not supported: `select s.a.b[*] from ossobject where a.c[*] > 0` .	400	None
JsonNodeExceedsMaxDepth	The error message returned because the depth of the root node of the JSON object has exceeded the limit.	400 or 206	None or 400

2.13. Multipart upload-relevant operations

2.13.1. 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.

2.13.2. 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.

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.

Request parameters

Parameter	Description
encoding-type	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 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 name in the response. Set the value to url. Type: string Default value: null

Request headers

Header	Description
Cache-Control	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	The presentational information of the object during the download. For more information, visit RFC 2616. Type: string Default value: null
Content-Encoding	The content encoding type of the object during the download. For more information, visit RFC 2616. Type: string Default value: null
Expires	The expiration time in milliseconds. For more information, visit RFC 2616. Type: integer Default value: null
x-oss-server-side-encryption	The encryption algorithm that OSS uses to encrypt each uploaded part of an object before storage. Type: string Valid values: AES256 and SM4

Response elements

Response elements

Element	Description
Bucket	The name of the bucket for which the multipart upload task is initiated. Type: string Parent nodes: InitiateMultipartUploadResult
InitiateMultipartUploadResult	The container that stores the response to the InitiateMultipartUpload request. Type: container Child nodes: Bucket, Key, and UploadId Parent nodes: none
Key	The name of the object for which the multipart upload task is initiated. Type: string Parent nodes: InitiateMultipartUploadResult
UploadId	The unique ID of the multipart upload task. Type: string Parent nodes: InitiateMultipartUploadResult
EncodingType	The encoding type of the object name in the response. If the encoding-type parameter is specified in the request, the object name is encoded in the response. Type: string Parent nodes: container

Detail analysis

- When you perform this operation to calculate the signature for authentication, you must add `?uploads` to CanonicalizedResource.
- The InitiateMultipartUpload request supports standard HTTP request headers Cache-Control, Content-Disposition, Content-Encoding, Content-Type, and Expires, as well as custom headers that start with `x-oss-meta-`.
- The InitiateMultipartUpload request does not affect existing objects with the same name.
- After the server receives the InitiateMultipartUpload request, the server returns a message body that is in the XML format. The request body contains the following nodes: Bucket, Key, and UploadID. Record the upload ID for subsequent multipart-related operations.
- 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.

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>
```

2.13.3. UploadPart

After the InitiateMultipartUpload request is completed, you can upload data in parts based on the specified object name and upload ID.

Each part has a part number that ranges from 1 to 10000. Numbers of parts that share the same upload ID uniquely identify each part and part positions in the object. If you upload data by using the same part number, OSS overwrites the existing data identified by this part number. The minimum size of each part, except the last part, is 100 KB. The size of the last part is not limited.

Request structure

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

Detail analysis

- Before you call UploadPart to upload a part, you must call InitiateMultipartUpload to obtain an upload ID issued by the OSS server.
- 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 size of the uploaded part because it 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.
- The part number ranges from 1 to 10000. If a part number is not within this range, OSS returns the InvalidArgument error code.
- 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, indicating the server-side encryption method of the part. For more information, see InitiateMultipartUpload.
- To ensure data integrity, OSS calculates the MD5 hash of the received data and compares the MD5 hash with the Content-MD5 header value. If the two MD5 hashes are different, OSS returns the InvalidDigest error code.

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 q***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
```

2.13.4. UploadPartCopy

You can call this operation to copy data from an existing object to upload a part.

To call UploadPartCopy, you can add the x-oss-copy-source request header to the UploadPart request. To copy an object larger than 1 GB, use UploadPartCopy. For more information about how to upload an object smaller than 1 GB, see CopyObject.

Request structure

```
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

The following table describes the x-oss-copy-source and x-oss-copy-source-range request headers except for common request headers in the UploadPartCopy request.

Request headers

Header	Description
x-oss-copy-source	The source object address. You must have permissions to read the source object. Type: string Default value: null
x-oss-copy-source-range	The range of bytes you want to copy from the source object. For example, if you specify bytes=0-9, the system transfers byte 0 to byte 9, a total of 10 bytes. This request header is not required when the entire source object is copied. Type: integer Default value: null

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

Request headers

Header	Description

Header	Description
x-oss-copy-source-if-match	The 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. Otherwise, OSS returns 412 Precondition Failed. Type: string Default value: null
x-oss-copy-source-if-none-match	The object copy condition. If the ETag value of the source object is different from the ETag value specified in the request, OSS copies the object. Otherwise, OSS returns 412 Precondition Failed. Type: string Default value: null
x-oss-copy-source-if-unmodified-since	The object copy 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. Type: string Default value: null
x-oss-copy-source-if-modified-since	The object copy condition. If the source object is modified after the time specified in the request, OSS copies the object. Otherwise, OSS returns 412 Precondition Failed. Type: string Default value: null

Response elements

Response elements

Field	Description
x-oss-copy-source-if-match	The 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. Otherwise, OSS returns 412 Precondition Failed. Type: string Default value: null
x-oss-copy-source-if-none-match	The object copy condition. If the ETag value of the source object is different from the ETag value specified in the request, OSS copies the object. Otherwise, OSS returns 412 Precondition Failed. Type: string Default value: null

Field	Description
x-oss-copy-source-if-unmodified-since	The object copy 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. Type: string Default value: null
x-oss-copy-source-if-modified-since	The object copy condition. If the source object is modified after the time specified in the request, OSS copies the object. Otherwise, OSS returns 412 Precondition Failed. Type: string Default value: null

Detail analysis

- Before you call UploadPart to upload a part, you must call InitiateMultipartUpload to obtain an upload ID issued by the OSS server.
- 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 size of the uploaded part because it cannot immediately determine which part is the last part. The size of the uploaded part is verified only when multipart upload is complete.
- 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 copied for this operation. For example, Content-Range: bytes 0-9/44 indicates that the length of the entire object is 44 bytes. The range of bytes copied is from byte 0 to byte 9. If the specified range is invalid, OSS copies the entire object and does not include Content-Range in the response.
- 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 header of UploadPart, indicating the server-side encryption method of the part.
- CopyObject cannot be used to copy the objects created by using AppendObject.

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 q***l=
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: 3***a
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>"5***E"</ETag>
</CopyPartResult>
```

2.13.5. CompleteMultipartUpload

You can call this operation to complete multipart upload of an object.

The CompleteMultipartUpload request must contain a list of all valid parts (including the part numbers and ETag values) of the object. After OSS receives the request, OSS checks the validity of each part. After all these parts are validated, OSS combines these parts into a complete object.

Request structure

```
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 parameters

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

Request parameters

Parameter	Description
encoding-type	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 name in the response. Set the value to url. Type: string Default value: null Valid value: url

Request elements

Request elements

Element	Description
CompleteMultipartUpload	The container that stores the content of the CompleteMultipartUpload request. Type: container Child nodes: Part Parent nodes: none
ETag	The ETag values that are returned by OSS after parts are uploaded. Type: string Parent nodes: Part
Part	The container that stores the uploaded parts. Type: container Child nodes: ETag and PartNumber Parent nodes: InitiateMultipartUploadResult
PartNumber	The number of parts. Type: integer Parent nodes: Part

Response elements

Response elements

Element	Description

Element	Description
Bucket	The name of the bucket. Type: string Parent nodes: CompleteMultipartUploadResult
CompleteMultipartUploadResult	The container that stores the response to the CompleteMultipartUpload request. Type: container Child nodes: Bucket, Key, ETag, and Location Parent nodes: none
ETag	The UUID of the object content. The ETag is created to identify the content of an object when the object is created using the CompleteMultipartUpload request. The ETag value of the object can be used to check whether the object content is modified. Type: string Parent nodes: CompleteMultipartUploadResult
Location	The URL of the created object. Type: string Parent nodes: CompleteMultipartUploadResult
Key	The name of the created object. Type: string Parent nodes: CompleteMultipartUploadResult
EncodingType	The encoding type of the object name in the response. If the encoding-type parameter is specified in the request, the object name is encoded in the response. Type: string Parent nodes: container

Detail analysis

- When you perform the CompleteMultipartUpload operation, OSS checks whether each part except the last part is larger than 100 KB in size and whether the part number and ETag are provided for each part in the list. Therefore, when parts are uploaded, the client needs to record the part number and the ETag value returned by the OSS each time a part is uploaded.
- It takes a while for the OSS to process the CompleteMultipartUpload request. During this period of time, if the client is disconnected from OSS, OSS continues to complete the request.
- Part numbers that are submitted in ListParts can be non-consecutive. For example, the first part number is 1 and the second part number is 5.
- After OSS processes the CompleteMultipartUpload request, the upload ID of this task becomes

invalid.

- An object may have different upload IDs that are independent of each other. When one upload task is completed, the corresponding upload ID becomes invalid and other upload IDs are not affected.
- If the InitiateMultipartUpload operation is called and the x-oss-server-side-encryption request header is specified, the x-oss-server-side-encryption response header is returned in response to the CompleteMultipartUpload request. The x-oss-server-side-encryption response header indicates the server-side encryption method for the object.
- If the Content-MD5 request header is specified, OSS calculates the Content-MD5 header value in the request body and checks whether the two values are the same. If they are different, InvalidDigest is returned.

Examples

Sample requests

```
POST /multipart.data? uploadId=0***4  HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 1056
Date: Fri, 24 Feb 2012 10:19:18 GMT
Authorization: OSS q***0=

<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"3***9"</ETag>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <ETag>"8***A"</ETag>
  </Part>
  <Part>
    <PartNumber>8</PartNumber>
    <ETag>"8***2"</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: 5***e
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">
   <Location>http://***.regionid.example.com /multipart.data</Location>
   <Bucket>oss-example</Bucket>
   <Key>multipart.data</Key>
   <ETag>B***D-3</ETag>
</CompleteMultipartUploadResult>
```

2.13.6. AbortMultipartUpload

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

After a multipart upload task is canceled, this upload ID cannot be used to perform any operations and the uploaded parts are deleted.

Request structure

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

Detail analysis

- When you cancel a multipart upload task, you cannot delete parts that are being uploaded. Therefore, if any parts are being uploaded, you may need to cancel the multipart upload task multiple times to release all storage that is used by all parts.
- If the entered Upload ID does not exist, OSS returns HTTP status code 404 and error code NoSuchUpload.

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***l=
```

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
```

2.13.7. ListMultipartUploads

You can call this operation to list all ongoing multipart upload tasks that have been initiated but not completed 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 MultipartUpload tasks.

Request structure

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

Request parameters

Request parameters

Parameter	Description
delimiter	The delimiter used to group objects by name. 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 and the next occurrence of the delimiter are grouped as a single result element in CommonPrefixes. Type: string

Parameter	Description
max-uploads	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
key-marker	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 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. Type: string
prefix	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. Type: string
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 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. Type: string
encoding-type	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. Type: string Default value: null

Response elements

Response elements

Element	Description
ListMultipartUploadsResult	The container that stores the response of the ListMultipartUpload request. Type: container Child nodes: Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MasUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated, and Upload Parent nodes: none
Bucket	The name of the bucket. Type: string Parent nodes: ListMultipartUploadsResult
EncodingType	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. Type: string Parent nodes: ListMultipartUploadsResult
KeyMarker	The name of the object after which the list begins. Type: string Parent nodes: ListMultipartUploadsResult
UploadIdMarker	The upload ID of the multipart upload task after which the list begins. Type: string Parent nodes: ListMultipartUploadsResult
NextKeyMarker	The object name marker in the response for the next request to return the remaining results. Type: string Parent nodes: ListMultipartUploadsResult
NextUploadMarker	The upload ID marker in the response for the next request to return the remaining results. Type: string Parent nodes: ListMultipartUploadsResult
MaxUploads	The maximum number of upload tasks returned by OSS. Type: integer Parent nodes: ListMultipartUploadsResult

Element	Description
IsTruncated	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. Type: Boolean Valid values: false and true Default value: false Parent nodes: ListMultipartUploadsResult
Upload	The container that stores the information about multipart upload tasks. Type: container Child nodes: Key, UploadId, and Initiated Parent nodes: ListMultipartUploadsResult
Key	The object name in an initiated multipart upload task. Type: string Parent nodes: Upload
UploadId	The ID of the multipart upload task. Type: string Parent nodes: Upload
Initiated	The time when the multipart upload task was initiated. Type: date Parent nodes: Upload

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>
```

2.13.8. ListParts

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

Request structure

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

Request parameters

Request parameters

Parameter	Description
uploadId	The ID of the multipart upload task. Type: string Default value: null
max-parts	The maximum number of parts that can be returned by OSS. Type: integer Default value: 1000
part-number-marker	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	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 name in the response. Set the value to url. Type: string Default value: null Valid value: url

Response elements

Response elements

Element	Description

Element	Description
ListPartsResult	The container that stores the response to the ListParts request. Type: container Child nodes: Bucket, Key, UploadId, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, and Part Parent nodes: none
Bucket	The name of the bucket. Type: string Parent nodes: ListPartsResult
EncodingType	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. Type: string Parent nodes: ListPartsResult
Key	The name of the object. Type: string Parent nodes: ListPartsResult
UploadId	The ID of the upload task. Type: string Parent nodes: ListPartsResult
PartNumberMarker	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 Parent nodes: ListPartsResult
NextPartNumberMarker	The number of the part after which the next listing begins. If the response does not contain all required results, all parts whose part numbers are greater than the value of this parameter are listed. Type: integer Parent nodes: ListPartsResult
MaxParts	The maximum number of parts in this response. Type: integer Parent nodes: ListPartsResult

Element	Description
IsTruncated	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. Type: Boolean Valid values: true and false Parent nodes: ListPartsResult
Part	The container that stores part information. Type: string Child nodes: PartNumber, LastModified, ETag, and Size Parent nodes: ListPartsResult
PartNumber	The number that identifies a part. Type: integer Parent nodes: ListPartsResult and Part
LastModified	The time when the part was uploaded. Type: date Parent nodes: ListPartsResult and Part
ETag	The ETag value of the content of the uploaded part. Type: string Parent nodes: ListPartsResult and Part
Size	The size of the uploaded part. Type: integer Parent nodes: ListPartsResult and Part

Detail analysis

- 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 ETag values) of ListParts to generate the final part list of CompleteMultipart.

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>&quot;3***9&quot;</ETag>
    <Size>6291456</Size>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <LastModified>2012-02-23T07:01:12.000Z</LastModified>
    <ETag>&quot;3***9&quot;</ETag>
    <Size>6291456</Size>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <LastModified>2012-02-23T07:02:03.000Z</LastModified>
    <ETag>&quot;7***9&quot;</ETag>
    <Size>1024</Size>
  </Part>
</ListPartsResult>
```

2.14. CORS

2.14.1. 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.

2.14.2. 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.

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-fight response</MaxAgeSeconds>
  </CORSRule>
  <CORSRule>
    ...
  </CORSRule>
...
</CORSConfiguration >
```

Request elements

Request elements

Element	Description	Required
CORSRule	The container for CORS rules. A maximum of 10 rules can be configured for each bucket. Type: container Parent nodes: CORSConfiguration	Yes
AllowedOrigin	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. Type: string Parent nodes: CORSRule	Yes
AllowedMethod	The allowed method for cross-origin requests. Type: enumeration (GET, PUT, DELETE, POST, and HEAD) Parent nodes: CORSRule	Yes
AllowedHeader	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. Type: string Parent nodes: CORSRule	No
ExposeHeader	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. Exposed headers cannot contain asterisks (*). Type: string Parent nodes: CORSRule	No
MaxAgeSeconds	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. Type: integer Parent nodes: CORSRule	No
CORSConfiguration	The container that stores CORS rules for the bucket. Type: container Parent nodes: none	Yes

Element	Description	Required

Detail analysis

- By default, CORS is disabled for buckets. All cross-origin requests are not allowed.
- To use CORS for applications when you access OSS from www.a.com by using the XMLHttpRequest function of a browser, you must call the PutBucketcors operation to manually upload a CORS rule and enable CORS. This rule is described in an XML document.
- The CORS settings for each bucket are specified by multiple CORS rules. A maximum of 10 rules can be configured for each bucket. The uploaded XML document cannot be larger than 16 KB.
- When OSS receives a cross-origin request (or an OPTIONS request) for a bucket, OSS reads the CORS rules of the bucket and checks relevant permissions. OSS checks the request against each rule. When OSS finds the first match, it returns a corresponding header. If no matches are found, OSS does not include any CORS header in the response.
- The following conditions must be met before OSS determines that the request matches a CORS rule:
  - 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.
- If the Content-MD5 request header is specified, OSS calculates the Content-MD5 header value in the request body and checks whether the two values are the same. If they are different, InvalidDigest is returned.

Examples

Sample request of adding CORS rules for a bucket

```
PUT /? cors HTTP/1.1
Host: ***.regionid.example.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS q***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
```

2.14.3. 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
```

Response elements

Response elements

Element	Description
CORSRule	The container for CORS rules. A maximum of 10 rules can be configured for each bucket. Type: container Parent nodes: CORSConfiguration
AllowedOrigin	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. Type: string Parent nodes: CORSRule
AllowedMethod	The allowed method for cross-origin requests. Type: enumeration (GET, PUT, DELETE, POST, and HEAD) Parent nodes: CORSRule
AllowedHeader	Indicates 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. Type: string Parent nodes: CORSRule
ExposeHeader	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. Exposed headers cannot contain asterisks (*). Type: string Parent nodes: CORSRule
MaxAgeSeconds	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. Type: integer Parent nodes: CORSRule

Element	Description
CORSConfiguration	The container that stores CORS rules for the bucket. Type: container Parent nodes: none

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and the NoSuchBucket error code.
- Only the bucket owner can obtain CORS rules. Otherwise, OSS returns 403 Forbidden and the AccessDenied error code.
- If CORS rules do not exist, OSS returns 404 Not Found and the NoSuchCORSConfiguration error code.

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>
```

2.14.4. 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
```

Detail analysis

- If the bucket does not exist, OSS returns 404 Not Found and the NoSuchBucket error code.
- Only the owner of a bucket can delete the CORS rules configured for the bucket. If you call the DeleteBucketcors operation on a bucket that does not exist, OSS returns 403 Forbidden and the AccessDenied error code.

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
```

2.14.5. 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

Request headers

Header	Description
Origin	The origin of the request, used to identify a cross-origin request. Type: string Default value: null

Header	Description
Access-Control-Request-Method	The method to use in the cross-origin request. Type: string Default value: null
Access-Control-Request-Headers	The custom headers to be sent in the actual cross-origin request. Type: string Default value: null

Response headers

Response headers

Header	Description
Access-Control-Allow-Origin	The origin that is contained in the request. If the origin is not allowed in a request, the request does not contain the origin. Type: string
Access-Control-Allow-Methods	The HTTP method used by the request. If the method is not allowed in a request, the request does not contain the method. Type: string
Access-Control-Allow-Headers	The list of headers included in the request. If the request contains a forbidden header, this header is not contained and the request is rejected. Type: string
Access-Control-Expose-Headers	The list of header fields that are accessible to JavaScript applications on the client. Type: string
Access-Control-Max-Age	The maximum duration for the browser to cache preflight results. Unit: seconds. Type: integer

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
```

2.15. 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
BucketNotEmpty	The error message returned because the bucket is not empty.	409
EntityTooLarge	The error message returned because the entity is too large.	400

Error code	Description	HTTP status code
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
FilePartInterity	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

Error code	Description	HTTP status code
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

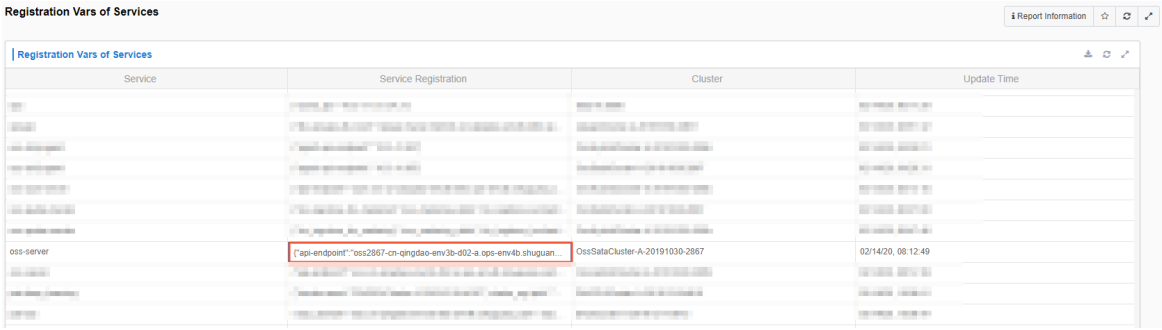
```
HTTP/1.1 400 Bad Request
Server: AliyunOSS
Date: Thu, 11 Aug 2016 01:44:54 GMT
Content-Type: application/xml
Content-Length: 322
Connection: keep-alive
x-oss-request-id: 57ABD896CCB80C366955187E
x-oss-server-time: 0
<? xml version="1.0" encoding="UTF-8"? >
<Error>
 <Code>NotImplemented</Code>
 <Message>A 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>
```

2.16. Obtain an endpoint

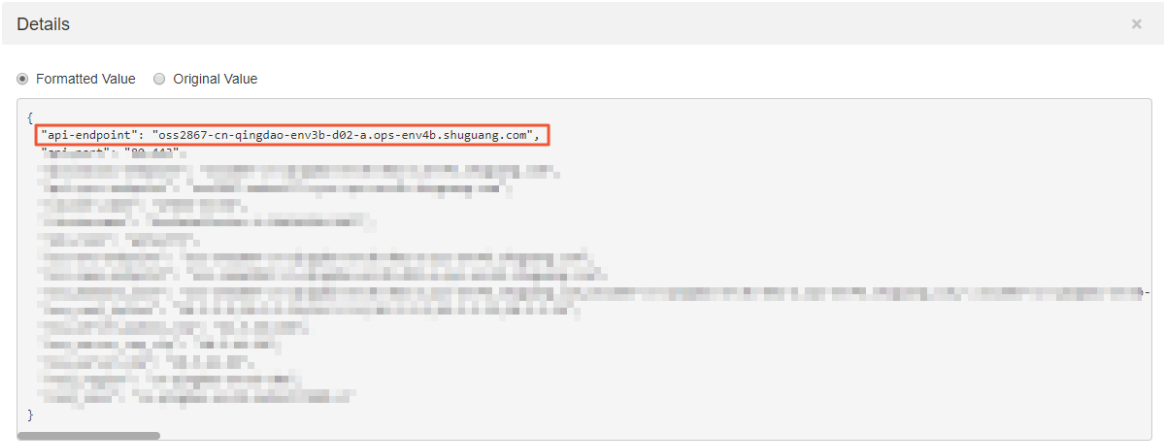
You can obtain the endpoint of OSS in the Apsara Stack Operations console.

Procedure

1. Log on to the Apsara Stack Operations console. For more information about how to log on to the console, see *Operations and Maintenance Guide*. Specifically, navigate in the guide as follows: **Operations of basic platforms > Apsara Stack Operations (ASO) > Log on to Apsara Stack Operations**.
2. In the left-side navigation pane, choose **Products > Product List > Apsara Stack O&M**.
3. Click **Apsara Infrastructure Management Framework** to go to the Apsara Infrastructure Management Framework console.
4. In the top navigation bar, choose **Reports > System Reports**.
5. On the **System Reports** page, click **Registration Vars of Services**.
6. On the **Registration Vars of Services** page, find the oss-server row, right-click the **Service Registration** column, and choose **Show More** from the shortcut menu.



7. In the **Details** dialog box that appears, find the OSS endpoint based on the key value of OSS, which is api-endpoint.



2.17. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only operations administrators and 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. Log on to the Apsara Stack Cloud Management (ASCM) console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, click the icon on the right of the organization to be added.
5. Select **AccessKey**.
6. In the message that appears, view the AccessKey pair of the organization.

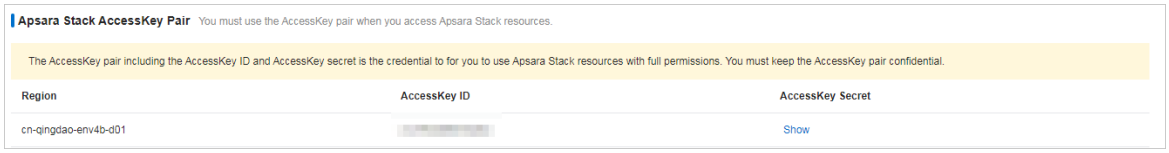
> **Note** An AccessKey pair is automatically allocated to a level-1 organization. Subordinate organizations use the same AccessKey pair as their level-1 organization.

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 ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.

3. In the **Apsara Stack AccessKey Pair** section of the User Information page, view your AccessKey pair.



> **Note** The AccessKey pair consists of an AccessKey ID and an AccessKey secret. They allow you to access Apsara Stack resources with full permissions for your account. You must keep them confidential.

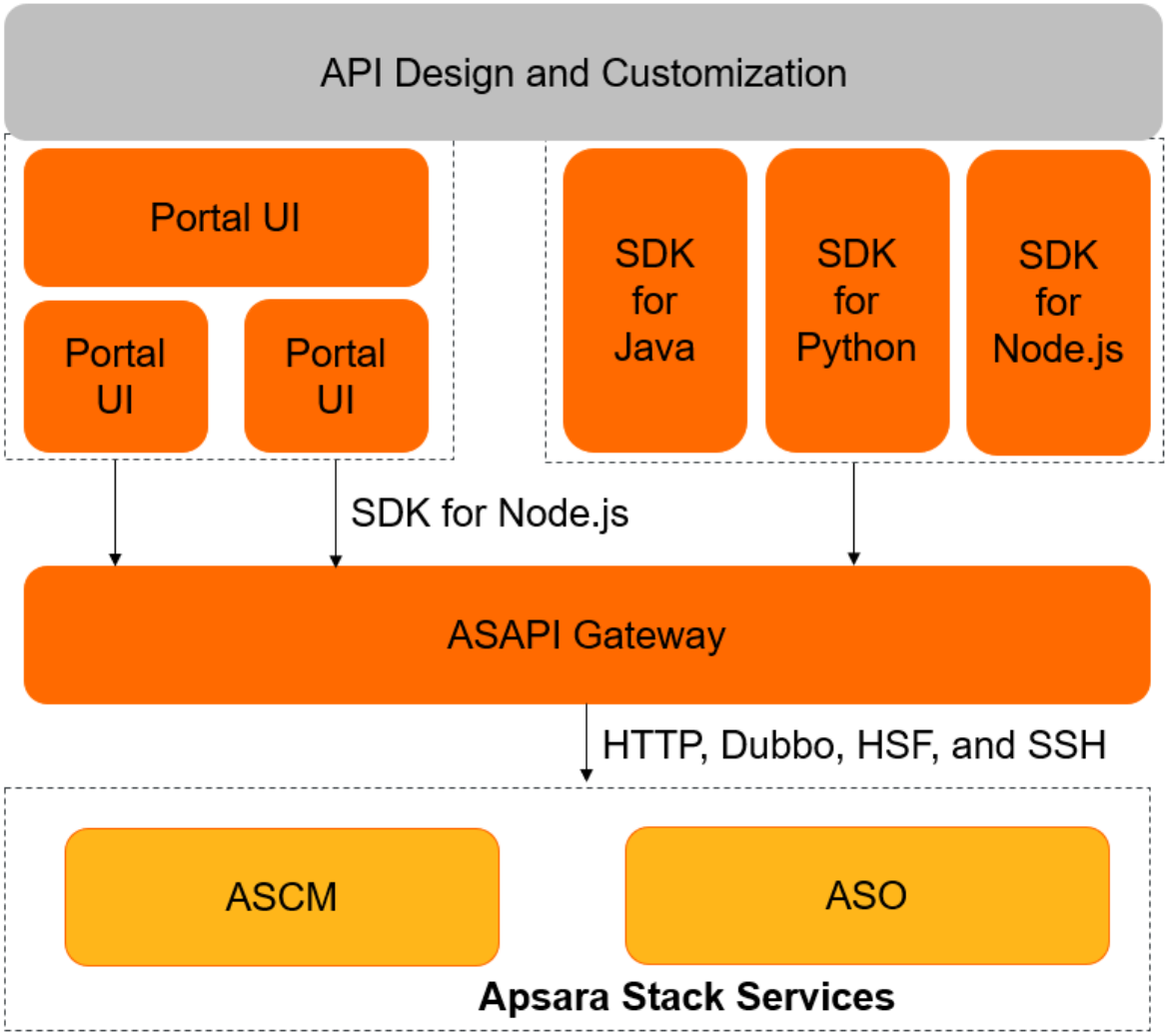
3.ApsaraDB for RDS

3.1. Use ASAPI to call Apsara Stack SDKs

3.1.1. Overview

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

ASAPI is a platform that integrates all features and is the only portal to connect to Apsara Stack. ASAPI can categorize operations and audit and manage their calls. It can also assemble and customize existing operations and adapt to your operations. Each cloud service can call ASAPI for required data and features, avoiding inconsistent operations caused by different call methods and separate encapsulation. This can ensure the operation program integrity and improve development efficiency and code portability.



ASAPI allows you to manage and call operations in a unified manner and customize development. ASAPI supports access of most cloud services deployed in Apsara Stack and passthrough or encapsulation of different backend APIs. This way, operations can be called with the same method. ASAPI allows you to use SDKs and API Portal to call APIs registered in ASAPI, customize SDKs, and manage versions of Apsara Stack, services, and APIs.

3.1.2. Log on to the API Tool console

You can log on to the API Tool console through the Apsara Stack Cloud Management (ASCM) console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password used to log on to the console from the operations administrator.

   > **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum 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. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%).

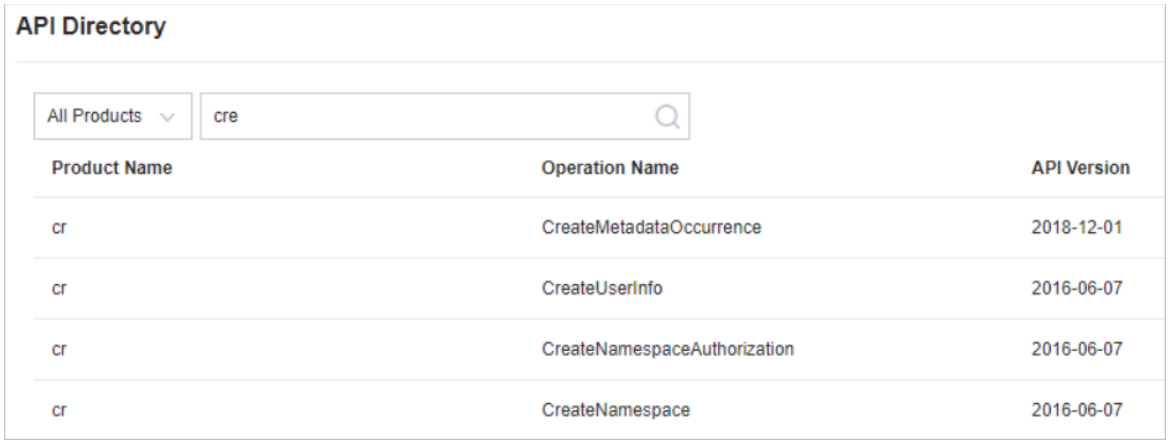
3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products** > **Others** > **API Tool**.

3.1.3. View the information about API operations

You must obtain a service name, API operation name, and API version. All the information is required when you use SDKs.

Procedure

1. Log on to the API Tool console.
2. In the left-side navigation pane, click **API fold**.
3. On the **API Directory** page, select the target service from the drop-down list, enter the API operation name in the search box, and then click the search icon.



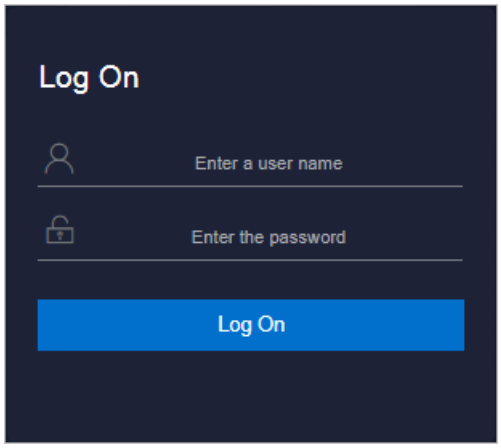
4. View and remember the service name, API operation name, and API version.

3.1.4. Obtain the endpoint of ASAPI

You can obtain the endpoint of Apsara Stack API (ASAPI) in the Apsara Stack Operations (ASO) 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.

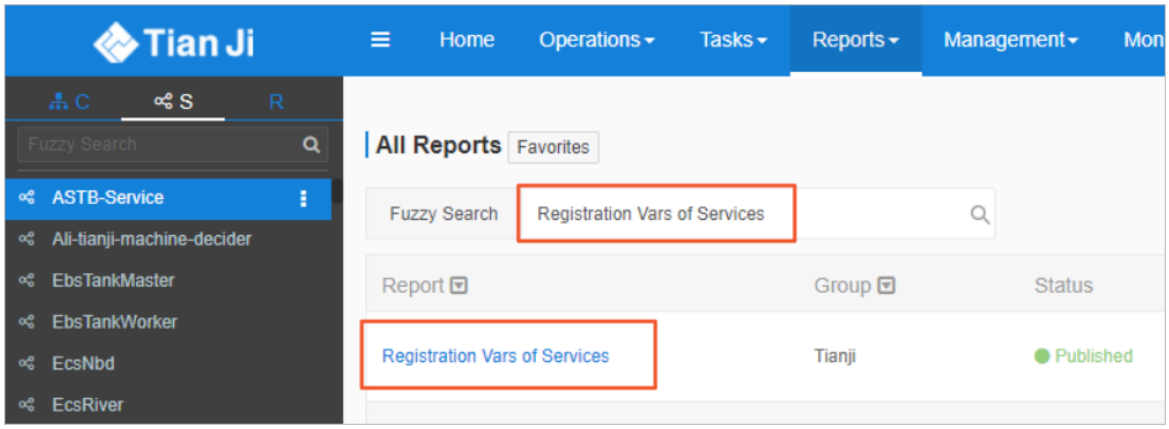
When you log on to the ASO console for the first time, you must change the password of your username as prompted.

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

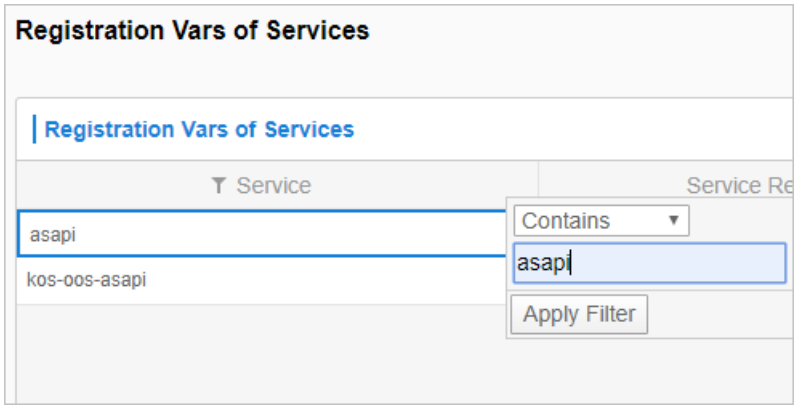
- It must contain uppercase and lowercase letters.

- 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.

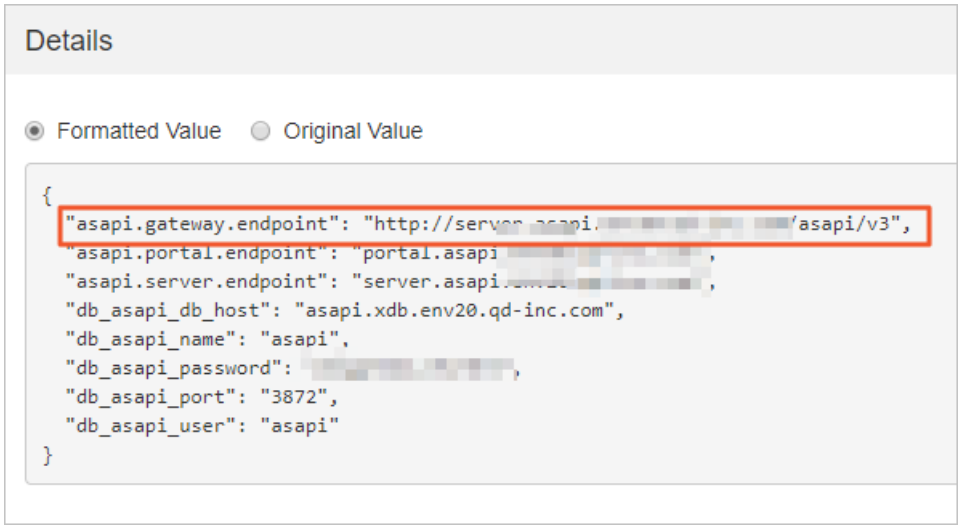
3. Click **Log On** to go to the **ASO** console.
4. In the left-side navigation pane, choose **Products** > **Product List**. On the Product List page, click **Apsara Infrastructure Management Framework**.
5. In the left-side navigation pane of the Apsara Infrastructure Management Framework console, click **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 corresponding to asapi and select **Show More**.
9. In the **Details** dialog box, view the value of **asapi.gateway.endpoint**. The value is the logon URL of the Apsara ASAPI Management platform.



3.1.5. Obtain the core package for ASAPI

Apsara Stack uses Asapi as the unified portal for all Apsara Stack products that are connected to Asapi. Asapi also provides SDKs to call product APIs.

To obtain the SDK package for Asapi, see Apsara Stack documentation.

3.1.6. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only operations administrators and 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:

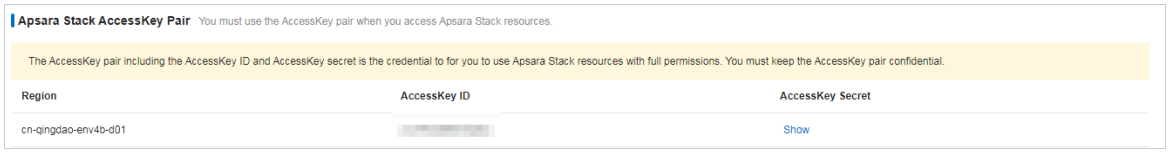
1. Log on to the Apsara Stack Cloud Management (ASCM) console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, click the icon on the right of the organization to be added.
5. Select **AccessKey**.
6. In the message that appears, view the AccessKey pair of the organization.

> **Note** An AccessKey pair is automatically allocated to a level-1 organization. Subordinate organizations use the same AccessKey pair as their level-1 organization.

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 ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section of the User Information page, view your AccessKey pair.



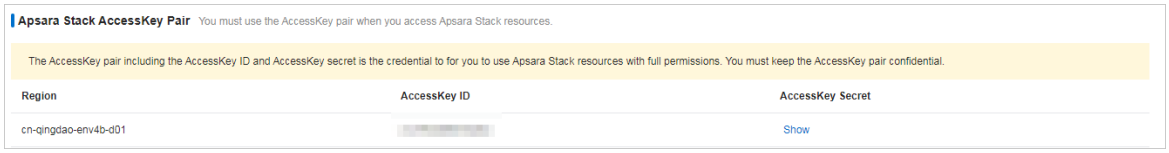
> **Note** The AccessKey pair consists of an AccessKey ID and an AccessKey secret. They allow you to access Apsara Stack resources with full permissions for your account. You must keep them confidential.

3.1.7. Query the region ID

You must specify the region ID when you call Apsara Stack SDK for Node.js. You must query the region ID in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section, you can view the region ID.



3.1.8. Call SDK for Java

This topic describes how to obtain and use Apsara Stack SDK for Java.

Prerequisites

- An authorized account and an AccessKey pair (AccessKey ID and AccessKey secret) are obtained. For more information about how to obtain and view your AccessKey pair in the ASCM console, see Obtain an AccessKey pair.
- The endpoint of ASAPI is obtained. For more information about how to obtain the endpoint of ASAPI, see Obtain the endpoint of Asapi.
- Java Development Kit (JDK) 1.6 or later is obtained.

Add dependencies

Add dependencies to pom.xml:

```
<dependency>
   <groupId>com.aliyun</groupId>
   <artifactId>asapi</artifactId>
   <version>0.0.1-SNAPSHOT</version>
   <scope>system</scope>
   <systemPath>${project.basedir}/lib/aliyun-java-sdk-asapi.jar</systemPath>
</dependency>
```

Use Apsara Stack SDK for Java

You can use GET and POST to call ASAPI API operations.

- GET requests

```
 ASClient client = new ASClient();

// Form 1
String result = client.doGet(endpoint, parameters);

// Form 2
String result = client.doRequest(endpoint, parameters);
```

- POST requests

```
// Set the endpoint of ASAPI.
String endpoint = "http://xxx/asapi/v3";

// Set request parameters.
Map<String, String> parameters = new HashMap<String, String>();

// Set parameters for the product.
parameters.put("Product", "rds");          // The product name must be the same as the name displayed in the corresponding API directory.
parameters.put("Action", "CreateDBInstance");        // The API name must be in the same case as that in the API directory.
parameters.put("Version", "2019-05-10");     // The API version.

// Set authorization-related parameters.
parameters.put("AccessKeyId", "<your-access-key-id>");        // The AccessKey ID of the authorized account.
parameters.put("AccessKeySecret", "<your-access-key-secret>");  // The AccessKey secret of the authorized account.

// Set business parameters.
parameters.put("username", "<your-username>");               // The username used to log on to the ASCM console.
parameters.put("password", "<your-password>");               // The password used to log on to the ASCM console.
JSONObject jsonObject = new JSONObject();
jsonObject.put("param1", "json parameter 1");
jsonObject.put("param2", "json parameter 2");
parameters.put("json", jsonObject.toJSONString());

// Create an ASClient object for initiating requests.
ASClient client = new ASClient();
// Add a request header.
client.addHeader("x-acs-resourceGroupId", "112233");
// Initiate the request and obtain the response.
String result = client.doPost(endpoint, parameters);
```

3.1.9. Call SDK for Node.js

This topic describes how to obtain and use Apsara Stack SDK for Node.js.

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 Apsara Stack API (ASAPI) is obtained. For more information, see Obtain the endpoint of Asapi.
- The region ID is obtained. For more information about how to query the region ID, see Query the region ID.

Add dependencies

1. Install Apsara Stack SDK for Node.js. `tnpm install --save @ali/asapi-nodejs-sdk`
2. Add the following configurations to *mocha.opts* if *mocha.opts* depends on eggjs:

```
--recursive
--colors
--timeout 60000
--reporter mochawesome
--require co-mocha
--require egg-ts-helper/register
--require espower-typescript/guess
--require intelli-espower-loader
```

Use SDK for Node.js

The following example shows how to call Apsara Stack SDK for Node.js:

1. Create a client instance. To create a client instance, you must set the region ID, endpoint, AccessKey ID, and AccessKey secret in the *config.ts* file. Example:

```
local: {
  regionId: 'obtained region ID',
  API_GATEWAY: 'obtained endpoint',
  accessKeyId: 'your AccessKey ID',
  accessKeySecret: 'your AccessKey secret',
},
```

2. Create an API request and configure parameters.
3. Initiate the request and handle any responses or exceptions.

```
// Specify the ASAPI endpoint.
const API_GATEWAY = "http://xxxx/asapi/v3"

// Create the rpc client.
const asapiClient = ctx.service.sdk.rpcClient;

// Set request parameters.
let params = {
   Action: 'CreateDBInstance',
   Product: 'rds',
   regionId: 'cn-qingdao-env12-d01',
   accessKeyId: 'xxxxxxx',
   accessKeySecret: 'xxxxxxx',
   Version: '2018-01-22',
};

// Set parameters in the request header.
asapiClient.addHeader("x-acs-regionId", "cn-hangzhou");

// Initiate the request and obtain the response.
const result = await asapiClient.doRequest(API_GATEWAY, params, {});
```

3.1.10. (Optional) Use POP SDK to call ASAPI

This topic describes how to switch the gateway to ASAPI if an application uses POP SDK to call cloud products.

Procedure

1. Switch the gateway to ASAPI and add common parameters required by the API operation to the request header, so that the API operation can be called in a transparent manner.
2. Add the **Product Name** field to the request header.

> **Note** Each product uses a unique endpoint in POP SDK. However, ASAPI provides only one domain name. You cannot identify the product to which an API operation belongs based on the API operation name and the version number. For example, both ECS and KVStore for Redis provide the DescribeInstances API operation of the 2015-01-01 version.

Sample code

An API operation of ECS is called in the following example:

```
try {
    CreateDBInstanceResponse 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());
}
```

3.2. API reference

3.2.1. List of operations by function

The following tables list API operations available for use in RDS.

RDS instance management

Operation	Description
CreateDBInstance	Creates an RDS Instance.
DeleteDBInstance	Releases an RDS instance.
RestartDBInstance	Restarts an RDS instance.
DescribeDBInstanceAttribute	Views the details of an RDS instance.
DescribeDBInstances	Queries the RDS instance list.
ModifyDBInstanceSpec	Modifies the specifications of an RDS instance.
DescribeDBInstanceNetInfo	Views all connection strings.
DescribeDBInstanceIPArrayList	Queries the IP whitelist of an RDS instance.
ModifySecurityIps	Modifies the IP whitelist of an RDS instance.
DescribeRegions	Queries the RDS region and zone information.
DescribeDBInstanceHAConfig	Queries the RDS zone information and data replication status.
PurgeDBInstanceLog	Clears the logs of an RDS instance.
ModifyDBInstanceDescription	Modifies the description of an RDS instance.

Operation	Description
ModifyDBInstanceMaintainTime	Modifies the maintenance window of an RDS instance.
ModifyDBInstanceHAConfig	Modifies the data replication and high availability polices of an RDS instance.
SwitchDBInstanceHA	Switches between the primary and secondary RDS instances.
CreateReadonlyDBInstance	Creates a read-only RDS instance.
DescribeDBInstanceSSL	Queries the SSL link of an RDS instance.
ModifyDBInstanceSSL	Modifies the SSL link of an RDS instance.

RDS database management

Operation	Description
CreateDatabase	Creates a database.
DeleteDatabase	Deletes a database.
DescribeDatabases	Queries the database list.
ModifyDBDescription	Modifies the description of a database.

RDS account management

Operation	Description
CreateAccount	Creates an account.
DeleteAccount	Deletes an account.
DescribeAccounts	Queries the account list.
GrantAccountPrivilege	Grants permissions to an account.
RevokeAccountPrivilege	Revokes permissions from an account.
ModifyAccountDescription	Modifies the description of an account.
ResetAccountPassword	Resets the password of an account.
ResetAccount	Resets an account.

RDS backup and restoration

Operation	Description
CreateBackup	Creates a backup.
CloneDBInstance	Clones an RDS instance.
DescribeBackups	Queries the list of backups.
DescribeBackupPolicy	Queries a backup policy.
ModifyBackupPolicy	Modifies a backup policy.
RestoreDBInstance	Restores a backup set to an instance.
DeleteBackup	Deletes data backup files.

RDS monitoring management

Operation	Description
DescribeResourceUsage	Queries the resource usage of an instance.
DescribeDBInstancePerformance	Queries the performance data of an instance.
DescribeDBInstanceMonitor	Queries the monitoring behaviors of an instance.
ModifyDBInstanceMonitor	Modifies the monitoring behaviors of an instance.

RDS parameter management

Operation	Description
DescribeParameterTemplates	Queries a database parameter template.
DescribeParameters	Queries the running database parameter list of the current instance.
ModifyParameter	Modifies the database parameter list.

RDS log management

Operation	Description
DescribeSlowLogs	Queries the slow log list.
DescribeSlowLogRecords	Queries slow log details.
DescribeErrorLogs	Queries error logs.
DescribeBinlogFiles	Queries binlogs.

Operation	Description
DescribeSQLCollectorPolicy	Queries the status of SQL collection.
ModifySQLCollectorPolicy	Changes the status of SQL collection.
DescribeSQLLogRecords	Views SQL log details.
DescribeSQLLogFiles	Queries the SQL audit file list.

RDS data migration

Operation	Description
ImportDatabaseBetweenInstances	Migrates data from other instances.
CancelImport	Cancels an import task.

RDS performance optimization

Operation	Description
DescribeSQLLogReports	Queries the reports of SQL logs.

3.2.2. Request method

3.2.2.1. Request structure

This topic describes the request structure of RDS APIs.

Endpoints

The endpoint of the RDS API is rds.example.com. For more information about how to obtain a service endpoint, see Obtain an endpoint.

Protocols

You can send requests over HTTP or HTTPS. We recommend that you send requests over HTTPS to enhance security.

Methods

The HTTP GET method can be used to send requests. Request parameters must be included in the request URL.

Request parameters

Each request must specify the action to be performed in the Action parameter. Each operation request must include common request parameters and the specific request parameters of the operation called.

Encoding

Requests and responses are encoded in UTF-8.

3.2.2.2. Common parameters

3.2.2.2.1. Common request parameters

Common request parameters must be included in the request of each API.

Parameter	Type	Required	Description
Format	String	No	The format in which to return the response. Valid values: JSON and XML. Default value: XML.
Version	String	Yes	The version number of the API. It is in the YYYY-MM-DD format. The current version is 2014-08-15.
AccessKeyId	String	Yes	The AccessKey ID provided to you by Alibaba Cloud.
Signature	String	Yes	The signature string of the current request. For more information about how signatures are calculated, see Request signatures.
SignatureMethod	String	Yes	The encryption method of the signature string. Set the value to HMAC-SHA1.
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. For example, 20:00:00, August 15, 2013, UTC+8 will be written as 2013-08-15T12:00:00Z.
SignatureVersion	String	Yes	The version of the signature encryption algorithm. Set the value to 1.0.
SignatureNonce	String	Yes	A unique, random number used to prevent replay attacks. You must use different numbers for different requests.

Example

```
https://rds.example.com/
? Format=xml
&Version=2014-08-15
&Signature=Pc5xxxxxxxxxxxxxxxxxxxxxx
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=key-test
&OwnerId=12345678
&Timestamp=2014-10-10T12:00:00Z
```

3.2.2.2.2. Common response parameters

The system returns a unique RequestId for each API request, regardless of whether the API operation is successful.

Examples

```
<? xml version="1.0" encoding="utf-8"? >
<!--Result root node-->
<API operation name+response>
   <!-Returned request tag-->
   <RequestId>4C4***38-****-****-****-*******6F216</RequestId>
   <!--Responses-->
</API operation name+response>
```

3.2.2.3. Response

3.2.2.3.1. Success responses

The response data for API requests is displayed in a uniform format. In the response data, a 2xx HTTP status code indicates that the API operation is called.

The response data for called API operations is returned in either the XML or JSON format. You can add parameters to an API request to specify the format of the response data. Response data uses the XML format by default.

Sample responses in this topic are formatted with line breaks or indents for a clear view. The actual responses are not formatted in the same manner as these examples.

- Responses in the XML format include information about whether the call is successful, and details specific to the operation. Example:

```
<!—Result root node-->
<API name + response>
<!—Response request tag-->
<RequestId>4C467B38-****-447D-****-AC04*******16</RequestId>
<!—Response data-->
</API name + response>
```

- JSON format

```
{ "RequestId": "4C467B38-****-447D-****-AC04*******16", /* Response Data */}
```

3.2.2.3.2. Sample error responses

If an error occurs when you call an API operation, no result is returned. You can locate the cause of the error by referencing Error codes in Appendix.

When an error occurs on an HTTP request for an API operation, a 4xx or 5xx HTTP status code is returned.

The returned body includes the specific error code and message.

The response body also contains the RequestId and HostId parameters. The RequestId parameter indicates the globally unique identifier (GUID) of the request, and the HostId parameter indicates the ID of the site that you accessed with the request. You can contact Alibaba Cloud Customer Services when the cause of the error cannot be found. We recommend that you provide HostId and RequestId to help us resolve the issue at the earliest opportunity.

XML format

```
<Error>
<RequestId>8906582E-****-****-A6C4-0E78*******5</RequestId>
<HostId>rds.example.com</HostId>
<Code>UnsupportedOperation</Code>
<Message>The specified action is not supported. </Message>
</Error>
```

JSON format

```
{ "RequestId": "7463B73D-****-4D19-****-6B8*******2EF",
"HostId": "rds.example.com",
"Code": "UnsupportedOperation",
"Message": "The specified action is not supported."
}
```

3.2.2.4. Request signatures

ApsaraDB for RDS authenticates each access request. Therefore, a request must contain signature information regardless of whether it is sent over HTTP or HTTPS. For information about how to obtain an AccessKey pair (AccessKey ID and AccessKey secret), see Obtain an AccessKey pair.

ApsaraDB for RDS implements symmetric encryption with an AccessKey pair to verify the identity of the request sender.

The AccessKey pair is issued by Alibaba Cloud. You can apply for AccessKey pairs and manage them on the official Alibaba Cloud website.

- The AccessKey ID is used to verify the identity of the user.
- The AccessKey secret is used to encrypt and verify the signature string on the server. You must keep your AccessKey secret strictly confidential.

Perform the following steps to sign an access request:

1. Use request parameters to construct a canonicalized query string.
   i. Sort all of the request parameters in alphabetical order.

      The request parameters include all common and operation-specific parameters except Signature.

      > Note If you use the GET method to send a request, the request parameters are included as part of the request URI. The request parameters in the URI are placed after a question mark (?) and separated by ampersands (&).

   ii. Encode the name and value of each request parameter.

      Parameter names and values must be encoded into a URL based on the UTF-8 character set.

      The following rules apply in the URL encoding process:

      - Uppercase letters, lowercase letters, digits, and some special characters such as hyphens (-), underscores (_), periods (.), and tildes (~) do not need to be encoded.
      - Other characters must be percent encoded in %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 (+).

        > Note Most libraries that support URL encoding, such as java.net.URLEncoder, comply with the Multipurpose Internet Mail Extensions (MIME) encoding rules of application/x-www-form-urlencoded.

        If this encoding method is used, replace the plus signs (+) in the encoded strings with %20, the asterisks (*) with %2A, and %7E with a tilde (~) to conform to the encoding rules.

   iii. Connect the encoded parameter names and values by using equal signs (=).

   iv. Sort the equal sign-connected strings by their parameter names in alphabetical order and connect them with ampersands (&).

2. Create a string-to-sign from the encoded canonicalized query string based on the following rules:

```
StringToSign=
HTTPMethod + "&" +
percentEncode("/") + "&" +
percentEncode(CanonicalizedQueryString)
```

   - HTTPMethod: specifies the HTTP method used to make the request, such as GET.
   - percentEncode("/"): specifies the encoded value for the forward slash (/) based on the URL encoding rules described in 1.b. The value is %2F.
   - percentEncode(CanonicalizedQueryString): specifies the string encoded from the canonicalized query string constructed in Step 1 based on the URL encoding rules described in 1.b.
3. Calculate the HMAC value of the string-to-sign by using the AccessKey secret as the key.

   > **Note** An ampersand (&) is added to the end of the key (ASCII: 38). The hashing algorithm used is SHA-1.

4. Encode the HMAC value in Base64 to obtain the signature string.
5. Add the signature string to the request as the Signature parameter.

   > **Note** When the signature string is submitted to the RDS server as the final request parameter value, the string must be URL-encoded like other parameters based on rules defined in RFC 3986.

   The following code is the request URL without a signature for the DescribeDBInstances operation:

```
http://rds.example.com/?TimeStamp=2013-06-01T10:33:56Z&Format=XML&AccessKeyId=testid&Action=DescribeDBInstances&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2014-08-15&SignatureVersion=1.0
```

   The following code is the string-to-sign:

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DDescribeDBInstances%26Format%3DXML%26RegionId%3Dregion1%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3DNwDAxvLU6tFE0DVb%26SignatureVersion%3D1.0%26TimeStamp%3D2013-06-01T10%253A33%253A56Z%26Version%3D2014-08-15
```

   Assume that the AccessKey ID is testid, the AccessKey secret is testsecret, and the key used for HMAC calculation is testsecret&. The following signature string is obtained:

```
BIPOMlu8LXBeZtLQkJTw6iFvw1E=
```

   The following code is the signed request URL that has the Signature parameter added:

```
http://rds.example.com/?TimeStamp=2013-06-01T10%3A33%3A56Z&Format=XML&AccessKeyId=testid&Action=DescribeDBInstances&SignatureMethod=HMAC-SH
```

3.2.3. Instance management

3.2.3.1. CreateDBInstance

Description

You can call this operation to create an RDS instance. For information about the instance type, see Instance types in Appendix.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateDBInstance.
RegionId	String	Yes	The region ID of the instance. It can be up to 50 characters in length. You can call the DescribeRegions operation to query the most recent region list.
ZoneId	String	No	The zone ID of the instance. You can call the DescribeRegions operation to query the most recent zone list.
Engine	String	Yes	The engine of the database. Set the value to MySQL.
EngineVersion	String	Yes	The version of the database engine. Valid value: MySQL 5.6.
DBInstanceClass	String	Yes	The type of the instance. For more information, see Instance types in Appendix.
DBInstanceStorage	Integer	Yes	The storage space of the instance. This value must be in 5 GB increments. Unit: GB. For more information, see Appendix.
DBInstanceNetType	String	Yes	The network connection type of the instance. Valid values: Internet and Intranet.
DBInstanceDescription	String	No	The description of the instance. It cannot start with http:// or https://. It must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-).

Parameter	Type	Required	Description
SecurityIPList	String	Yes	The list of IP addresses that are allowed to access all databases in the instance. You must separate different IP addresses by using commas (,) and ensure that there are no duplicates. You can add a maximum of 1,000 IP addresses. Format: %, 0.0.0.0/0, 10.23.12.24 (IP address), or 10.23.12.24/24 (CIDR block, where /24 indicates the prefix of the CIDR block is 24-bit. You can replace 24 with a value ranging from 1 to 32.). 0.0.0.0/0 indicates that the instance can be accessed by all devices.
ClientToken	String	Yes	The client token that is used to ensure the idempotence of the request.
InstanceNetworkType	String	No	The network type of the instance. Valid value: Classic.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
DBInstanceId	String	The ID of the instance.
OrderId	String	The ID of the order.
ConnectionString	String	The endpoint used to connect to the database.
Port	String	The port used to connect to the database.

Examples

- XML format

```
<CreateDBInstanceResponse>
    <OrderId>100789370230206</OrderId>
    <ConnectionString>***************.mysql.rds.aliyuncs.com </ConnectionString>
    <DBInstanceId>rdsaiiabnaiiabn</DBInstanceId>
    <port>3306</port>
    <RequestId>1E*****E0-****-43DA-****-EAF2*****4DC</RequestId>
</CreateDBInstanceResponse>
```

- JSON format

```
{
    "OrderId": "100789370230206",
    "ConnectionString": "rd***************.mysql.rds.aliyuncs.com",
    "DBInstanceId": "rds**********bn",
    "Port": "3306",
    "RequestId": "1E4*****0-****-43DA-****-EAF******4DC"
}
```

```
{"availableZoneInfoList":[
{"zoneID":"cn-shanghai-b",
"vSwitchID":"vsw-uf6cl821gkhsikw2fk7cq",
"ipAddress":"192.168.0.1","role":"master"},
{"zoneID":"cn-shanghai-b",
"vSwitchID":"vsw-uf6cl821gkhsikw2fk7cq",
"ipAddress":"192.168.1.1",
"role":"slave"}
]}
```

3.2.3.2. DeleteDBInstance

Description

You can call this operation to release an RDS instance. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is not manually locked.
- The instance is a pay-as-you-go primary instance, read-only instance, disaster recovery instance, or temporary instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DeleteDBInstance.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=DeleteDBInstance
&DBInstanceId=rds**********abn
&<[Common request parameters]>
```

Sample success responses

- XML format

```
<DeleteDBInstanceResponse>
      <RequestId>65****32-****-4122-****-B38*****EE64</RequestId>
</DeleteDBInstanceResponse>
```

- JSON format

```
{
  "RequestId": " 65****32-****-4122-****-B38*****EEE64"
}
```

3.2.3.3. RestartDBInstance

Description

You can call this operation to restart an RDS instance. The instance will be restarted in 10 seconds.

If there are a large number of transactions that must be submitted or rolled back, the restart process may be delayed for a minute.

Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is not performing a backup task.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to RestartDBInstance.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.

Examples

- XML format

```
<RestartDBInstanceResponse>
      <RequestId>65B****2-****-4122-****-B38*****EE64</RequestId>
</RestartDBInstanceResponse>
```

- JSON format

```
{
   "RequestId": " 65B*****2-****-4122-****-B38*****EE64"
}
```

3.2.3.4. DescribeDBInstanceAttribute

Description

You can call this operation to query the details of a specified RDS instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstanceAttribute.
DBInstanceId	String	Yes	The ID of the instance to be queried. Separate multiple IDs with commas (,). Up to 30 IDs can be entered at a time.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
Items	List<DBInstanceAttribute>	None

DBInstanceAttribute parameters

Parameter	Type	Description
DBInstanceId	String	The ID of the instance.
DBInstanceType	String	The role of the instance. Valid values: • Primary: primary instance • Readonly: read-only instance • Guard: disaster recovery instance • Temp: temporary instance
InstanceNetworkType	String	The network type of the instance. Valid values: • Classic • VPC
RegionId	String	The ID of the region.
ZoneId	String	The ID of the zone.
ConnectionString	String	The endpoint of the instance.
Port	String	The port number to connect to the instance.
Engine	String	The engine of the database.
EngineVersion	String	The version of the database engine.
DBInstanceClass	String	The type of the instance.
DBInstanceMemory	Long	The memory capacity of the instance. Unit: MB.
DBInstanceStorage	Integer	The disk space of the instance. Unit: GB.
DBInstanceNetType	String	The network connection type of the instance. Valid values: • Internet: public network • Intranet: internal network
DBInstanceStatus	String	The status of the instance. For more information, see Instance states in Appendix.

Parameter	Type	Description
DBInstanceDescription	String	The description of the instance.
LockMode	String	The lock status of the instance. Valid values: • Unlock: normal • ManualLock: locked manually • LockByExpiration: locked automatically upon expiration • LockByRestoration: locked automatically before instance rollback • LockByDiskQuota: locked automatically due to full disk space
LockReason	String	The reason why the instance is locked.
DBMaxQuantity	Integer	The maximum number of databases that can be created in an instance.
AccountMaxQuantity	Integer	The maximum number of accounts that can be created.
CreationTime	String	The time when the instance was created. The time is displayed in the YYYY-MM-DDThh:mm:ssZ format. Example: 2011-05-30T12:11:4Z
MaintainTime	String	The maintenance window of the instance. Example: 00:00Z-02:00Z indicates that routine maintenance can be performed from 00:00 to 02:00.
AvailabilityValue	String	Indicates the availability status of the instance in the current year. Unit: %
MaxIOPS	Integer	The maximum number of input/output operations per second (IOPS) supported by the instance.
MaxConnections	Integer	The maximum number of concurrent connections to an instance.

Parameter	Type	Description
MasterInstanceId	String	The ID of the primary instance. If this parameter is not returned (null), the instance is a primary instance.
IncrementSourceDBInstanceId	String	The instance ID of the incremental data source. For example, the incremental data source of a disaster recovery instance or read-only instance is a primary instance, and the incremental data source of the primary instance is null.
GuardDBInstanceId	String	The ID of the disaster recovery instance if the instance has a disaster recovery instance.
TempDBInstanceId	String	The ID of the temporary instance if the instance has a temporary instance.
ReadOnlyDBInstanceIds	List<DBInstanceId>	The list of IDs of read-only instances attached to the primary instance.
SecurityIPList	String	This API operation is discontinued. For more information, see DescribeDBInstanceIPArrayList.

DBInstanceAttribute parameters

> **Note** You can call the DescribeDBInstanceHAConfig operation to query the primary and secondary node information of the instance.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeDBInstanceAttribute
&DBInstanceId=rds*********abn
&<Common request parameters>
```

Sample success responses

JSON format

```
{
  "Items": {
    "DBInstanceAttribute": [
      {
        "LockMode": "Unlock",
        "ConnectionString": "rds**********.mysql.rds.aliyuncs.com",
        "CreationTime": "2014-10-10T10:28:48Z",
        "DBInstanceNetType": "Internet",
```

```
      "ReadOnlyDBInstanceIds": {
        "ReadOnlyDBInstanceId": []
      },
      "MaxConnections": 60,
      "LockReason": "",
      "Engine": "MySQL",
      "AvailabilityValue": "100.0%",
      "AccountMaxQuantity": 50,
      "DBMaxQuantity": 200,
      "RegionId": "cn-hangzhou",
      "TempDBInstanceId": "",
      "DBInstanceMemory": 240,
      "MaxIOPS": 150,
      "DBInstanceType": "Primary",
      "DBInstanceStatus": "Running",
      "MasterInstanceId": "",
      "EngineVersion": "5.6",
      "IncrementSourceDBInstanceId": "",
      "GuardDBInstanceId": "",
      "DBInstanceStorage": 10,
      "DBInstanceDescription": "",
      "DBInstanceId": "rm-bpxxxxxxx",
      "PayType": "Postpaid",
      "ExpireTime": "",
      "MaintainTime": "",
      "DBInstanceClass": "rds.mys2.small",
      "SecurityIPList": "11.11.11.11",
      "Port": "3306"
    }
   ]
  },
  "RequestId": "27B****A-****-4162-****-FC0******1C1"
}
```


Sample error responses

JSON format

```
{
  "Code":"UnsupportedOperation",
  "HostId":"rds.aliyuncs.com",
  "Message":"The specified action is not supported.",
  "RequestId":"7463B73D-35CC-4D19-A010-6B8D65D242EF"
}
```

3.2.3.5. DescribeDBInstances

Description

You can call this operation to query instances or RAM-authorized instances.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDBInstances	The operation that you want to perform. Set the value to DescribeDBInstances.
RegionId	String	Yes	cn-hangzhou	The region ID.
ConnectionMode	String	No	Performance	The access mode of the instance. Valid values: • Performance: standard access mode • Safty: safe access mode • If no value is specified, instances in both access modes are returned.

Parameter	Type	Required	Example	Description
DBInstanceType	String	No	Primary	The role of the instance. Valid values: • Primary: primary instance • Readonly: read-only instance • Guard: disaster recovery instance • Temp: temporary instance • If no value is specified, instances of all roles are returned.
InstanceNetworkType	String	No	Classic	The network type of the instance. Valid values: • VPC • Classic • If no value is specified, instances of both network types are returned.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1. Valid values: any non-zero positive integer. Default value: 1.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	30	The number of entries to return on each page. Valid values: 30, 50, and 100. Default value: 30.
Tags	String	No	{"key1":"value1"}	The tag of the instance. Specify the parameter in the JSON format. The tag includes a key-value pair of TagKey and TagValue. You must specify TagKey, and you can specify TagValue as needed. Example: {"key1":"value1"}.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
TotalRecordCount	Integer	1	The total number of records that were returned.
PageRecordCount	Integer	1	The number of instances on the current page.
Items			The array that consists of instance data.
└DBInstanceId	String	rm-bp1842vmucoa5w874	The ID of the instance.
└DBInstanceDescription	String	testforRemarks	The description of the instance.

Parameter	Type	Example	Description
└PayType	String	Postpaid	The billing method of the instance. Valid values: • Postpaid: pay-as-you-go • Prepaid: subscription
└DBInstanceType	String	Primary	The role of the instance. Valid values: • Primary: primary instance • ReadOnly: read-only instance • Guard: disaster recovery instance • Temp: temporary instance
└InstanceNetworkType	String	Classic	The network type of the instance. Valid values: • Classic • VPC
└ConnectionMode	String	Performance	The access mode of the instance. Valid values: • Performance: standard access mode • Safty: safe access mode
└RegionId	String	cn-hangzhou	The region ID of the instance.
└ExpireTime	String	None	The expiration time of the instance. This parameter is not returned for pay-as-you-go instances.
└DBInstanceStatus	String	Intranet	The status of the instance. For more information, see Instance states in Appendix.

Parameter	Type	Example	Description
└Engine	String	MySQL	The engine of the database.
└DBInstanceNetType	String	Internet	The network connection type of the instance. Valid values: • Internet: public network • Intranet: internal network
└LockMode	String	Unlock	The lock status of the instance. Valid values: • Unlock: normal • ManualLock: locked manually • LockByExpiration: locked automatically upon expiration • LockByRestoration: locked automatically before instance rollback • LockByDiskQuota: locked automatically due to full disk space
└LockReason	String	None	The reason why the instance is locked.
└MasterInstanceId	String	null	The ID of the primary instance. If this parameter is not returned (null), the instance is a primary instance.
└GuardDBInstanceId	String	None	The ID of the disaster recovery instance attached to the instance.
└TempDBInstanceId	String	None	The ID of the temporary instance attached to the instance.

Parameter	Type	Example	Description
└VpcCloudInstanceId	String	vpc-23rsxdfxxxxxxxxxxxxx	The ID of the VPC.
└ReadOnlyDBInstanceIds			None
└DBInstanceId	String	rm-bp1xxxxxxxx874	The ID of the instance.
RequestId	String	1AD222E9-E606-4A42-BF6D-8A4442913CEF	The ID of the request.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeDBInstances
&RegionId=cn-hangzhou
&Engine=MySQL
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeDBInstancesResponse>
 <PageRecordCount>2</PageRecordCount>
 <Items>
  <DBInstance>
   <DBInstanceDescription>testforRemarks</DBInstanceDescription>
   <ExpireTime>2014-10-10T16:00:00Z</ExpireTime>
   <DBInstanceId>rdsxxxxxxxxx</DBInstanceId>
   <DBInstanceNetType>Internet</DBInstanceNetType>
   <PayType>Prepaid</PayType>
   <DBInstanceStatus>Running</DBInstanceStatus>
   <DBInstanceType>Primary</DBInstanceType>
   <Engine>MySQL</Engine>
   <LockMode>Unlock</LockMode>
   <LockReason/>
   <RegionId>cn-hangzhou</RegionId>
   <ZoneId>cn-hangzhou-a</ZoneId>
   <MasterInstanceId/>
   <GuardDBInstanceId/>
```

```
    <TempDBInstanceId/>
    <ReadOnlyDBInstanceIds>
     <ReadOnlyDBInstanceId/>
    </ReadOnlyDBInstanceIds>
   </DBInstance>
   <DBInstance>
    <DBInstanceDescription>testforcreate</DBInstanceDescription>
    <ExpireTime/>
    <DBInstanceId>rdsxxxxxxxxx</DBInstanceId>
    <DBInstanceNetType>Intranet</DBInstanceNetType>
    <PayType>Postpaid</PayType>
    <DBInstanceStatus>Running</DBInstanceStatus>
    <DBInstanceType>Primary</DBInstanceType>
    <Engine>MySQL</Engine>
    <LockMode>Unlock</LockMode>
    <LockReason/>
    <RegionId>cn-hangzhou</RegionId>
    <MasterInstanceId/>
    <GuardDBInstanceId/>
    <TempDBInstanceId/>
    <ReadOnlyDBInstanceIds>
     <ReadOnlyDBInstanceId/>
    </ReadOnlyDBInstanceIds>
   </DBInstance>
  </Items>
  <PageNumber>1</PageNumber>
  <TotalRecordCount>2</TotalRecordCount>
  <RequestId>2553A660-E4EB-4AF4-A402-8AFF70A49143</RequestId>
</DescribeDBInstancesResponse>
```

- JSON format

```
{
  "Items":{
    "DBInstance":[
      {
        "DBInstanceDescription":"testforRemarks",
        "DBInstanceId":"rdsxxxxxxxxx",
        "DBInstanceNetType":"Internet",
        "DBInstanceStatus":"Running",
        "DBInstanceType":"Primary",
        "Engine":"MySQL",
```

```
            "ExpireTime":"2014-10-10T16:00:00Z",
            "GuardDBInstanceId ":"",
            "LockMode":"Unlock",
            "LockReason":"",
            "MasterInstanceId":"",
            "PayType":"Prepaid",
            "ReadOnlyDBInstanceIds":{
              "ReadOnlyDBInstanceId":[]
            },
            "RegionId":"cn-hangzhou",
            "TempDBInstanceId":"",
            "ZoneId":"cn-hangzhou-a"
        },
        {
            "DBInstanceDescription":"testforcreate",
            "DBInstanceId":"rdsxxxxxxxxx",
            "DBInstanceNetType":"Intranet",
            "DBInstanceStatus":"Running",
            "DBInstanceType":"Primary",
            "Engine":"MySQL",
            "ExpireTime":"",
            "GuardDBInstanceId ":"",
            "LockMode":"Unlock",
            "LockReason":"",
            "MasterInstanceId":"",
            "PayType":"Postpaid",
            "ReadOnlyDBInstanceIds":{
              "ReadOnlyDBInstanceId":[]
            },
            "RegionId":"cn-hangzhou",
            "TempDBInstanceId":""
        }
    ]
},
"PageNumber":1,
"PageRecordCount":2,
"RequestId":"2553A660-E4EB-4AF4-A402-8AFF70A49143",
"TotalRecordCount":2
```


- **Sample error responses**

JSON format

```
{
  "Code":"UnsupportedOperation",
  "HostId":"rds.aliyuncs.com",
  "Message":"The specified action is not supported.",
  "RequestId":"7463B73D-35CC-4D19-A010-6B8D65D242EF"
}
```

3.2.3.6. ModifyDBInstanceSpec

Description

You can call this operation to change the type or storage space of instances. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is not performing a backup task.
- Either the instance type (DBInstanceClass) or storage space (DBInstanceStorage) parameter must be specified in the request.
- When reducing the amount of disk space, ensure that the target amount of disk space is no less than 1.1 times the actual space used.
- Only regular and read-only instances allow specifications changes. Specifications of disaster recovery and temporary instances cannot be changed.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyDBInstanceSpec	The operation that you want to perform. Set the value to ModifyDBInstanceSpec.
DBInstanceId	String	Yes	rm-bp1xxxxxxxx874	The ID of the instance.
PayType	String	Yes	Postpaid	The billing method of the instance. Valid values: • Postpaid: pay-as-you-go • Prepaid: subscription

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	LTAIKxxxxxxxxOw	The AccessKey ID provided to you by Alibaba Cloud.
ClientToken	String	No	ETnLxxxxxxxxxxxxxxxxxxYBQ	The client token that is used to ensure the idempotence of the request.
DBInstanceClass	String	No	rds.mys2.small	The type of the instance. For more information, see Instance types in Appendix.
DBInstanceStorage	Integer	No	20	The user-defined storage space. Unit: GB. This value must be in 5 GB increments. Valid values for a MySQL instance: 5 to 2000.
EffectiveTime	String	No	Immediate	The time when the changes take effect. Valid values: • Immediately: The modified values of parameters immediately take effect. • MaintainTime: The modified values of parameters take effect within the maintenance window. Default value: Immediate.

Parameter	Type	Required	Example	Description
OwnerAccount	String	No	testuser@aliyun.com	The Apsara Stack tenant account.

Response parameters

Parameter	Type	Example	Description
RequestId	String	1AD222E9-E606-4A42-BF6D-8A4442913CEF	The ID of the request.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyDBInstanceSpec
&DBInstanceId=rds**********bn
&PayType=Postpaid
&DBInstanceStorage=10
&<Common request parameters>
```

Sample success responses

- XML format

```
<ModifyDBInstanceSpecResponse>
      <RequestId>3C****EE-****-4DED-****-1D7******C575</RequestId>
</ModifyDBInstanceSpecResponse>
```

- JSON format

```
{
   "RequestId": " 3C****EE-****-4DED-****-1D7******C575 "
 }
```

Sample error responses

JSON format

```
{
  "Code":"UnsupportedOperation",
  "HostId":"rds.aliyuncs.com",
  "Message":"The specified action is not supported.",
  "RequestId":"7463B73D-35CC-4D19-A010-6B8D65D242EF"
}
```

3.2.3.7. DescribeDBInstanceNetInfo

Description

You can call this operation to view the connection information of a specified instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstanceNetInfo.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information about common response parameters, see Common response parameters.
DBInstanceNetInfos	List<DBInstanceAttribute>	The array that consists of DBInstanceAttribute data.
InstanceNetworkType	String	The network type: Classic: classic network instance

DBInstanceNetInfo parameters

Parameter	Type	Description
ConnectionString	String	The DNS connection string.
IPAddress	String	The IP address of the instance.

Parameter	Type	Description
IPType	String	Valid values for classic network instances: Inner and Public.
Port	String	The port information.
VPCId	String	The ID of the VPC.
VSwitchId	String	The ID of the VSwitch.

3.2.3.8. DescribeDBInstanceIPArrayList

Description

You can call this operation to query the IP whitelist of an instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstanceIPArrayList.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information about common response parameters, see Common response parameters.
Items	List<DBInstanceIPArray>	The array that consists of DBInstanceIPArray data.

DBInstanceIPArray parameters

Parameter	Type	Description
DBInstanceIPArrayName	String	The name of the IP whitelist.
DBInstanceIPArrayAttribute	String	The attribute of the IP whitelist. This parameter is empty by default. Whitelists tagged with "hidden" are not displayed in the console.

Parameter	Type	Description
SecurityIPList	String	The IP addresses in an IP address whitelist. Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats of IP addresses are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or Classless Inter-Domain Routing (CIDR) blocks such as 10.23.12.24/24, where /24 indicates the prefix of the CIDR block is 24-bit. You can replace 24 with a value ranging from 1 to 32.

3.2.3.9. ModifySecurityIps

Description

You can call this operation to modify the IP whitelist of an instance.

Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is not locked.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifySecurityIps.
DBInstanceId	String	Yes	The ID of the instance.
SecurityIps	String	No	The IP addresses in an IP address whitelist. Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats of IP addresses are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or Classless Inter-Domain Routing (CIDR) blocks such as 10.23.12.24/24, where /24 indicates the prefix of the CIDR block is 24-bit. You can replace 24 with a value ranging from 1 to 32.
DBInstanceIPArrayName	String	No	The name of the IP whitelist. If no value is specified, the whitelist named Default will be modified.
DBInstanceIPArrayAttribute	String	No	The attribute of the IP whitelist. This parameter is empty by default. Whitelists tagged with "hidden" are not displayed in the console.

Response parameters

For more information about common response parameters, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifySecurityIps
&DBInstanceId=rds*********abn
&SecurityIps=192. ***.1.2
&<Common request parameters>
```

Sample success responses

- XML format

```
<ModifySecurityIpsResponse>
      <RequestId>65B****2-****-4122-****-B38********E64</RequestId>
</ModifySecurityIpsResponse>
```

- JSON format

```
{
   "RequestId": " 65B****2-****-4122-****-B38********E64"
}
```

3.2.3.10. DescribeRegions

Description

You can call this operation to query the available RDS regions and zones.

You must call this operation to query the available regions before calling the CreateDBInstance operation.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeRegions.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.
Regions	List<RDSRegion>	The array that consists of RDSRegion data.

RDSRegion parameters

Parameter	Type	Description
RegionId	String	The region ID.
ZoneId	String	The zone ID.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeRegions
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeRegionsResponse>
 <Regions>
  <RDSRegion>
<RegionId>cn-hangzhou</RegionId>
<ZoneId>cn-hangzhou-a</RegionId>
  </RDSRegion>
  <RDSRegion>
<RegionId>cn-hangzhou</RegionId>
<ZoneId>cn-hangzhou-b</ZoneId>
  </RDSRegion>
  <RDSRegion>
<RegionId>cn-qingdao</RegionId>
<ZoneId>cn-qingdao-a</ZoneId>
  </RDSRegion>
  <RDSRegion>
<RegionId>cn-beijing</RegionId>
<ZoneId>cn-beijing-a</ZoneId>
  </RDSRegion>
  <RDSRegion>
<RegionId>cn-hongkong</RegionId>
<ZoneId>cn-hongkong-a</ZoneId>
  </RDSRegion>
  <RDSRegion>
<RegionId>cn-shenzhen</RegionId>
<ZoneId>cn-shenzhen-a</ZoneId>
  </RDSRegion>
</Regions>
<RequestId>A36D9720-7902-42A4-B8B9-014A2135E6C3</RequestId>
</DescribeRegionsResponse>
```

- JSON format

```
{
   "Regions": {
    "RDSRegion": [
      {"RegionId": "cn-hangzhou","ZoneId":"cn-hangzhou-a"},
      {"RegionId": "cn-hangzhou","ZoneId":"cn-hangzhou-b"},
      { "RegionId": "cn-qingdao","ZoneId":"cn-qingdao-b"},
      {"RegionId": "cn-beijing","ZoneId": "cn-beijing-a"},
      { "RegionId": "cn-hongkong","ZoneId": "cn-hongkong-a"},
      {"RegionId": "cn-shenzhen","ZoneId": "cn-shenzhen-a"}
    ]
},
   "RequestId": "A3*****0-****-42A4-****-014******E6C3"
}
```

3.2.3.11. DescribeDBInstanceHAConfig

Description

You can call this operation to query the zone information and data replication status of instances.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstanceHAConfig.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
DBInstanceId	String	The ID of the instance.
SyncMode	String	• Sync: synchronous replication • Semi-sync: semi-synchronous replication • Async: asynchronous replication

Parameter	Type	Description
HAMode	String	• RPO: Data persistence is preferred. • RTO: Instance availability is preferred.
HostInstanceInfos	List<NodeInfo>	The array that consists of NodeInfo data.

NodeInfo parameters

Parameter	Type	Description
NodeId	String	The unique ID of the primary or secondary instance.
NodeType	String	• Master: primary instance • Slave: secondary instance
RegionId	String	The ID of the region.
ZoneId	String	The ID of the zone.
SyncStatus	String	• NotAvailable: A failure has occurred and synchronization is unavailable. • Syncing: Synchronization is in process, with a probability of data loss during failovers. • Synchronized: Synchronization is completed. • NotSupport: Synchronization is not supported by the engine type or version.
LogSyncTime	String	The time when the secondary instance receives logs.
DataSyncTime	String	The time when the secondary instance receives data.

3.2.3.12. PurgeDBInstanceLog

Description

You can call this operation to clear all binlogs for a MySQL instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to PurgeDBInstanceLog.
DBInstanceId	String	Yes	The ID of the instance for which all binlogs are cleared.

Response parameters

For more information about common response parameters, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=PurgeDBInstanceLog
&DBInstanceId=rds********abn
&<Common request parameters>
```

Sample success responses

- XML format

```
<PurgeDBInstanceLogResponse>
      <RequestId>65*****2-****-4122-****-B38*********64</RequestId>
</PurgeDBInstanceLogResponse>
```

- JSON format

```
{
   "RequestId": " 65*****2-****-4122-****-B38*********64"
}
```

3.2.3.13. ModifyDBInstanceDescription

Description

You can call this operation to modify the description of an instance to identify the instance easily. For example, you can modify the description of an instance to *Alibaba Cloud testing environment instance*.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyDBInstanceDescription.
DBInstanceId	String	Yes	The ID of the instance.

Parameter	Type	Required	Description
DBInstanceDescription	String	Yes	The description of the instance.

Response parameters

This operation returns only common response parameters. For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyDBInstanceDescription
&DBInstanceId=rds**********abn
&DBInstanceDescription=testwangyichengdescribe
&<Common request parameters>
```

Sample responses

- XML format

```
< ModifyDBInstanceDescriptionResponse>
      <RequestId>17F****E-****-4337-****-9C2*******D74</RequestId>
</ModifyDBInstanceDescriptionResponse>
```

- JSON format

```
{
   "RequestId": " 17F****E-****-4337-****-9C2*******D74"
}
```

3.2.3.14. ModifyDBInstanceMaintainTime

Description

You can call this operation to modify the scheduled period for routine maintenance. You can set the maintenance window to an off-peak period.

Alibaba Cloud performs routine maintenance during the maintenance window to minimize the impact on your services.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyDBInstanceMaintainTime.
DBInstanceId	String	Yes	The ID of the instance.
MaintainTime	String	Yes	The maintenance window of the instance. Valid values: • 22:00Z-02:00Z • 02:00Z-06:00Z • 06:00Z-10:00Z • 10:00Z-14:00Z • 14:00Z-18:00Z • 18:00Z-22:00Z

Response parameters

For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyDBInstanceMaintainTime
&DBInstanceId=rds*********abn
&MaintainTime=22:00Z-02:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<ModifyDBInstanceMaintainTimeResponse>
      <RequestId>65B****2-****-4122-****-B38*******64</RequestId>
</ModifyDBInstanceMaintainTimeResponse>
```

- JSON format

```
{
   "RequestId": " 65B****2-****-4122-****-B38*******64"
 }
```

3.2.3.15. ModifyDBInstanceHAConfig

Description

You can call this operation to modify the instance data replication mode and high availability switching rules.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyDBInstanceHAConfig.
DBInstanceId	String	Yes	The ID of the instance.
SyncMode	String	Yes	• Sync: synchronous replication • Semi-sync: semi-synchronous replication • Async: asynchronous replication
HAMode	String	Yes	• RPO: Data persistence is preferred. • RTO: Instance availability is preferred.

Response parameters

For more information, see Common response parameters.

3.2.3.16. SwitchDBInstanceHA

Description

You can call this operation to switch between the primary and secondary instances. RDS instances are composed of primary and secondary instances distributed across different servers. The new primary instance bears the service traffic.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to SwitchDBInstanceHA.
DBInstanceId	String	Yes	The ID of the instance.
NodeId	String	Yes	The unique ID of the instance. You can call the DescribeDBInstanceHAConfig operation to query the value of this parameter.

Parameter	Type	Required	Description
Force	String	Yes	• Yes: forced. • No: unforced. Note Default value: No.

Response parameters

For more information, see Common response parameters.

3.2.3.17. CreateReadOnlyDBInstance

Description

You can call this operation to create a read-only instance for an instance. Before you call this operation, ensure that the following requirements are met:

- The primary instance is in the running but not failover state.
- The read-only instance and the primary instance are in the same region.
- An RDS instance can have up to five read-only instances.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateReadOnlyDBInstance.
RegionId	String	Yes	The region ID of the instance. It can be up to 50 characters in length. You can call the DescribeRegions operation to query the most recent region list.
ZoneId	String	Yes	The zone ID of the instance. You can call the DescribeRegions operation to query the most recent zone list.
DBInstanceId	String	Yes	The ID of the primary instance.
EngineVersion	String	Yes	The database version of the instance. Set the value to 5.6.
DBInstanceClass	String	Yes	The type of the instance. For more information, see Instance types in Appendix.
DBInstanceStorage	Integer	Yes	The user-defined storage space. Valid values: 5 to 1000. This value must be in 5 GB increments. Unit: GB.
DBInstanceDescription	String	No	The description of the instance. It cannot start with http:// or https://. It must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-).

Parameter	Type	Required	Description
PayType	String	Yes	Postpaid
ClientToken	String	Yes	The client token that is used to ensure the idempotence of the request.
InstanceNetworkType	String	No	• VPC • Classic Note If no value is specified, a classic network-type instance is created by default.
VPCId	String	No	The ID of the VPC.
VSwitchId	String	No	The ID of the VSwitch.
PrivateIpAddress	String	No	The VPC IP address that belongs to the specified VSwitch. If no value is specified, the system automatically assigns a VPC IP address based on VPCId and VSwitchId.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
DBInstanceId	String	The ID of the instance.
OrderId	String	The ID of the order.
ConnectionString	String	The endpoint used to connect to the database.
Port	String	The port used to connect to the database.

Examples

Sample requests

```
https://rds.example.com/?Action=CreateReadOnlyDBInstance
&RegionId=cn-hangzhou
&ZoneId=cn-hangzhou-b
&DBInstanceId=rds*********abn
&EngineVersion=5.6
&DBInstanceClass=rds.mys2.small
&DBInstanceStorage=10
&PayType=Postpaid
&ClientToken=ETnLK******************wHlYBQ
&<[Common request parameters]>
```

Sample success responses

- XML format

```
<CreateDBInstanceResponse>
    <OrderId>100789370230206</OrderId>
    <ConnectionString>rds*********abn.mysql.rds.aliyuncs.com </ConnectionString>
    <DBInstanceId>rds********abn</DBInstanceId>
    <port>3306</port>
    <RequestId>1E4****0-****-43DA-****-EAF********DC</RequestId>
</CreateDBInstanceResponse>
```

- JSON format

```
 {
  "OrderId": "100789370230206",
  "ConnectionString": "rds*********abn.mysql.rds.aliyuncs.com",
  "DBInstanceId": "rds********abn",
  "Port": "3306",
  "RequestId": "1E4****0-****-43DA-****-EAF********DC"
 }
```

3.2.3.18. DescribeDBInstanceSSL

Description

This operation is only supported on MySQL 5.6.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstanceSSL.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
ConnectionString	String	The endpoint protected by the SSL certificate.
SSLExpireTime	String	The expiration time of the SSL certificate. The time is displayed in the YYYY-MM-DDThh:mm:ssZ format.
RequireUpdate	String	• Yes: An immediate upgrade is required. • No: An immediate upgrade is not required.
RequireUpdateReason	String	The reasons why an upgrade is recommended.

3.2.3.19. ModifyDBInstanceSSL

Description

This operation is only supported on MySQL 5.6.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyDBInstanceSSL.
DBInstanceId	String	Yes	The ID of the instance.
ConnectionString	String	Yes	Each instance can only have one endpoint protected by the SSL certificate. The SSL certificate can be created or updated for the endpoint.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.
SSLExpireTime	String	The expiration time of the SSL certificate. The time is in the yyyy-MM-ddThh:mm:ssZ format.

3.2.4. Database management

3.2.4.1. CreateDatabase

Description

You can call this operation to create a new database in an instance. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is not locked by the system.
- The maximum number of databases in the instance is not reached.
- The instance is a primary instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateDatabase.
DBInstanceId	String	Yes	The ID of the instance.
DBName	String	Yes	The unique name of the database. The name can be up to 64 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter. For more information, see Forbidden keywords in Appendix.
CharacterSetName	String	Yes	The character set. Valid values for a MySQL instance: • utf8 • gbk • latin1 • utf8mb4

Parameter	Type	Required	Description
DBDescription	String	No	The description of the database. It cannot start with http:// or https://. It must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-).

Response parameters

For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=CreateDatabase
&CharacterSetName=gbk
&DBName=test****
&DBInstanceId=riau********************449201
&<[Common request parameters]>
```

Sample success responses

- XML format

```
<CreateDatabaseResponse>
      <RequestId>5A7****0-****-4E08-****-590********27</RequestId>
</CreateDatabaseResponse>
```

- JSON format

```
{
  "RequestID":"5A7****0-****-4E08-****-590********27"
 }
```

3.2.4.2. DeleteDatabase

Description

You can call this operation to delete a database from an instance.

Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is a primary instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set this parameter to DeleteDatabase.
DBInstanceId	String	Yes	The ID of the instance.
DBName	String	Yes	The name of the database.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=DeleteDatabase
&DBName=test****
&DBInstanceId=ria**********9201
&<[Common request parameters]>
```

Sample success responses

- XML format

```
<DeleteDatabaseResponse>
      <RequestId>5A7****0-****-4E08-****-590*******927</RequestId>
</DeleteDatabaseResponse>
```

- JSON format

```
{
    "RequestId":"07F*****E-****-408A-****-072*******0F3"
 }
```

3.2.4.3. DescribeDatabases

Description

You can call this operation to query the databases of a specified instance or information of a specified database. If the request parameters are entered incorrectly, an error prompt is returned and the returned data is empty.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDatabases.
DBInstanceId	String	Yes	The ID of the instance.
DBName	String	No	The name of the database.
DBStatus	String	No	The status of the database. Valid values: • Creating: The database is being created. • Running: The database is being used. • Deleting: The account is being deleted.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
Databases	List<Database>	The array that consists of Database data.

Database parameters

Parameter	Type	Description
DBName	String	The name of the database.
DBInstanceId	String	The ID of the instance to which the database belongs.
Engine	String	The engine of the database.
DBStatus	String	The status of the database. • Creating: The database is being created. • Running: The database is being used. • Deleting: The account is being deleted.
CharacterSetName	String	The character set.

Parameter	Type	Description
DBDescription	String	The description of the database.
Accounts	List<AccountPrivilegeInfo>	The array that consists of AccountPrivilegeInfo data.
Account	String	The name of the account.
AccountPrivilege	String	The permissions of the account on the database.

AccountPrivilegeInfo parameters

Parameter	Type	Description
Account	String	The name of the account.
AccountPrivilege	String	The permissions of the account on the database.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeDatabases
&DBInstanceId=rds**********ju
&<Common request parameters>
```

Sample success responses

- XML format

```
< DescribeDatabasesResponse>
      <RequestId>260****6-****-4903-****-61A*******D94</RequestId>
       <Databases>
         <Database>
           <Engine>MySQL</Engine>
           <DBName>testdb</DBName>
           <CharacterSetName>utf8</CharacterSetName>
           <DBStatus>Creating</DBStatus>
           <DBInstanceId>rds**********ju</DBInstanceId>
           <Accounts>
               <AccountPrivilegeInfo></AccountPrivilegeInfo>
           <Accounts>
         </Database>
         <Database>
           <Engine>MySQL</Engine>
           <DBName>testdb2</DBName>
           <CharacterSetName>gbk</CharacterSetName>
           <DBStatus>Creating</DBStatus>
           <DBInstanceId>rds**********ju</DBInstanceId>
           <Accounts>
               <AccountPrivilegeInfo></AccountPrivilegeInfo>
           <Accounts>
         </Database>
      </Databases>
</ DescribeDatabasesResponse>
```

- JSON format

```
{
  "RequestId": "26*****6-****-4903-****-61A*******CD94",
  "Databases": {
   "Database": [
    {
     "Engine": "MySQL",
     "CharacterSetName": "utf8",
     "DBStatus": "Creating",
     "DBDescription": "",
     "DBInstanceId": "rds3*********zju",
     "Accounts": {
      " AccountPrivilegeInfo": []
     },
     "DBName": "testdb"
    },
    {
     "Engine": "MySQL",
     "CharacterSetName": "gbk",
     "DBStatus": "Creating",
     "DBDescription": "",
     "DBInstanceId": "rds3m*********zju",
     "Accounts": {
      " AccountPrivilegeInfo": []
     },
     "DBName": "testdb2"
    }
   ]
  }
 }
```


3.2.4.4. ModifyDBDescription

Description

You can call this operation to modify the description of a database to facilitate user records. For example, you can modify the description of a database to Instance database A for Alibaba Cloud testing environment.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyDBDescription.
DBInstanceId	String	Yes	The ID of the instance.
DBName	String	Yes	The name of the database.
DBDescription	String	Yes	The description of the database.

Response parameters

For more information about common response parameters, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyDBDescription
&DBInstanceId=rds**********abn
&DBInstanceDescription=testwangyichengDBdescribe
&<Common request parameters>
```

Sample success responses

- XML format

```
< ModifyDBDescriptionResponse>
      <RequestId>17F****E-****-4337-****-9C2********D74</RequestId>
</ ModifyDBDescriptionResponse>
```

- JSON format

```
{
  "RequestId": " 17F****E-****-4337-****-9C2********D74"
}
```

3.2.5. Account management

3.2.5.1. CreateAccount

Description

> **Note**
>
> In the same instance, one account can manage multiple databases with different permissions.

You can call this operation to create an account for a database. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The database is in the running state.
- The instance is not locked.
- The maximum number of accounts for a single instance is not reached.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateAccount.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The unique name of the account. The name can be up to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter. For more information, see Forbidden keywords in Appendix.
AccountPassword	String	Yes	The password for the account. It must be 6 to 32 characters in length and can contain letters, digits, and underscores (_).
AccountType	String	No	Valid values: • Normal: a standard account • Super: a super account Note Default value: Normal.
AccountDescription	String	No	The description of the account. It must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It cannot start with http:// or https://.

Response parameters

For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=CreateAccount
&AccountName=testacc02
&AccountPassword=pw***4
&DBInstanceId=riauvjz6za**********449201
&<[Common request parameters]>
```

Sample success responses

- XML format

```
<CreateAccountResponse>
      <RequestId>D4D****A-****-440A-****-EE3*******9DD</RequestId>
</CreateAccountResponse>
```

- JSON format

```
{
    "RequestId":"D4D****A-****-440A-****-EE3*******9DD"
 }
```

3.2.5.2. DeleteAccount

Description

You can call this operation to delete a database account.

The instance must be in the running state.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DeleteAccount.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The unique name of the account. The name can be up to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter. For more information, see Forbidden keywords in Appendix.

Response parameters

For more information about common response parameters, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=DeleteAccount
&DBInstanceId=riaqu********************018
&AccountName=wangy*******st
&<[Common request parameters]>
```

Sample success responses

- XML format

```
<DeleteAccountResponse>
      <RequestId>91E****5-****-4955-****-C7*******C66</RequestId>
</DeleteAccountResponse>
```

- JSON format

```
{
    "RequestID":"91E****5-****-4955-****-C7*******C66"
 }
```

3.2.5.3. DescribeAccounts

Description

You can call this operation to query the accounts of a specified instance or information of a specified account.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeAccounts.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The name of the account.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
Accounts	List<DBInstanceAccount>	The array that consists of DBInstanceAccount data.

DBInstanceAccount parameters

Parameter	Type	Description
DBInstanceId	String	The ID of the instance.
AccountName	String	The name of the account.
AccountStatus	String	The status of the account. Valid values: • Unavailable • Available
AccountDescription	String	The description of the account.
DatabasePrivileges	List<DatabasePrivilege>	The array that consists of DatabasePrivilege data.
AccountType	String	Valid values: • Normal: a standard account • Super: a super account

DatabasePrivilege parameters

Parameter	Type	Description
DBName	String	The name of the database.
AccountPrivilege	String	The permissions of the account on the database.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeAccounts
&DBInstanceId=rds*********uie
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeAccountsResponse>
      <RequestId>260****6-****-4903-****-61A*******94</RequestId>
       <Accounts>
          <DBInstanceAccount>
            <AccountName>MySQL</AccountName>
            <DBInstanceId>testdb</DBInstanceId>
            <AccountStatus>utf8</AccountStatus>
            <AccountDescription></AccountDescription>
            <DatabasePrivileges>
    <DatabasePrivilege></DatabasePrivilege>
</DatabasePrivileges>
          <DBInstanceAccount>
       </Accounts>
</DescribeAccountsResponse>
```

- JSON format

```
{
   "Accounts": {
    "DBInstanceAccount": [
     {
       "AccountDescription": "",
       "DBInstanceId": "rds**********ie",
       "DatabasePrivileges": {
        "DatabasePrivilege": []
       },
       "AccountStatus": "Unavailable",
       "AccountName": "testaccount"
     }
    ]
   },
   "RequestId": "8D9****9-****-43B4-****-0A5*********008"
 }
```

3.2.5.4. GrantAccountPrivilege

Description

You can call this operation to grant database access permissions to accounts. An account can be authorized to access one or more databases.

If the specified account already has access permissions on the specified database, the operation directly returns a successful response.

The instance must be in the running state.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to GrantAccountPrivilege.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The name of the account.
DBName	String	Yes	The name of the database to be accessed by the account.
AccountPrivilege	String	Yes	The permissions of the database account on the database. Valid values: • ReadOnly • ReadWrite

Response parameters

For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=GrantAccountPrivilege
&AccountName=testacc02
&AccountPrivilege=readwrite
&DBName=testdb03
&DBInstanceId=riauvj**********201
&<Common request parameters>
```

Sample success responses

- XML format

```
<GrantAccountPrivilegeResponse>
      <RequestId>81B****9-****-4B7F-****-5F********DEA7</RequestId>
</GrantAccountPrivilegeResponse>
```

- JSON format

```
{
      "RequestId":"81B****9-****-4B7F-****-5F********DEA7"
}
```

3.2.5.5. RevokeAccountPrivilege

Description

You can call this operation to revoke database access permissions from an account. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The database is in the running state.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to RevokeAccountPrivilege.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The name of the account.
DBName	String	Yes	The name of the database.

Response parameters

For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=RevokeAccountPrivilege
&AccountName=testacc02
&DBName=testdb03
&DBInstanceId=riauvj******************9201
&<Common request parameters>
```

Sample success responses

- XML format

```
<RevokeAccountPrivilegeResponse>
      <RequestId>E22****A-****-4992-****-CE8********626</RequestId>
</RevokeAccountPrivilegeResponse>
```

- JSON format

```
{
      "RequestId":"E22****A-****-4992-****-CE8********626"
}
```

3.2.5.6. ModifyAccountDescription

Description

You can call this operation to modify the description of an account to facilitate user records. For example, you can modify the description of an account to Account A for Alibaba Cloud testing environment.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyAccountDescription.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The unique name of the account. The name can be up to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter. For more information, see Forbidden keywords in Appendix.

Parameter	Type	Required	Description
AccountDescription	String	Yes	The description of the account. It must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It cannot start with http:// or https://. It must start with a letter.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyAccountDescription
&DBInstanceId=rds**********abn
&AccountName=wang********test
&AccountDescription=testAccoutdescribe
&<Common request parameters>
```

Sample success responses

- XML format

```
< ModifyAccountDescription>
      <RequestId>17F****E-****-4337-****-9C********3D74</RequestId>
</ ModifyAccountDescription>
```

- JSON format

```
{
   "RequestId": " 17F****E-****-4337-****-9C********3D74"
}
```

3.2.5.7. ResetAccountPassword

Description

You can call this operation to reset the account password. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The current instance is not locked.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ResetAccountPassword.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The name of the account.
AccountPassword	String	Yes	The new password. It must be 6 to 32 characters in length and can contain letters, digits, and underscores (_).

Response parameters

For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ResetAccountPassword
&AccountName=testacc02
&AccountPassword=ne*******4
&DBInstanceId=riauvjz6za******************201
&<Common request parameters>
```

Sample success responses

- XML format

```
<ResetAccountPasswordResponse>
      <RequestId>D4D****A-****-440A-****-EE3*********DD</RequestId>
</ResetAccountPasswordResponse>
```

- JSON format

```
{
    "RequestId":"D4D****A-****-440A-****-EE3*********DD"
}
```

3.2.5.8. ResetAccount

Description

You can call this operation to reset the initial account of an instance.

You cannot perform this operation on ApsaraDB RDS for MySQL 5.6 instances that do not have privileged accounts.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ResetAccount.
DBInstanceId	String	Yes	The ID of the instance.
AccountName	String	Yes	The name of the account.
AccountPassword	String	Yes	The password of the database account.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

3.2.6. Backup and restoration

3.2.6.1. CreateBackup

Description

You can call this operation to manually create a backup. Up to 10 backups can be created for an instance in a single day.

Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The most recent backup has been completed.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateBackup.
DBInstanceId	String	Yes	The ID of the instance.
BackupMethod	String	No	• Logical • Physical Note Default value: Physical. Logical backup cannot be used for instances without databases.
BackupStrategy	String	No	The logical backup range. Valid values: • db: single-database backup. • instance: instance backup. Note This parameter is valid only when BackupMethod is set to Logical.
DBName	String	No	The name of the database for logical single-database backup. Note This parameter is valid only when BackupMethod is set to Logical and BackupStrategy is set to db.
BackupType	String	No	• Auto: specifies whether it is a full backup or an incremental backup through automatic calculation. • FullBackup: specifies a full backup. Note Default value: Auto.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action= CreateBackup
&DBInstanceId=riau**********9201
&<Common request parameters>
```

Sample success responses

- XML format

```
<CreateBackupResponse>
    <RequestId>DA1****9-****-4417-****-65******40D</RequestId>
</CreateBackupResponse>
```

- JSON format

```
{
    "RequestId":"DA1****9-****-4417-****-65******40D"
}
```

3.2.6.2. CloneDBInstance

Description

You can call this operation to clone a new instance with historical data based on the backup file of an existing instance or a specified time.

The new instance will have the same whitelist, SQL audit, alert threshold, backup, and parameter settings as the existing instance.

The data in the new instance is the same as historical data in the backup file. The clone feature can only be used with MySQL engines.

Database account information is cloned differently for different account types:

- If the primary instance has a privileged account, instances that are cloned from it will also carry the same privileged account information.
- If the primary instance has only standard accounts, instances that are cloned from it will also carry the information of the account used when the backup file is generated or at the specified time.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CloneDBInstance.
DBInstanceId	String	Yes	The ID of the instance.
BackupId	String	No	The ID of the backup set. You can call the DescribeBackups operation to query backup set IDs.
RestoreTime	String	No	A point in time within the backup retention period to obtain a backup set from. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format. Example: 2011-06-11T16:00:00Z. Either BackupId or RestoreTime is required.
PayType	String	Yes	The billing method of the instance. Valid values: • Postpaid: pay-as-you-go • Prepaid: subscription
Period	String	No	This parameter is required if PayType is set to Prepaid. The subscription period of the instance. Valid values: • Year: subscription on a yearly basis • Month: subscription on a monthly basis
UsedTime	String	No	The number of the subscription units. This parameter is required if PayType is set to Prepaid. Valid values: positive integers such as 1, 2, and 3.
DBInstanceClass	String	No	The type of the instance. For more information, see Instance types in Appendix. This parameter is optional. If it is not specified, the new instance is of the same type as the primary instance.
DBInstanceStorage	String	No	The disk space. Unit: GB. This value must be in 5 GB increments and is the same as that of the primary instance by default. RDS instances of the dedicated host type: 3000. RDS instances of other types: 5 to 2000.
InstanceNetworkType	String	No	The network type of the instance. Valid values: • VPC • Classic Note The new instance has the same network type as the primary instance by default.
VPCId	String	No	The ID of the VPC.
VSwitchId	String	No	The ID of the VSwitch.

Parameter	Type	Required	Description
PrivateIpAddress	String	No	The VPC IP address that belongs to the specified VSwitch. If no value is specified, the system automatically assigns a VPC IP address.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
DBInstanceId		The ID of the instance.
OrderId		The ID of the order.

3.2.6.3. DescribeBackups

Description

You can call this operation to query the backup sets of an instance. Only backup sets in the Success state can be used for database restoration.

RDS provides backup files for you to download.

- When the DownloadLink is NULL, RDS does not provide a download URL.
- When DownloadLink is not NULL, you can use this URL to download the backup file through Wget (with double quotation marks), a browser, or a user program. The expiration time for this URL is specified by LinkExpiredTime. Use it before the expiration time.

  If the URL has expired, the following error code is displayed:

```
<? xml version="1.0" encoding="UTF-8" ? >
<Error>
<Code>AccessDenied</Code>
<Message>Request has expired. </Message>
<Expires>2012-12-25T09:47:52.000Z</Expires>
<ServerTime>2012-12-25T09:49:00.000Z</ServerTime>
<RequestId>50D9768CA801C2F102005C70</RequestId>
<HostId>oss-test.aliyun-inc.com</HostId>
</Error>
```

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeBackups.
DBInstanceId	String	Yes	The ID of the instance.
BackupId	Integer	No	The ID of the backup set.
BackupStatus	String	No	The status of the backup set. Valid values: • Success: The backup task is completed. • Failed: The backup task fails.
BackupMode	String	No	The type of the backup. Valid values: • Automated: a periodic backup task. • Manual: a temporary backup task.
StartTime	String	Yes	The beginning of the time range where the query is performed. Specify the time in the yyyy-MM-ddTHH:mmZ format. Example: 2011-06-11T15:00Z.
EndTime	String	Yes	The end of the time range where the query is performed. Specify the time in the yyyy-MM-ddTHH:mmZ format. Example: 2011-06-11T16:00Z. The end time must be later than the start time.
PageSize	Integer	No	The number of records to return on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNumber	Integer	No	The number of the page to return. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
TotalRecordCount	Integer	The total number of backup sets.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of backup sets on the current page.
Items	List<Backup>	The array that consists of Backup data.

3.2.6.4. DescribeBackupPolicy

Description

You can call this operation to query the backup policy configured for the system. RDS periodically backs up an instance based on the user-defined system configuration.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeBackupPolicy.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
BackupRetentionPeriod	String	The period during which the backup is retained in the system. Unit: days. Valid values: 7 to 730 days.
PreferredBackupTime	String	The time to execute data backup. Specify the time in the HH:mmZ- HH:mmZ format.
PreferredBackupPeriod	String	The cycle to execute data backup. Valid values: • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday
BackupLog	String	Specifies whether to enable log backup. Valid values: • Enable • Disabled
LogBackupRetentionPeriod	String	The log backup retention period. Valid values: 7 to 730 days.

3.2.6.5. ModifyBackupPolicy

Description

You can call this operation to modify a backup policy. RDS periodically backs up instances based on the user-defined system configuration.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyBackupPolicy.
DBInstanceId	String	Yes	The ID of the instance.
PreferredBackupTime	String	Yes	The time to execute backup. Specify the time in the HH:mmZ- HH:mmZ format.
PreferredBackupPeriod	String	Yes	The cycle to execute data backup. Valid values: • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday
BackupRetentionPeriod	String	No	The period during which the backup is retained in the system. Unit: days. Valid values: 7 to 730. Default value: 7.
BackupLog	String	No	Specifies whether to enable log backup. Valid values: • Enable • Disabled
LogBackupRetentionPeriod	String	No	The period during which the log backup is retained in the system. It cannot exceed BackupRetentionPeriod. Valid values: 7 to 730 days.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyBackupPolicy
&DBInstanceId=riauvj******************201
&PreferredBackupTime=00:00Z-01:00Z
&PreferredBackupPeriod=Monday
&<Common request parameters>
```

Sample success responses

- XML format

```
<CreateBackupResponse>
      <RequestId>DA1****9-****-4417-****-65E*********D</RequestId>
</CreateBackupResponse>
```

- JSON format

```
{
      "RequestId":"DA1****9-****-4417-****-65E*********D"
}
```

3.2.6.6. RestoreDBInstance

Description

You can call this operation to restore a backup set to an instance.

Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance does not have an ongoing migration task.
- The instance is not locked.
- The instance backup set is in the Success state.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to RestoreDBInstance.
DBInstanceId	String	Yes	The ID of the instance.

Parameter	Type	Required	Description
BackupId	Integer	Yes	The ID of the backup set.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=RestoreDBInstance
&BackupId=103*****60
&DBInstanceId=rds***********61k3
&<Common request parameters>
```

Sample success responses

- XML format

```
<RestoreDBInstanceResponse>
    <RequestId>374****9-****-40AA-****-9E**********29</RequestId>
</RestoreDBInstanceResponse>
```

- JSON format

```
{"RequestId":"374****9-****-40AA-****-9E**********29"}
```

3.2.6.7. DeleteBackup

Description

You can call this operation to delete data backup files that have been stored for longer than seven days or longer than the log backup retention period.

This operation only deletes the specified backup sets for the target instance. It does not delete the backup sets for associated instances, such as read-only, disaster recovery, or cloned instances.

> **Note** The instance must be in the running state.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DeleteBackup.
DBInstanceId	String	Yes	The ID of the instance.
BackupId	String	Yes	The ID of the backup set. Backup set IDs can be obtained through DescribeBackups. You can enter up to 100 backup set IDs in a single request. Separate multiple IDs with commas (,). Only backup sets that are in the Enabled state can be deleted by the DescribeBackups operation.

Response parameters

For more information about common response parameters, see Common response parameters.

3.2.7. Monitoring management

3.2.7.1. DescribeResourceUsage

Description

You can call this operation to query the resource usage information of an instance. When this operation is called, the system returns the amount of space occupied by the specified instance.

The DBInstanceId parameter is used to obtain instance resource usage information.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeResourceUsage.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
DBInstanceId	String	The ID of the instance.
Engine	String	The engine of the database.

Parameter	Type	Description
DiskUsed	Integer	The total space occupied by data files and log files. Unit: bytes. -1 indicates that no space is used.
DataSize	Integer	The disk space that is used to store user data. Unit: bytes. -1 indicates that no space is used.
LogSize	Integer	The disk space that is used to store log files. Unit: bytes. -1 indicates that no space is used.
BackupSize	Integer	The disk space that is used to store backup files. Unit: bytes. -1 indicates that no space is used.
ColdBackupSize	Integer	The total space occupied by cold backup data. Unit: bytes. -1 indicates that no space is used.
SQLSize	Integer	The total space occupied by SQL data. Unit: bytes. -1 indicates that no space is used.

3.2.7.2. DescribeDBInstancePerformance

Description

You can call this operation to query the performance metrics of an instance over a specified period of time based on the performance parameters.

There are three output formats based on different query time ranges:

- When the query time range is less than or equal to one day, performance metrics are displayed at one-minute intervals. Queries with the time range from one to seven days are not supported.
- When the query time range is greater than seven days but less than or equal to 15 days, performance metrics are displayed at one-hour intervals. Queries with the time range greater than 15 days and less than 30 days are not supported.
- When the query time range is greater than or equal to 30 days but less than or equal to one year, performance metrics are displayed at one-day intervals.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstancePerformance.
DBInstanceId	String	Yes	The ID of the instance.
Key	Integer	Yes	The monitoring metrics for database performance. Separate multiple metrics with commas (,). For more information about the metrics, see Performance metrics.

Parameter	Type	Required	Description
StartTime	String	Yes	The beginning of the time range where the query is performed. Specify the time in the YYYY-MM-DDTHH:mmZ format. Example: 2012-06-11T15:00Z.
EndTime	String	Yes	The end of the time range where the query is performed. Specify the time in the YYYY-MM-DDTHH:mmZ format. Example: 2012-06-11T15:00Z.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
DBInstanceId	String	The name of the instance you have created.
Engine	String	The engine of the database.
StartTime	String	The beginning of the time range where the query was executed. The time is displayed in the YYYY-MM-DDTHH:mmZ format. Example: 2011-05-30T03:29Z.
EndTime	String	The end of the time range where the query was executed. The time is displayed in the YYYY-MM-DDTHH:mmZ format. Example: 2011-05-30T03:29Z. The end time is later than the start time.
PerformanceKeys	List<PerformanceKey>	The array that consists of PerformanceKey data. Format: {perf1, perf2, perf3, ...}.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeDBInstancePerformance
&DBInstanceId=riauvjz6**********449201
&key= MySQL_N**********ffic
&StartTime=2013-01-11T15:00:00Z
&EndTime=2013-06-05T15:00:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeDBInstancePerformanceResponse>
  <RequestId>A54****2-****-4BF3-****-0A8*******E7</RequestId>
  <DBInstanceID>riau******************29449201</DBInstanceID>
  <StartTime>2013-01-11T15:00:00Z</StartTime>
  <EndTime>2013-10-17T15:00Z</EndTime>
  <Engine>MySQL</Engine>
   <PerformanceKeys>
   <PerformanceKey>
      <Key>MySQL_N**********ffic</Key>
      <Unit>KB</Unit>
      <ValueFormat>recv_k&amp;amp;sent_k</ValueFormat>
      <Values><Values/>
   </PerformanceKey>
   </PerformanceKeys>
</DescribeDBInstancePerformanceResponse>
```

- JSON format

```
{
  "RequestId"A5409D02-D661-4BF3-8F3D-0A814D0574E7,
  "DBInstanceID": riauvj******************01,
  "StartTime": "2012-06-11T15:00Z",
  "EndTime": "2013-10-17T15:00Z",
  "Engine": "MySQL",
  "PerformanceKeys": {
     PerformanceKey[
        {
          "Key": "MySQL_NetworkTraffic",
          "Unit": "KB",
          "ValueFormat": "recv_k&sent_k",
          "Values": {
             "PerformanceValue":[]
          }
        }
     ]
  }
}
```

3.2.7.3. DescribeDBInstanceMonitor

Description

You can call this operation to query the monitoring behavior of an instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDBInstanceMonitor.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.
Period	String	The monitoring period. Unit: seconds.

3.2.7.4. ModifyDBInstanceMonitor

Description

You can call this operation to modify the monitoring behavior of an instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyDBInstanceMonitor.
DBInstanceId	String	Yes	The ID of the instance.
Period	String	Yes	The monitoring period. Unit: seconds. Valid values: 60 and 300. Default value: 300.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.

3.2.8. Parameter management

3.2.8.1. DescribeParameterTemplates

Description

You can call this operation to query the parameter template list. The list includes the following parameters:

- parameters
- default parameter values
- whether the parameter can be modified
- whether the database must be restarted for them to take effect
- regular expressions acting as parameter validation rules

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeParameterTemplates.
Engine	String	Yes	The engine of the database. Valid value: MySQL.
EngineVersion	String	Yes	The version of the database engine. Valid values for MySQL databases: 5.6.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
Engine	String	The engine of the database.
EngineVersion	String	The version of the database engine.
ParameterCount	Integer	The number of parameters.
Parameters	List<TemplateRecord>	The array that consists of TemplateRecord data. Format: {parameter1, parameter2, parameter3, ...}.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeParameterTemplates
&Engine=MySQL
&EngineVersion=5.6
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeParameterTemplatesResponse>
    <Engine>mssql</Engine>
    <EngineVersion>2008r2</EngineVersion>
    <ParameterCount>1</ParameterCount>
    <Parameters>
      <TemplateRecord>
      <CheckingCode>[0-100]</CheckingCode>
      <ForceRestart>True</ForceRestart>
      <Factor>1</Factor>
      <ParameterDescription> This option sets the default fill factor value for the server range. A fill f
actor is provided to optimize index data storage and performance. </ParameterDescription>
      <ParameterName>fill factor</ParameterName>
      <ParameterValue>0</ParameterValue>
      <ForceModify>True</ForceModify>
      <Unit>INT</Unit>
      </TemplateRecord>
    </Parameters>
    <RequestId>7B9****A-****-4979-****-7D*********C0</RequestId>
</DescribeParameterTemplatesResponse>
```

- JSON format

```
{
"Engine": "mssql",
"EngineVersion": "2008r2",
"ParameterCount":1
   "Parameters": {
        "TemplateRecord": [
          {
            "ParameterDescription": "This option sets the default fill factor value for the server range.
A fill factor is provided to optimize index data storage and performance.",
            "ForceRestart": "True",
            "CheckingCode": "[0-100]"
            "Factor":"1"
            "ParameterName":"fill factor"
            "ParameterValue":"0"
            "ForceModify":"True"
            "Unit":"INT"
          }
         ]
       },
   "RequestId": "7B9****A-****-4979-****-7D*********C0"
}
```


3.2.8.2. DescribeParameters

Description

You can call this operation to query the current instance parameter configuration.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeParameters.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description

<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.
Engine	String	The engine of the database.
EngineVersion	String	The engine version of the database.
RunningParameters	List<DBInstanceParameter>	The array that consists of running DBInstanceParameter data. Format: {parameter1,parameter2,parameter3, ...}.
ConfigParameters	List<DBInstanceParameter>	The array that consists of synchronizing DBInstanceParameter data. Format: {parameter1,parameter2,parameter3, ...}.

DBInstanceParameter parameters

Parameter	Type	Description
ParameterName	String	The name of the parameter.
ParameterValue	String	The value of the parameter.
ParameterDescription	String	The description of the parameter.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeParameters
&DBInstanceId=ria**********449201
&<Common request parameters>
```

Sample success responses

- XML format

```
< DescribeParametersResponse>
   <RequestId>2A7****2-****-4D6B-****-69*********3F1</RequestId>
   <Engine>mssql</ Engine >
   <EngineVersion>2008r2</ EngineVersion >
   <RunningParameters>
    <DBInstanceParameter>
    <ParameterDescription> This option sets the default fill factor value for the server range. A fill f
actor is provided to optimize index data storage and performance. </ParameterDescription>
    <ParameterName>fill factor</ParameterName>
    <ParameterValue>0</ParameterValue>
    </DBInstanceParameter>
   </RunningParameters>
   <ConfigParameters>
    <DBInstanceParameter>
    <ParameterDescription> This option sets the default fill factor value for the server range. A fill f
actor is provided to optimize index data storage and performance. </ParameterDescription>
    <ParameterName>fill factor</ParameterName>
    <ParameterValue>50</ParameterValue>
   </DBInstanceParameter>
   </ConfigParameters>
</DescribeParametersResponse>
```

- JSON format

```
{
  "ConfigParameters": {
    "DBInstanceParameter": [
      {
        "ParameterDescription": "This option sets the default fill factor value for the server range.
A fill factor is provided to optimize index data storage and performance.",
        "ParameterName": "fill factor",
        "ParameterValue": "50"
      }
    ]
  },
  "Engine": "mssql",
  "EngineVersion": "2008r2",
  "RunningParameters": {
    "DBInstanceParameter": [
      {
        "ParameterDescription": "This option sets the default fill factor value for the server range.
A fill factor is provided to optimize index data storage and performance.",
        "ParameterName": "fill factor",
        "ParameterValue": "0"
      }
    ]
  },
  "RequestId": "2A748162-8040-4D6B-813E-6910C8C033F1"
}
```


3.2.8.3. ModifyParameter

Description

You can call this operation to modify instance parameters. After you submit a request, RDS assigns a task to apply new parameters to the instance. If there are any parameter modifications that require the database to be restarted, RDS restarts the database.

Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The instance is not locked by the system.

There are three types of parameter values:

- 1 to 65535: indicates a numerical range. Regular identification is used to extract the minimum and maximum values. The input parameter is then verified based on the extracted minimum and maximum values.

> **Notice** The parameter must be a multiple of a divisible factor.

- utf8 | gbk | latin1: indicates a set value determination rule. Regular identification is used to extract the fixed value. The input parameter is then verified based on these fixed values.
- Others: These parameters must be implemented with regular expressions.

Before assigning the task, RDS checks the parameter as follows:

1. Whether the parameter exists.
2. Whether the parameter can be modified.
3. Whether the parameter is valid.

   If the parameter is invalid, RDS returns the 400 HTTP status code and information about the invalid parameter. Examples:

```
{"HttpStatusCode":400,"Code":"InvalidParameter.Format",
"Message":"Specified parameter is not valid.[auto_increment_increment:a,character_set_client:41
]"}
```

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyParameter.
DBInstanceId	String	Yes	The ID of the instance.
Parameters	String	Yes	The JSON string for the parameter and its value. The parameter value must be of the string type. {"auto_increment_increment":"1", "character_set_client":"utf8"}.
Forcerestart	String	No	• true: specifies to forcibly restart the database. **Note** If a parameter can take effect only after database restart, Forcerestart must be set to true. Otherwise, the parameter modification will not take effect. • false: specifies not to forcibly restart the database. Default value: false.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=ModifyeParameter
&DBInstanceId=riauvj*****************201L
&Parameters={"key_buffer_size":"456","character_set_client":"gbk"}
&<Common request parameters>
```

Sample success responses

- XML format

```
<ModifyeParameterResponse>
    <RequestId>542****6-****-45CC-****-B0********0A</RequestId>
</ModifyeParameterResponse>
```

- JSON format

```
{
    "RequestId":"542****6-****-45CC-****-B0********0A",
}
```

3.2.9. Log management

3.2.9.1. DescribeSlowLogs

Description

You can call this operation to query the slow log summary of the databases in a specified instance over a specified period of time.

This operation supports paged query.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSlowLogs.
DBInstanceId	String	Yes	The ID of the instance.
StartTime	String	Yes	The beginning of the time range to query. Specify the date in the YYYY-MM-DDZ format. Example: 2011-05-30Z.
EndTime	String	Yes	The end of the time range to query. The end date must be later than the start date. Specify the date in the YYYY-MM-DDZ format. Example: 2011-05-30Z.
DBName	String	No	The name of the database.
SortKey	String	No	The basis by which query results are sorted. Valid values: TotalExecutionCounts: in the descending order of execution times. TotalQueryTimes: in the descending order of the total execution duration. TotalLogicalReads: in the descending order of the total number of logical reads.
PageSize	Integer	No	The number of entries to return on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNumber	Integer	No	The number of the page to return. Pages start from page 1. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information about common response parameters, see Common response parameters.
Engine	String	The engine of the database.
StartTime	String	The beginning of the date range where the query is performed. The date is in the YYYY-MM-DDZ format. Example: 2011-05-30Z.
EndTime	String	The end of the date range where the query is performed. The date is in the YYYY-MM-DDZ format. Example: 2011-05-30Z.

Parameter	Type	Description
TotalRecordCount	Integer	The total number of logs.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of SQL slow queries on the current page.
Items	List<SQLSlowLog>	

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeSlowLogs
&DBInstanceId=ria**********201
&StartTime=2011-06-11Z
&EndTime=2011-12-11Z
&SortKey= Tota**********Counts
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeSlowLogsResponse>
    <RequestId>A54****2-****-4BF3-****-0A*********E7</RequestId>
    <DBInstanceID>riauvjz********************449201</DBInstanceID>
    <Engine>MySQL</Engine>
    <StartTime>2011-06-11Z</StartTime>
    <EndTime>2011-12-11Z</EndTime>
    <TotalRecordCount>1</TotalRecordCount>
    <PageNumber>1</PageNumber>
    <PageRecordCount>1</PageRecordCount>
    <Items>
      <SQLSlowLog>
      <SQLText>update test.zxb set id=0 limit 1</SQLText>
      <SQLServerTotalExecutionCounts>178</SQLServerTotalExecutionCounts>
      <SQLServerTotalExecutionTimes>189</SQLServerTotalExecutionTimes>
      <TotalLogicalReadcounts>89</TotalLogicalReadcounts>
      <TotalPhysicalReadcounts>90</TotalPhysicalReadcounts>
      <ReportTime>2013-11-12Z</ReportTime>
     </SQLSlowLog>
    </Items>
</DescribeSlowLogsResponse>
```

- JSON format

```
{
    "RequestId":"A54****2-****-4BF3-****-0A*********E7"
"StartTime":"2011-06-11Z ",
"EndTime":"2011-12-11Z ",
"Engine":"MySQL",
"PageNumber":1,
"PageRecordCount":1,
"TotalRecordCount": 1,
"Items":
{"SQLSlowLog":
[
{ "SQLText":"update test.zxb set id=0 limit 1"
 "SQLServerTotalExecutionCounts":178
 "SQLServerTotalExecutionTimes":189
 "TotalLogicalReadcounts":89
 "TotalPhysicalReadcounts":90
 "ReportTime":"2013-11-12Z "
 }
]
}
}
```


3.2.9.2. DescribeSlowLogRecords

Description

You can call this operation to query the slow log details of the databases in a specified instance over a specified period of time. This operation is only supported on MySQL instances.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSlowLogRecords.
DBInstanceId	String	Yes	The ID of the instance.
StartTime	String	Yes	The beginning of the time range to query. Specify the time in the yyyy-MM-ddTHH:mmZ format. Example: 2011-06-11T15:00Z.

Parameter	Type	Required	Description
EndTime	String	Yes	The end of the time range to query. Specify the time in the yyyy-MM-ddTHH:mmZ format. Example: 2011-06-11T16:00Z. The end time must be later than the start time.
DBName	String	No	The name of the database.
PageSize	Integer	No	The number of entries to return on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNumber	Integer	No	The number of the page to return. Pages start from page 1. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information about common response parameters, see Common response parameters.
Engine	String	The type of the database.
TotalRecordCount	Integer	The total number of logs.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of SQL slow queries on the current page.
Items	List<SQLSlowRecord>	None

SQLSlowLog parameters

Parameter	Type	Description
HostAddress	String	The host address to connect to the database.
DBName	String	The name of the database.
SQLText	String	The query statement.
QueryTimes	Long	The execution time of the slow query. Unit: seconds.
LockTimes	Long	The lock time of the slow query. Unit: seconds.
ParseRowCounts	Long	The number of entries parsed.

Parameter	Type	Description
ReturnRowCounts	Long	The number of entries returned.
ExecutionStartTime	String	The beginning of the time range where the query was executed. The time is in the yyyy-MM-ddTHH:mm:ssZ format. Example: 2011-06-11T15:00:08Z.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeSlowLogRecords
&DBInstanceId=ria********************49201
&StartTime=2011-06-11T15:00Z
&EndTime=2011-06-11T16:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeSlowLogRecordsResponse>
    <RequestId>542****6-****-45CC-****-B0********30A</RequestId>
    <DBInstanceID>ria********************9201</DBInstanceID>
    <Engine>MySQL</Engine>
    <TotalRecordCount>1</TotalRecordCount>
    <PageNumber>1</PageNumber>
    <PageRecordCount>1</PageRecordCount>
    <Items>
     <SQLSlowRecord>
     <HostAddress>192.168.0.123</HostAddress>
     <DBName>test</DBName>
     <SQLText>update test.zxb set id=0 limit 1</SQLText>
     <QueryTimes>123</QueryTimes>
     <LockTimes>12</LockTimes>
     <ParseRowCounts>125</ParseRowCounts>
     <ReturnRowCounts>1</ReturnRowCounts>
     <ExecutionStartTime>2011-06-11T15:00:08Z</ExecutionStartTime>
     </SQLSlowRecord>
    </Items>
</DescribeSlowLogRecordsResponse>
```

- JSON format

```
{
 "RequestId":"542****6-****-45CC-****-B0********30A"
"Engine":"MySQL",
"PageNumber":1,
"PageRecordCount":1,
"TotalRecordCount":1
"Items":
{"SQLSlowRecord":
[
{
HostAddress:"192.168.0.123"
DBName:"test"
SQLText:" update test.zxb set id=0 limit 1"
QueryTimes:"123"
LockTimes"12"
ParseRowCounts:"125"
ReturnRowCounts:"1"
ExecutionStartTime:"2011-06-11T15:00:08Z"
}
]
}
}
```


3.2.9.3. DescribeErrorLogs

Description

You can call this operation to query the error logs generated for an instance over a specified period of time.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeErrorLogs.
DBInstanceId	String	Yes	The ID of the instance.
StartTime	String	Yes	The beginning of the time range to query. Specify the time in the yyyy-MM-ddTHH:mmZ format. Example: 2011-05-30T12:10Z.

Parameter	Type	Required	Description
EndTime	String	Yes	The end of the time range to query. Specify the time in the yyyy-MM-ddTHH:mmZ format. Example: 2011-05-30T20:10Z.
PageSize	Integer	No	The number of entries to return on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNumber	Integer	No	The number of the page to return. Pages start from page 1. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information about common response parameters, see Common response parameters.
TotalRecordCount	Integer	The total number of logs.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of error logs on the current page.
Items	List<ErrorLogs>	None

ErrorLogs parameters

Parameter	Type	Description
ErrorInfo	String	The error log information.
CreateTime	String	The time when the error log was generated. The time is in the yyyy-MM-ddTHH:mm:ssZ format. Example: 2011-05-30T12:11:4Z.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeErrorLogs
&DBInstanceId=ria**********201
&StartTime=2011-06-11T15:00Z
&EndTime=2011-06-11T20:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeErrorLogsResponse>
   <RequestId>985****7-****-40FC-****-E4********12C</RequestId>
   <TotalRecordCount>1</TotalRecordCount>
   <PageNumber>1</PageNumber>
   <PageRecordCount>1<PageRecordCount>
   <Items>
    <ErrorLog>
    <ErrorInfo>spid52 DBCC TRACEON 3499, server process ID (SPID) 52. This is an informational mess
age only; no user action is required</ErrorInfo>
    <CreateTime>2013-06-04T15:00:00</CreateTime>
    </ErrorLog>
  </Items>
</DescribeErrorLogsResponse>
```

- JSON format

```
{
"RequestId":"985****7-****-40FC-****-E4********12C"
"PageNumber":1,
"TotalRecordCount":1,
"PageRecordCount":1
"Items":
{"ErrorLog":
[
{
ErrorInfo:"spid52 DBCC TRACEON 3499, server process ID (SPID) 52. This is an informational message
only; no user action is required"
CreateTime:"2013-06-04T15:00:00"
}
]
}
}
```

3.2.9.4. DescribeBinlogFiles

Description

You can call this operation to query the binlog files of an instance on multiple pages. RDS provides binlog files for you to download.

- When DownloadLink is NULL, RDS does not provide a download URL.
- When DownloadLink is not NULL, you can use this URL to download the backup file through Wget (with double quotation marks), a browser, or a user program.

> **Note** The expiration time for this URL is specified by LinkExpiredTime. Use it before the expiration time.

If the download URL has expired, the following error code is displayed:

```
<? xml version="1.0" encoding="UTF-8" ? >
<Error>
<Code>AccessDenied</Code>
<Message>Request has expired. </Message>
<Expires>2012-12-25T09:47:52.000Z</Expires>
<ServerTime>2012-12-25T09:49:00.000Z</ServerTime>
<RequestId>50D9768CA801C2F102005C70</RequestId>
<HostId>oss-test.aliyun-inc.com</HostId>
</Error>
```

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeBinlogFiles.
DBInstanceId	String	Yes	The ID of the instance.
StartTime	String	Yes	The beginning of the time range to query the binlog files. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format.
EndTime	String	Yes	The end of the time range to query the binlog files. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format. The end time must be later than the start time.
PageSize	Integer	No	The number of entries to return on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNumber	Integer	No	The number of the page to return. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information, see Common response parameters.
TotalRecordCount	Integer	The total number of binlog files.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of binlog files on the current page.
Items	List<BinLogFile>	None

BinLogFile parameters

Parameter	Type	Description
FileSize	Long	The size of the binlog file. Unit: bytes.
LogBeginTime	String	The beginning of the time range where the binlog file was recorded.
LogEndTime	String	The end of the time range where the binlog file was recorded.
DownloadLink	String	The HTTP-compliant download URL. If the value is NULL, no download URL exists.
HostInstanceID	String	The ID of the instance to which the binlog belongs. This parameter indicates whether the binlog is from the primary or secondary instance.
LinkExpiredTime	String	The expiration time of the link. Example: 2011-06-11T15:00:00Z.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeBinlogFiles
&DBInstanceId=ria**********49201
&StartTime=2011-06-11T15:00:00Z
&EndTime=2013-06-05T15:00:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeBinlogFilesResponse>
   <RequestId>668****2-****-4C8D-****-253********A5</RequestId>
   <TotalRecordCount>1</TotalRecordCount>
   <PageNumber>1</PageNumber>
   <PageRecordCount>1<PageRecordCount>
   <Items>
    <BinLogFile>
    <FileSize>123</FileSize>
    <LogBeginTime>2013-06-01T15:00:00Z</LogBeginTime>
     <LogEndTime>2013-06-02T18:00:00Z</LogEndTime>
     <DownloadLink>http://rdslog-real.oss.aliyuncs.com/custins47742/hostins46770/mysql-bin.00012
0.zip</DownloadLink>
    <LinkExpiredTime>2013-06-09T18:00:00Z</LinkExpiredTime>
   </BinLogFile>
   </Items>
</DescribeBinlogFilesResponse>
```

- JSON format

```
{
    "RequestId":"66816822-CEC1-4C8D-AB26-2530A7D4DCA5"
"PageNumber":1,
"TotalRecordCount":1,
"PageRecordCount":1
"Items":
{"BinLogFile":
[
{
FileSize:"123"
LogBeginTime:"2013-06-01T15:00:00Z"
LogEndTime:"2013-06-02T18:00:00Z"
DownloadLink:"http://rdslog-real.oss.aliyuncs.com/custins47742/hostins46770/mysql-bin.000120.zip
"
LinkExpiredTime"2013-06-09T18:00:00Z"
}
]
}
}
```

3.2.9.5. DescribeSQLCollectorPolicy

Description

You can call this operation to check whether SQL collection is enabled for a specified RDS instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSQLCollectorPolicy.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information, see Common response parameters.
SQLCollectorStatus	String	• Enabled: specifies to enable SQL collection. • Disabled: specifies to disable SQL collection.

3.2.9.6. ModifySQLCollectorPolicy

Description

You can call this operation to enable or disable SQL collection for the specified RDS instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifySQLCollectorPolicy.
DBInstanceId	String	Yes	The ID of the instance.
SQLCollectorStatus	String	Yes	Enabled: specifies to enable SQL collection. Disabled: specifies to disable SQL collection.

Response parameters

For more information about common response parameters, see Common response parameters.

3.2.9.7. DescribeSQLLogRecords

Description

You can call this operation to query the SQL audit logs generated for an instance.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSQLLogRecords.
DBInstanceId	String	Yes	The ID of the instance.
Database	String	No	The name of the database to be queried. All databases are queried by default.
User	String	No	The username to be queried. All users are queried by default.
Form	String	No	• File: generates audit log files. Note If you set this parameter to File, only common response parameters will be returned. You need to call the DescribeSQLLogFiles operation to obtain the download URL of audit log files. • Stream: generates data streams. Default value: Stream.
QueryKeywords	String	No	The keywords used for query. Up to 10 keywords can be entered at a time. Separate multiple keywords with spaces.
StartTime	String	Yes	The beginning of the time range to query the logs. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format.
EndTime	String	Yes	The end of the time range to query the logs. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format. The end time must be later than the start time.
PageSize	Integer	No	The number of entries to return on each page. Valid value: 30, 50, 100, and 200. Default value: 100.
PageNumber	Integer	No	The number of the page to return. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	The common response parameters. For more information, see Common response parameters.
TotalRecordCount	Integer	The total number of SQL audit logs.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of SQL logs on the current page.
Items	List<SQLRecord>	None

SQLRecord parameters

Parameter	Type	Description
DBName	String	The name of the database.
AccountName	String	The name of the account.
HostAddress	String	The IP address of the client.
SQLText	String	The SQL statement recorded in the log.
TotalExecutionTimes	Long	The time consumed to execute the statement. Unit: milliseconds.
ReturnRowCounts	Long	The number of entries returned.
ThreadID	String	The ID of the thread.
ExecuteTime	String	The time when the statement is executed. It is in the yyyy-MM-ddTHH:mm:ssZ format. Example: 2011-05-30 T12:11:20Z.

Examples

Sample requests

```
https://rds.example.com/?Action=DescribeSQLLogRecords
&DBInstanceId=ria******************449201
&StartTime=2011-06-11T15:00:00Z
&EndTime=2013-06-05T15:00:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeSQLLogRecordsResponse>
   <RequestId>08A****B-****-4B03-****-CC*********28</RequestId>
   <TotalRecordCount>1</TotalRecordCount>
   <PageNumber>1</PageNumber>
   <PageRecordCount >1<PageRecordCount>
   <Items>
    <SQLRecord>
    <DBName>test</DBName>
    <AccountName>acc******test</AccountName>
    <HostAddress>192.168.0.121</HostAddress>
    <SQLText>update test.zxb set id=0 limit 1</SQLText>
    <TotalExecutionTimes >12</TotalExecutionTimes>
    <ReturnRowCounts>34</ReturnRowCounts>
    <ExecuteTime>2011-06-11T15:00:23Z</ExecuteTime>
    </SQLRecord>
  </Items>
</DescribeSQLLogRecordsResponse>
```

- JSON format

```
{
"PageNumber":1,
"TotalRecordCounts":1,
"ItemsCounts":1
"SQLItems":
{"SQLItem":
[
{
 "DBName":"test"
 "AccountName":"acc******st"
 "HostAddress":" 192.168.0.121"
"SQLText":"update test.zxb set id=0 limit 1"
"TotalExecutionTimes":12
"ReturnRowCounts":34
"ExecuteTime":"2011-06-11T15:00:23Z"
}
]
},
"RequestId": "08A*****-****-4B03-****-CC********828"
}
```


3.2.9.8. DescribeSQLLogFiles

Description

You can call this operation to query the list of SQL audit log files.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSQLLogFiles.
DBInstanceId	String	Yes	The ID of the instance.

Response parameters

Parameter	Type	Description
<Common response parameters>		The common response parameters. For more information, see Common response parameters.

Parameter	Type	Description
TotalRecordCount	Integer	The total number of SQL audit logs.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of entries on the current page.
Items	List<LogFile>	The array that consists of audit logs.

LogFile parameters

Parameter	Type	Description
FileID	Integer	The ID of an audit log file.
DBInstanceId	String	The ID of the instance.
LogStatus	String	• Success: The database has generated audit log files. • Failed: The database fails to generate audit log files. • Generating: The database is generating audit log files.
LogStartTime	String	The beginning of the time range where an audit log starts to record data.
LogEndTime	String	The end of the time range where an audit log stops recording data.
LogDownloadURL	String	The URL to download audit logs. If the download URL is unavailable, an empty string is returned.
LogSize	Long	The size of the log file. Unit: bytes.

3.2.10. Data migration

3.2.10.1. ImportDatabaseBetweenInstances

Description

You can call this operation to migrate data from other RDS instances. This operation is also supported on dedicated MySQL instances.

MySQL instances support batch database migration. During migration, the source instance status changes to Migrating and the destination instance status changes to Importing Data.

Database migration is performed between different instances that belong to the same user. Before you call this operation, ensure that the following requirements are met:

- The instance is in the running state.
- The database is in the running state.
- The destination instance is not locked by the system.

- The reserved storage space of the destination instance is greater than the space calculated by the formula: The used space of the destination instance - The database space of the destination instance + The database space of the source instance.
- Databases to be migrated exist in both the source and destination instances and are activated.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ImportDatabaseBetweenInstances.
DBInstanceId	String	Yes	The ID of the instance.
SourceDBInstanceId	String	Yes	The name of the source instance. It cannot be the same as that of the destination instance.
DBInfo	String	Yes	The database information to be migrated, in the JSON format. For MySQL instances, the parameter value is an array. Example: {"DBNames":["mydb","mydb2"]}. For MySQL instances, the names of the source and destination databases must be the same. In this example, data is migrated from the two databases mydb and mydb2. These databases must exist in both the source and destination instances.

DBInfo parameters

Parameter	Type	Required	Description
DBNames	List	Yes	The names of databases to be migrated. Example: ["mydb","mydb2"].

Response parameters

Parameter	Type	Description
<Common response parameters>		For more information about common response parameters, see Common response parameters.
ImportId	Integer	The ID of the destination instance.

Examples

Sample requests

```
https://rds.example.com/?Action=ImportDatabaseBetweenInstances
&DBInstanceId=ria**********9201
&SourceDBInstanceId=rdsm*********nqi
&{"DBNames":["mydb","mydb2"]}
&<Common request parameters>
```

Sample success responses

- XML format

```
<ImportDatabaseBetweenInstancesResponse>
        <ImportId>2122321</ImportId>
      <RequestId>5A7****0-****-4E08-****-59**********927</RequestId>
</ImportDatabaseBetweenInstancesResponse>
```

- JSON format

```
{
      "ImportId":2122321
      "RequestId":"5A7****0-****-4E08-****-59**********927"
}
```

3.2.10.2. CancelImport

Description

You can call this operation to cancel migration tasks from MySQL instances.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CancelImport.
DBInstanceId	String	Yes	The ID of the instance.
ImportId	Integer	Yes	The migration task ID.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.

Examples

Sample requests

```
https://rds.example.com/?Action=CancelImport
&DBInstanceId=rds************abn
&ImportId=12******23
&<Common request parameters>
```

Sample success responses

- XML format

```
<CancelImportResponse>
      <RequestId>17F****E-****-4337-****-9C*********D74</RequestId>
</CancelImportResponse>
```

- JSON format

```
{
   "RequestId": " 17F****E-****-4337-****-9C*********D74"
 }
```

3.2.11. Performance optimization

3.2.11.1. DescribeSQLLogReports

Description

You can call this operation to query the reports of SQL logs of an instance.

> Note We recommend that you set the query time on the hour such as 2 o'clock to 3 o'clock.

Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSQLLogReports.
DBInstanceId	String	Yes	The ID of the instance.

Parameter	Type	Required	Description
StartTime	String	Yes	The beginning of the time range to query. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format.
EndTime	String	Yes	The end of the time range to query. Specify the time in the yyyy-MM-ddTHH:mm:ssZ format. The end time must be later than the start time.
PageSize	Integer	No	The number of entries to return on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNumber	Integer	No	The number of the page to return. Pages start from page 1. Valid values: any non-zero positive integer. Default value: 1.

Response parameters

Parameter	Type	Description
<Common response parameters>	N/A	For more information about common response parameters, see Common response parameters.
TotalRecordCount	Integer	The total number of reports.
PageNumber	Integer	The page number of the returned page.
PageRecordCount	Integer	The number of reports on the current page.
Items	List<SQLRecord>	N/A

SQL log report parameters

Parameter	Type	Description
SQLText	String	The SQL statement.
TotalExecutionCounts	String	The total execution time.
TotalExecutionTimes	String	The total time consumed.
ReturnTotalRowCounts	String	The total number of entries returned.
TotalExecutionTimes	String	The total time consumed.

Examples

Sample requests

```
https://rds.aliyuncs.com/?Action= DescribeSQLReports
&DBInstanceId=ria**********9201
&StartTime=2011-06-11T15:00:00Z
&EndTime=2013-06-05T15:00:00Z
&<Common request parameters>
```

Sample success responses

- XML format

```
<DescribeSQLLogReportsResponse>
  <RequestId>08A****B-****-4B03-****-CC********828</RequestId>
  <TotalRecordCount>0</TotalRecordCount>
  <PageNumber>1</PageNumber>
  <PageRecordCount>30</PageRecordCount>
</DescribeSQLLogReportsResponse>
```

- JSON format

```
{
"PageNumber":1,
"TotalRecordCount":0,
"PageRecordCount":30,
"RequestId": "08A****B-****-4B03-****-CC********828"
}
```

3.2.12. RAM authorization

Use RAM for RDS resource authorization

All RDS instances created through an Apsara Stack tenant account are resources owned by that account. By default, you have full operation permissions on the resources under your account.

Resource Access Management (RAM) allows you to grant RAM users the access and management permissions on RDS resources owned by an Apsara Stack tenant account.

If you do not need to use RAM, ignore this topic.

Resource types in RAM

The only type of RDS resources that can be authorized through RAM is dbinstance. The following table lists the resource description method when the resources are authorized through RAM.

Resource	Resource description in authorization policy

Resource	Resource description in authorization policy
Instance	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid acs:rds:$regionid:$accountid:dbinstance/ *acs:rds::dbinstance/*

The $regionid field must be the ID of a region or an asterisk (*). The $instanceid field must be the ID of an instance or an asterisk (*). Similarly, $account-id is the numerical ID of your Apsara Stack tenant account, which can be replaced by an asterisk (*).

Authorization operations in RAM

In RAM, the following operations can be authorized to an RDS resource.

Operation 1	Operation 2
CreateDBInstance	ModifyDBInstanceSpec
DeleteDBInstance	CreateReadonlyDBInstance
DescribeDBInstances	DescribeDBInstanceAttribute
SwitchDBInstanceNetType	RestartDBInstance
ModifyDBInstanceDescription	ModifySecurityIps
ModifyDBInstanceMaintainTime	UpgradeDBInstanceEngineVersion
PurgeDBInstanceLog	CreateDatabase
DeleteDatabase	DescribeDatabases
ModifyDBDescription	ModifyeParameter
DescribeParameters	DescribeOptimizeAdviceByDBA
DescribeOptimizeAdviceOnMissIndex	ImportDataBaseBetweenInstances
CancelImport	CreateAccount
ResetAccountPassword	GrantAccountPrivilege
RevokeAccountPrivilege	DescribeAccounts
DeleteAccount	ModifyAccountDescription
CreateBackup	DescribeBackups
CreateTempDBInstance	DescribeBackupPolicy
ModifyBackupPolicy	DescribeResourceUsage

Operation 1	Operation 2
DescribeDBInstancePerformance	DescribeSlowLogs
DescribeSlowLogRecords	DescribeErrorLogs
DescribeBinlogFiles	DescribeSQLLogReports
DescribeSQLLogRecords	DescribeOptimizeAdviceOnStorage
DescribeOptimizeAdviceOnMissPK	DescribeOptimizeAdviceOnExcessIndex

> **Note** All operations except DescribeRegions and DescribeParameterTemplates can be authorized.

RAM authentication rules of API operations

When you call API operations to access resources as a RAM user, the RDS backend checks whether the RAM user is granted the required permissions by querying RAM. The permissions to be checked are determined by the resources used by the API operation. The following table lists the authentication rules for each API operation.

Operation	Authentication rule
CreateDBInstance	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DeleteDBInstance	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeDBInstances	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
SwitchDBInstanceNetType	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyDBInstanceDescription	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyDBInstanceMaintainTime	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
PurgeDBInstanceLog	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DeleteDatabase	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyDBDescription	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
CancelImport	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ResetAccountPassword	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
RevokeAccountPrivilege	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DeleteAccount	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
CreateBackup	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid

Operation	Authentication rule
CreateTempDBInstance	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyBackupPolicy	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeDBInstancePerformance	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeSlowLogRecords	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeBinlogFiles	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeSQLLogRecords	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeOptimizeAdviceOnMissPK	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeOptimizeAdviceOnMissIndex	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeParameters	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
CreatePrepaidDBInstanceForChannel	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyPrepaidDBInstanceSpec	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
CreatePostpaidDBInstanceForChannel	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyPostpaidDBInstanceSpec	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeDBInstanceAttribute	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
RestartDBInstance	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifySecurityIps	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
UpgradeDBInstanceEngineVersion	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
CreateDatabase	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeDatabases	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ImportDataBaseBetweenInstances	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
CreateAccount	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
GrantAccountPrivilege	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeAccounts	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyAccountDescription	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeBackups	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeBackupPolicy	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid

Operation	Authentication rule
DescribeResourceUsage	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeSlowLogs	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeErrorLogs	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeSQLLogReports	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeOptimizeAdviceOnStorage	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeOptimizeAdviceOnExcessIndex	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
DescribeOptimizeAdviceByDBA	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid
ModifyeParameter	acs:rds:$regionid:$accountid:dbinstance/$dbinstanceid

3.2.13. Appendix

3.2.13.1. Client error codes

3.2.13.1.1. Common errors

Error code	Error message	HTTP status code
MissingParameter	The input parameter "parameter name" that is mandatory for processing this request is not supplied.	400
InvalidParameter	The specified parameter "Action or Version" is not valid.	400
InvalidAccessKeyId.NotFound	The Access Key ID provided does not exist in our records.	404
IncompleteSignature	The request signature does not conform to Aliyun standards.	400
IllegalTimestamp	The input parameter "Timestamp" that is mandatory for processing this request is not supplied.	400
InvalidOwnerId	The specified OwnerId is not valid.	400
InvalidOwnerAccount	The specified OwnerAccount is not valid.	400
InvalidOwner	OwnerId and OwnerAccount can't be used at one API access.	400
Throttling	Request was denied due to request throttling.	400

Error code	Error message	HTTP status code
InvalidAction	The specified action is not valid.	403
ActionUnauthorized	The specified action is not available for you	403
UnsupportedHTTPMethod	This http method not supported.	403
UnsupportedParameter	The parameter "<parameter name>" is not supported.	400
Forbidden.InstanceNotFound	The specified Instance is not found.	404
Forbidden.RAM	User not authorized to operate on the specified resource, or this API doesn't support RAM.	403
Forbedden.NotSupportRAM	This action does not support accessed by RAM mode.	403
Forbidden.RiskControl	This operation is forbidden by Aliyun Risk Control system.	403
InsufficientBalance	Your account does not have enough balance.	400
Forbidden.Authentication	This operation is forbidden by Aliyun Realname Authentication system.	403
Invalid<parameter name>.ValueNotSupported	The specified parameter “parameter name” is not valid.	400
Invalid<parameter name>.Malformed	The specified parameter "parameter name n" is not valid.	400
InvalidParameter	The specified parameter " parameter name n " is not valid.	400

3.2.13.1.2. Specific errors

Error code	Error message	HTTP status code
InvalidRegionId.NotFound	The DBInstanceId provided does not exist in our records.	404
InvalidSecurityIPList.Duplicate	The Security IP address is not in the available range or occupied.	400
InvalidSecurityIPListLength.Malformed	The quota of security ip exceeds.	400

Error code	Error message	HTTP status code
InvalidEngineVersionInRegion.NotAvailable	The EngineVersion in the Region is not available.	403
InvaildEngineInRegion.NotAvailable	The Engine in the Region is not available.	403
InvaildDBInstanceClassInRegion.NotAvailable	The DBInstanceClass in the Region is not available.	403
OperationDenied	The resource is out of usage.	403
RegionUnauthorized	There is no authority to create instance in the specified region.	403
QuotaExceeded.CreateInstance	The quota of create instance exceeds.	403
InvalidDBInstanceId.NotFound	The DBInstanceId provided does not exist in our records.	404
MissingParameter	The input parameters "DBInstanceClass" and "DBInstanceStorage" all blank or no change.	400
OperationDenied.DBInstanceType	The operation is not permitted due to type of instance.	400
OperationDenied.DBInstanceStorageReduce	The operation is not permitted due to reduce storage.	403
OperationDenied.DBInstanceStatus	The operation is not permitted due to instance status.	403
DependencyViolation	The operation is not permitted due to instance mount read only or guard instance.	403
InvalidConnectionStringPrefix.Duplicate	The connection string is not in the available range or occupied.	400
QuotaExceeded.SwitchTime	The quota of switch time exceed.	400
OperationDenied.LockMode	The operation is not permitted due to lock of instance.	403
InvalidSecurityIps.Duplicate	The quota of security ip exceeds.	400
OperationDenied.CurrentEngineVersion	The operation is not permitted due to specified current version.	400
OperationDenied.ReadOnlyDBInstanceLower	The operation is not permitted due to higher read only instance version.	403

Error code	Error message	HTTP status code
OperationDenied.GuardDBInstanceLower	The operation is not permitted due to higher guard instance version.	403
OperationDenied.MasterDBInstanceState	The operation is not permitted due to status of master instance.	403
QuotaExceeded.DBName	The quota of db exceeds.	400
InvalidDBName.Duplicate	The Security db Name is not in the available range or occupied.	400
InvalidParameter.Keyword	The specified parameter "DBName" is keyword	400
OperationDenied.DBStatus	The operation is not permitted due to status of database.	403
OperationDenied.DBInstanceEngine	The operation is not permitted due to engine of instance.	403
FTPServiceUnavailable	The request has failed due to a temporary failure of the FTP server.	403
QuotaExceeded.Upload	The quota of file exceeds.	403
InvalidFileName.NotCompleted	The specified "FileName" is not completed.	404
InvalidFileName.NotAvailable	The specified "FileName" is not Available.	404
QuotaExceeded.FileSize	The quota of file size exceeds.	403
QuotaExceeded.DBInstanceStorage	The quota of instance storage exceeds.	403
InvalidDBInfo.Malformed	The specified parameter "DBInfo" is not valid or db not exist.	400
OperationDenied.Region	The operation is not permitted due to different region.	403
OperationDenied.Engine	The operation is not permitted due to different engine.	403
OperationDenied.DBInstanceId	The operation is not permitted due to DBInstanceId and SourceDBInstanceId is same.	403
InvalidTaskId.NotFound	The ImportId provided does not exist in our records.	400
OperationDenied.IncorrectTaskStatus	The operation is not permitted due to status of import.	403

Error code	Error message	HTTP status code
InvalidAccountName.keyword	The specified parameter "AccountName" is a keyword.	400
InvalidAccountName.Duplicate	The Security db Name is not in the available range or occupied.	400
QuotaExceeded.AccountName	The quota of account exceeds.	403
OperationDenied.AccountStatus	The operation is not permitted due to status of account.	403
InvalidDBName.NotFound	The DBName provided does not exist in our records.	404
OperationDenied.AccountType	The operation is not permitted due to type of account.	403
OperationDenied.BackupJobExists	The operation is not permitted due to backup job exist.	403
OperationDenied.NoDatabase	The operation is not permitted due to no database.	403
InvalidStartTimeAndEndTime.Malformed	The end time must be greater than the start time	400
OperationDenied.BackupJobExists	The operation is not permitted due to backup job exist.	403
InvalidBackupId.NotFound	The BackupId provided does not exist in our records.	404
OperationDenied.NotFoundBackup	The operation is not permitted due to no backup.	403
OperationDenied.BackupStatus	The operation is not permitted due to type of backup.	403
OperationDenied.BackupMethod	The operation is not permitted method of backup.	403
OperationDenied.TempDBInstanceExists	The operation is not permitted due to temp instance exist.	403
Forbidden.Authentication	The operation is forbidden by Aliyun Realname Authentication System.	403
IncorrectDBInstanceConnType	Current DB instance conn type does not support this operation.	404

Error code	Error message	HTTP status code
IncorrecttVpcId.Mismatch	The specified parameter "VpcId" is not valid.	400
IntranetNetTypeNotExists	Current DB instance net type does not support this operation.	404
InvalidDBInstanceName.NotFound	The DBInstanceId provided does not exist in our records.	403
InvalidConnectionMode.ValueNotSupported	The specified parameter "ConnectionMode" is not valid.	400
InvalidConnectionString.DuplicateSpecified	connection string already exists in the Aliyun RDS.	403
InvalidConnectionString.DuplicateSpecified	connection string or port want to be modified is the same with current net type.	400
InvalidConnectionStringPrefix.Duplicated	Specified connection string already exists in the Aliyun RDS.	403
InvalidCurrentConnectionString.NotFound	Specified current connection string doesn't exist.	400
InvalidDBInstanceId.NotFound	The DBInstanceId provided does not exist in our records.	400
InvalidDBInstanceId.QuotaExceeded	The quota of instance id exceeds.	400
InvalidDBInstanceNetType.NotFound	The Specified DB instance net type is not found.	404
InvalidDBinstanceNetType.ValueNotSupport	The specified parameter "DBinstanceNetType" is not valid.	400
InvalidDBInstanceType.ValueNotSupport	The specified parameter "DBInstanceType" is not valid.	400
InvalidEngineVersion.Malformed	The specified parameter "EngineVersion" is not valid.	400
InvalidInstanceNetworkType.ValueNotSupported	The specified parameter "InstanceNetworkType" is not valid.	400
InvalidParameter	The specified parameter "dbInstanceId" is not valid.	400
InvalidPort.Malformed	Specified port is not valid.	400
InvalidPrivateIpAddress.Duplicated	Specified private IP address is duplicated.	400

Error code	Error message	HTTP status code
InvalidPrivateIpAddress.Mismatch	Specified private IP address is not in the CIDR block of virtual switch.	400
InvalidVPCId.NotFound	Specified virtual vpc is not found.	400
InvalidVSwitchId.Mismatch	Specified instance and virtual switch are not in the same zone.	400
InvalidVSwitchId.NotFound	Specified virtual switch is not found in specified VPC.	400
NetTypeExists	Specified net type already existed.	400
NetTypeExists	Specified public net type already existed.	403
OperationDenied.DBInstanceNetType	The operation is not permitted due to net type of instance.	403
OperationDenied.DBInstanceStatus	The operation is not permitted due to status of instance.	403
OperationDenied.DBInstanceType	The operation is not permitted due to type of instance.	403
OperationDenied.LockMode	The operation is not permitted due to lock of instance.	403
OperationDenied.Switch	Specified instance cannot be switched.	400
OperationDenied.Switch	Specified instance cannot be switched to VPC.	403
OperationDenied.SwitchToVPC	Specified instance cannot be switched to VPC.	403
OperationDenied	The resource is out of usage.	403
OperationDenied	The vpc cloudinstanceIp is use, please check params.	406
OperationDenied	The vpc service is error, please check params.	400
PARAMETER_ERROR	The vpc request parameter may not be null.	406
PARAMETER_MUST_NOT_NULL	The parameter must not be null.	406
PublicConnectionExists	PublicConnection exists.	403
bpcIzNoNotEqualsRdsMultIzNo	Specified vpc IzNo not equal rds IzNo.	400
OperationDenied.DBInstanceStatus	The operation is not permitted due to status of instance.	403

Error code	Error message	HTTP status code
OperationDenied.DBInstanceType	The operation is not permitted due to type of instance.	403
OperationDenied.LockMode	The operation is not permitted due to lock of instance.	403
OperationDenied.Switch	Specified instance cannot be switched.	400
OperationDenied.SwitchToVPC	Specified instance cannot be switched to VPC.	403
OperationDenied	The resource is out of usage.	403
OperationDenied	The vpc cloudinstanceIp is use, please check params.	406
OperationDenied	The vpc service is error, please check params.	400
PARAMETER_ERROR	The vpc request parameter may not be null.	406
PARAMETER_MUST_NOT_NULL	The parameter must not be null.	406
PublicConnectionExists	PublicConnection exists.	403
bpcIzNoNotEqualsRdsMultIzNo	Specified vpc IzNo not equal rds IzNo.	400

3.2.13.2. Error codes on the server

Error code	Error message	HTTP status code
ServiceUnavailable	The request has failed due to a temporary failure of the server.	503
InternalError	The request processing has failed due to some unknown error, exception or failure.	500

3.2.13.3. Instance states

State	Description
Creating	An instance is being created.
Running	An instance is in the running state.
Deleting	An instance is being deleted.

State	Description
Rebooting	An instance is being restarted.
DBInstanceClassChanging	An instance is being upgraded or downgraded.
TRANSING	Data is being migrated.
EngineVersionUpgrading	Engine version is being upgraded.
TransingToOthers	Data is being migrated to other RDS instances.
GuardDBInstanceCreating	A disaster recovery instance is being created.
Restoring	A backup is being restored.
Importing	Data is being imported.
ImportingFromOthers	Data is being imported from other RDS instances.
DBInstanceNetTypeChanging	The network type is being switched between Intranet and Internet.
GuardSwitching	The instance is undergoing disaster recover failover.
INS_CLONING	The instance is being cloned.

3.2.13.4. Performance parameters

ApsaraDB RDS for MySQL

API parameter	Description
MySQL_NetworkTraffic	The average inbound traffic to a MySQL instance per second and average outbound traffic from a MySQL instance per second. Unit: KB.
MySQL_QPSTPS	The average number of SQL statements executed per second and average number of transactions per second.
MySQL_Sessions	The number of active connections and total number of current connections.
MySQL_InnoDBBufferRatio	The read hit rate of the InnoDB buffer pool, utilization of the InnoDB buffer pool, and percentage of dirty data blocks in the InnoDB buffer pool.
MySQL_InnoDBDataReadWriten	The average volume of data that InnoDB reads per second and average volume of data that InnoDB writes per second. Unit: KB.

API parameter	Description
MySQL_InnoDBLogRequests	The average number of read operations performed from the InnoDB buffer pool per second and average number of write operations performed on the InnoDB buffer pool per second.
MySQL_InnoDBLogWrites	The average number of log write requests per second, average number of physical writes performed on log files per second, and average number of fsync() writes completed on log files per second.
MySQL_TempDiskTableCreates	The number of temporary tables automatically created in the hard disk during MySQL statement execution.
MySQL_MyISAMKeyBufferRatio	The average Key Buffer utilization of MyISAM per second, average Key Buffer read hit rate of MyISAM per second, and average Key Buffer write hit rate of MyISAM per second.
MySQL_MyISAMKeyReadWrites	The average number of data reads from the buffer pool by MyISAM per second, average number of data writes to the buffer pool by MyISAM per second, average number of data reads from the hard disk by MyISAM per second, and average number of data writes to the hard disk by MyISAM per second.
MySQL_COMDML	The average number of DELETE statements executed per second, average number of INSERT statements executed per second, average number of INSERT_SELECT statements executed per second, average number of REPLACE statements executed per second, average number of REPLACE_SELECT statements executed per second, average number of SELECT statements executed per second, and average number of UPDATE statements executed per second.
MySQL_RowDML	The average number of rows read from the InnoDB table per second, average number of rows updated in the InnoDB table per second, average number of rows deleted from the InnoDB table per second, average number of rows inserted into the InnoDB table per second, and average number of physical writes to log files per second.
MySQL_MemCpuUsage	The CPU utilization and memory usage of a MySQL instance within the entire operating system.
MySQL_IOPS	The IOPS of a MySQL instance.
MySQL_DetailedSpaceUsage	The disk space occupied by a MySQL instance, including ins_size (total space), data_size (data space), log_size (log space), tmp_size (temporary space), and other_size (system space).
slavestat	The latency of read-only instances.

3.2.13.5. Forbidden keywords

ApsaraDB RDS for MySQL

Parameter field	Keyword description
DBName/ AccountName	admin, aurora, replicator, xtrabak, root(MySQL5.5/5.6), mysql, test, eagleye, information_schema, guest, add, analyze, asc, between, blob, call, change, check, condition, continue, cross, current_timestamp, database, day_microsecond, dec, default, desc, distinct, double, each, enclosed, exit, fetch, float8, foreign, goto, having, hour_minute, ignore, infile, insensitive, int1, int4, interval, iterate, keys, leading, like, lines, localtimestamp, longblob, low_priority, mediumint, minute_microsecond, modifies, no_write_to_binlog, on, optionally, out, precision, purge, read, references, rename, require, revoke, schema, select, set, spatial, sqlexception, sql_big_result, ssl, table, tinyblob, to, true, unique, update, using, utc_timestamp, varchar, when, with, xor, all, and, asensitive, bigint, both, cascade, char, collate, connection, convert, current_date, current_user, databases, day_minute, decimal, delayed, describe, distinctrow, drop, else, escaped, explain, float, for, from, grant, high_priority, hour_second, in, inner, insert, int2, int8, into, join, kill, leave, limit, load, lock, longtext, match, mediumtext, minute_second, natural, null, optimize, or, outer, primary, raid0, reads, regexp, repeat, restrict, right, schemas, sensitive, show, specific, sqlstate, sql_calc_found_rows, starting, terminated, tinyint, trailing, undo, unlock, usage, utc_date, values, varcharacter, where, write, year_month, alter, as, before, binary, by, case, character, column, constraint, create, current_time, cursor, day_hour, day_second, declare, delete, deterministic, div, dual, elseif, exists, false, float4, force, fulltext, group, hour_microsecond, if, index, inout, int, int3, integer, is, key, label, left, linear, localtime, long, loop, mediumblob, middleint, mod, not, numeric, option, order, outfile, procedure, range, real, release, replace, return, rlike, second_microsecond, separator, smallint, sql, sqlwarning, sql_small_result, straight_join, then, tinytext, trigger, union, unsigned, use, utc_time, varbinary, varying, while, x509, zerofill, galaxy

3.2.13.6. Instance types

Instances of different editions, versions, and types each perform differently from one another.

ApsaraDB RDS for MySQL instance types

Edition	Version	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Disk capacity	Zone-disaster recovery deployment
High-availability Edition	MySQL 5.6 supported	Shared instance	rds.mysql.t1.small	1 core, 1 GB	300	600	5 GB to 2 TB	Single-data center deployment Dual-data center deployment
			rds.mysql.s1.small	1 core, 2 GB	600	1,000		
			rds.mysql.s2.large	2 cores, 4 GB	1,200	2,000		
			rds.mysql.s2.xlarge	2 cores, 8 GB	2,000	4,000		
			rds.mysql.s3.large	4 cores, 8 GB	2,000	5,000		
			rds.mysql.m1.medium	4 cores, 16 GB	4,000	7,000		
			rds.mysql.c1.large	8 cores, 16 GB	4,000	8,000		
			rds.mysql.c1.xlarge	8 cores, 32 GB	8,000	12,000		
			rds.mysql.c2.xlarge	16 cores, 64 GB	16,000	14,000		
			rds.mysql.c2.xlp2	16 cores, 96 GB	24,000	16,000		
		Dedicated instance (X8)	mysql.x8.medium.2	2 cores, 16 GB	2,500	4,500	250 GB	
			mysql.x8.large.2	4 cores, 32 GB	5,000	9,000	500 GB	
			mysql.x8.xlarge.2	8 cores, 64 GB	10,000	18,000	1 TB	
			mysql.x8.2xlarge.2	16 cores, 128 GB	20,000	36,000	2 TB	
			mysql.x4.large.2	4 cores, 16 GB	2,500	4,500	250 GB	
			mysql.x4.xlarge.2	8 cores, 32 GB	5,000	9,000	500 GB	

Edition	Version	Dedicated instance family (X4)	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Disk capacity	Zone-disaster recovery deployment
			mysql.x4.2xlarge.2	16 cores, 64 GB	10,000	18,000	1 TB	
			mysql.x4.4xlarge.2	32 cores, 128 GB	20,000	36,000	2 TB	
		Dedicated host	rds.mysql.st.d13	30 cores, 220 GB	64,000	20,000	3 TB	
MySQL Enterprise Edition (three-node)	MySQL 5.6 supported	Dedicated instance (computing)	mysql.x4.large.3	4 cores, 16 GB	2,500	4,500	250 GB to 500 GB	Single-data center deployment
			mysql.x4.xlarge.3	8 cores, 32 GB	5,000	9,000	500 GB to 1 TB	
			mysql.x4.2xlarge.3	16 cores, 64 GB	10,000	18,000	1 TB to 2 TB	
			mysql.x4.4xlarge.3	32 cores, 128 GB	20,000	36,000	2 TB to 3 TB	
		Dedicated instance (with high memory)	mysql.x8.medium.3	2 cores, 16 GB	2,500	4,500	250 GB	
			mysql.x8.large.3	4 cores, 32 GB	5,000	9,000	500 GB	
			mysql.x8.xlarge.3	8 cores, 64 GB	10,000	18,000	1 TB	
			mysql.x8.2xlarge.3	16 cores, 128 GB	20,000	36,000	2 TB	
			mysql.x8.4xlarge.3	32 cores, 256 GB	40,000	72,000	3 TB	
MySQL		Dedicated	mysql.x8.medium.4	2 cores, 16 GB	2,500	4,500	250 GB	Dual-
			mysql.x8.large.4	4 cores, 32 GB	5,000	9,000	500 GB	
			mysql.x8.xlarge.4	8 cores, 64 GB	10,000	18,000	1 TB	

Enterprise Edition (four-node)	MySQL 5.6 supported version	Instance (with high memory family)	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Disk capacity	Dual-data center disaster recovery deployment
			mysql.x8.2xlarge.4	16 cores, 128 GB	20,000	36,000	2 TB	
			mysql.x8.4xlarge.4	32 cores, 256 GB	40,000	72,000	3 TB	
Read-only instance	MySQL 5.6 supported	General-purpose	rds.mysql.t1.small	1 core, 1 GB	300	600	5 GB to 2 TB	Single-data center deployment and dual-data center deployment
			rds.mysql.s1.small	1 core, 2 GB	600	1,000		
			rds.mysql.s2.large	2 cores, 4 GB	1,200	2,000		
			rds.mysql.s2.xlarge	2 cores, 8 GB	2,000	4,000		
			rds.mysql.s3.large	4 cores, 8 GB	2,000	5,000		
			rds.mysql.m1.medium	4 cores, 16 GB	4,000	7,000		
			rds.mysql.c1.large	8 cores, 16 GB	4,000	8,000		
			rds.mysql.c1.xlarge	8 cores, 32 GB	8,000	12,000		
			rds.mysql.c2.xlarge	16 cores, 64 GB	16,000	14,000		
			rds.mysql.c2.xlp2	16 cores, 96 GB	24,000	16,000		
		Dedicated instance (X8)	mysqlro.x8.medium.1	2 cores, 16 GB	2,500	4,500	250 GB	
			mysqlro.x8.large.1	4 cores, 32 GB	5,000	9,000	500 GB	
			mysqlro.x8.xlarge.1	8 cores, 64 GB	10,000	18,000	1 TB	
			mysqlro.x8.2xlarge.1	16 cores, 128 GB	20,000	36,000	2 TB	

Edition	Version	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Disk capacity	Zone-disaster recovery deployment
		Dedicated instance (X4)	mysqlro.x4.large.1	4 cores, 16 GB	2,500	4,500	250 GB	
			mysqlro.x4.xlarge.1	8 cores, 32 GB	5,000	9,000	500 GB	
			mysqlro.x4.2xlarge.1	16 cores, 64 GB	10,000	18,000	1 TB	
			mysqlro.x4.4xlarge.1	32 cores, 128 GB	20,000	36,000	2 TB	
		Dedicated host	rds.mysql.st.d13	30 cores, 220 GB	64,000	20,000	3 TB	

4.Data Transmission Service (DTS)

4.1. API reference

4.1.1. Call OpenAPI

4.1.1.1. List of operations by function

The following tables list API operations available for use in Data Transmission Service (DTS).

Data migration

Operation	Description
ConfigureMigrationJob	Configures a data migration task.
StartMigrationJob	Starts or restarts a data migration task.
SuspendMigrationJob	Pauses a data migration task.
StopMigrationJob	Stops a data migration task.
DeleteMigrationJob	Deletes a data migration task.
DescribeMigrationJobStatus	Queries the status of a data migration task.
DescribeMigrationJobDetail	Queries detailed information of a data migration task.
DescribeMigrationJobs	Queries the data migration tasks.

Change tracking

Operation	Description
CreateSubscriptionInstance	Purchases a change tracking instance.
ConfigureSubscriptionInstance	Configures a change tracking task.
StartSubscriptionInstance	Starts a change tracking task.
DeleteSubscriptionInstance	Deletes a change tracking task.
DescribeSubscriptionInstanceStatus	Queries the status of a change tracking task.
DescribeSubscriptionInstances	Queries the change tracking instances.
ModifySubscriptionObject	Modifies the objects for change tracking.
ModifyConsumptionTimestamp	Modifies the consumption checkpoint.

4.1.2. Data migration APIs

4.1.2.1. ConfigureMigrationJob

4.1.2.1.1. Overview

You can call this operation to configure a data migration task, including the source and destination instances and the objects to be migrated.

Before a migration task is started, you can call this operation repetitively to modify task configurations.

4.1.2.1.2. Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ConfigureMigrationJob.
MigrationJobId	String	Yes	The ID of the data migration task.
MigrationJobName	String	No	The name of the data migration task.
SourceEndpoint.InstanceType	String	Yes	The type of the source instance in the data migration task. Valid values: • LocalInstance: a user-created database in an on-premises data center

Parameter	Type	Required	Description
SourceEndpoint.Region	String	Yes	The region where the source instance resides. If the source instance is a user-created database, you can select the region that is closest to your data center. Valid values: • cn-hangzhou: Alibaba Cloud in the China (Hangzhou) region • cn-shanghai: Alibaba Cloud in the China (Shanghai) region • cn-qingdao: Alibaba Cloud in the China (Qingdao) region • cn-beijing: Alibaba Cloud in the China (Beijing) region • cn-zhangbei: Alibaba Cloud in the China (Zhangbei) region • cn-shenzhen: Alibaba Cloud in the China (Shenzhen) region • cn-hongkong: Alibaba Cloud in the China (Hong Kong) region • us-west-1: Alibaba Cloud in the US (Silicon Valley) region • us-east-1: Alibaba Cloud in the US (Virginia) region • ap-southeast-1: Alibaba Cloud in the Singapore (Singapore) region • cn-hangzhou-finance-1: Finance Cloud in the China (Hangzhou) region • cn-qingdao-finance-1: Finance Cloud in the China (Qingdao) region • cn-shanghai-finance-1: Finance Cloud in the China (Shanghai) region • cn-shenzhen-finance-1: Finance Cloud in the China (Shenzhen) region • me-east-1: Alibaba Cloud in the UAE (Dubai) region • en-central-1: Alibaba Cloud in the Germany (Frankfurt) region
SourceEndpoint.EngineName	String	No	The database type of the source instance. This parameter is required if the value of the SourceEndpoint.InstanceType parameter is not RDS. Valid values: MySQL.
SourceEndpoint.IP	String	No	The endpoint of the source instance. This parameter is required if the source instance is a user-created database.

Parameter	Type	Required	Description
SourceEndpoint.Port	String	No	The port on which the source instance is listening. This parameter is required if the source instance is a user-created database.
SourceEndpoint.UserName	String	Yes	The username that is used to access the source instance.
SourceEndpoint.Password	String	Yes	The password that is used to access the source instance.
DestinationEndpoint.InstanceType	String	Yes	The type of the destination instance in the data migration task. Valid values: • LocalInstance: a user-created database in an on-premises data center

Parameter	Type	Required	Description
DestinationEndpoint.Region	String	Yes	The region where the destination instance resides. If the destination instance is a user-created database, you can select the region that is closest to your data center. Valid values: • cn-hangzhou: Alibaba Cloud in the China (Hangzhou) region • cn-shanghai: Alibaba Cloud in the China (Shanghai) region • cn-qingdao: Alibaba Cloud in the China (Qingdao) region • cn-beijing: Alibaba Cloud in the China (Beijing) region • cn-zhangbei: Alibaba Cloud in the China (Zhangbei) region • cn-shenzhen: Alibaba Cloud in the China (Shenzhen) region • cn-hongkong: Alibaba Cloud in the China (Hong Kong) region • us-west-1: Alibaba Cloud in the US (Silicon Valley) region • us-east-1: Alibaba Cloud in the US (Virginia) region • ap-southeast-1: Alibaba Cloud in the Singapore (Singapore) region • cn-hangzhou-finance-1: Finance Cloud in the China (Hangzhou) region • cn-qingdao-finance-1: Finance Cloud in the China (Qingdao) region • cn-shanghai-finance-1: Finance Cloud in the China (Shanghai) region • cn-shenzhen-finance-1: Finance Cloud in the China (Shenzhen) region • me-east-1: Alibaba Cloud in the UAE (Dubai) region • en-central-1: Alibaba Cloud in the Germany (Frankfurt) region
DestinationEndpoint.EngineName	String	No	The database type of the destination instance. This parameter is required if the destination instance is a user-created database. Valid values: MySQL.
DestinationEndpoint.IP	String	No	The endpoint of the destination instance. This parameter is required if the destination instance is a user-created database.
DestinationEndpoint.Port	String	No	The port on which the destination instance is listening. This parameter is required if the destination instance is a user-created database.

Parameter	Type	Required	Description
DestinationEndpoint.UserName	String	Yes	The username that is used to access the destination instance.
DestinationEndpoint.Password	String	Yes	The password that is used to access the destination instance.
MigrationMode.StructureIntialization	Boolean	Yes	Specifies whether the data migration task performs schema migration. Valid values: • true: yes • false: no
MigrationMode.DataIntialization	Boolean	Yes	Specifies whether the data migration task performs full data migration. Valid values: • true: yes • false: no
MigrationMode.DataSynchronization	Boolean	Yes	Specifies whether the data migration task performs incremental data synchronization. Valid values: • true: yes • false: no
MigrationObject	List	Yes	The objects to be migrated. For more information, see the MigrationObject parameter definition topic.

4.1.2.1.3. MigrationObject parameter definitions

The MigrationObject parameters are defined as JSON strings. Certain regular expressions can be used in the object configuration.

The MigrationObject parameter definitions are as follows:

```
MigrationObject:{
    DBName:'The name of the database to be migrated',
    NewDBName:"The mapped name of the database to be migrated in the destination instance"
    SchemaName:'The name of the schema to be migrated',
    NewSchemaName:'The mapped name of the schema to be migrated in the destination instance',
    TableIncludes:[{
        TableName: 'The name of the table to be migrated',
        NewTableName:'The mapped name of the table to be migrated in the destination instance',
        FilterCondition:'where condition',
        ColumnIncludes:[{
            ColumnName:'The name of the column that needs to be migrated in the table',
            NewColumnName:'The mapped name of the column that needs to be migrated in the dest
ination instance'
        }]
        ColumnExcludes:[{
            ColumnName:'The name of the column that does not need to be migrated in the table'
        }]
    }]
    TableExcludes:[{                    — The table that does not need to be migrated
        TableName:'The name of the table that does not need to be migrated in the database'
    }]
}
```

SchemaName and NewSchemaName are applicable only to SQLServer. DBName and NewDBName are used to specify the names of other databases.

If you only specify DBName but do not specify lower levels of objects, all objects in the database are migrated.

If you specify DBName and some lower levels of objects, the objects that are not specified are not migrated by default.

You can use the following regular expressions to define different object names:

- A period (.) matches any single character except '\r\n'.
- An asterisk (*) matches zero or more occurrences of a subexpression which occurs before. h.*llo matches strings such as hllo and heeeello.
- A question mark (?) matches zero or one occurrence of a subexpression which occurs before. h.? llo matches hllo and hello, but not haello.
- The [characters] character set matches any character included in the braces. For example, h[aello] matches hallo and hello.
- The [^characters] negative character set does not match any character included in the braces. For example, h[^ae]llo matches hcllo and hdllo, but not hallo or hello.
- The [character1-character2] character range matches any character included in the range from character1 to character2, such as [0-9] and [a-z].

4.1.2.1.4. Response parameters

Name	Type	Description
Common response parameters	-	The request ID.

4.1.2.1.5. Example

The following example shows how to use the Java SDK to configure a migration task.

```
private static void configureMigrationJob(String MigrationJobId){
    String MigrationJobName="jiangliutest";
    /*Source Endpoint*/
    String SourceEndpoint_InstanceType="RDS";
    String SourceEndpoint_Region="cn-hangzhou";
    String SourceEndpoint_InstanceId="rm-xxxxxxxxxxxxxxxxx";
    String SourceEndpoint_UserName="*******";
    String SourceEndpoint_Password="*******";
    /*Destination Endpoint*/
    String DestinationEndpoint_InstanceType="RDS";
    String DestinationEndpoint_InstanceId="rm-xxxxxxxxxxxxxxxxx";
    String DestinationEndpoint_UserName="******";
    String DestinationEndpoint_Password="******";
    Boolean MigrationMode_StructureIntialization=true;
    Boolean MigrationMode_DataIntialization=true;
    Boolean MigrationMode_DataSynchronization=true;
    String MigrationObject="[{\"DBName\":\"jiangliutest\",\"NewDBName\":\"jiangliutest\","
        + "\"TableIncludes\":[{\"TableName\":\"t1\",\"NewTableName\":\"t1\"}]}]";
    ConfigureMigrationJobRequest request = new ConfigureMigrationJobRequest();
    request.setMigrationJobId(MigrationJobId);
    request.setMigrationJobName(MigrationJobName);
    request.setSourceEndpointRegion(SourceEndpoint_Region);
    request.setSourceEndpointInstanceType(SourceEndpoint_InstanceType);
    request.setSourceEndpointInstanceID(SourceEndpoint_InstanceId);
    request.setSourceEndpointUserName(SourceEndpoint_UserName);
    request.setSourceEndpointPassword(SourceEndpoint_Password);
    request.setDestinationEndpointInstanceType(DestinationEndpoint_InstanceType);
    request.setDestinationEndpointInstanceID(DestinationEndpoint_InstanceId);
    request.setDestinationEndpointUserName(DestinationEndpoint_UserName);
    request.setDestinationEndpointPassword(DestinationEndpoint_Password);
    request.setMigrationModeStructureIntialization(MigrationMode_StructureIntialization);
    request.setMigrationModeDataIntialization(MigrationMode_DataIntialization);
```

```
        request.setMigrationModeDataSynchronization(MigrationMode_DataSynchronization);
        request.setMigrationObject(MigrationObject);
        try {
            ConfigureMigrationJobResponse response = client.getAcsResponse(request);
            System.out.println("Configure "+ MigrationJobId+ " Succeed");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Configure "+MigrationJobId + " Failed!") ;
            System.out.println(e.toString());
        }
    }
```

4.1.2.2. CreateMigrationJob

4.1.2.2.1. Overview

You can call this operation to purchase a database migration instance. After the purchase, you need to call the ConfigureMigrationJob operation to configure a migration task.

4.1.2.2.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateMigrationJob.
Region	String	Yes	The region of the migration instance. Valid values: • cn-hangzhou: Alibaba Cloud in Hangzhou • cn-shanghai: Alibaba Cloud in Shanghai • cn-qingdao: Alibaba Cloud in Qingdao • cn-beijing: Alibaba Cloud in Beijing • cn-zhangbei: Alibaba Cloud in Zhangbei • cn-shenzhen: Alibaba Cloud in Shenzhen • cn-hongkong: Alibaba Cloud in Hong Kong • us-west-1: Alibaba Cloud in Western US • us-east-1: Alibaba Cloud in Eastern US • ap-southeast-1: Alibaba Cloud in Singapore • cn-hangzhou-finance-1: Finance Cloud in Hangzhou • cn-qingdao-finance-1: Finance Cloud in Qingdao • cn-shanghai-finance-1: Finance Cloud in Shanghai • cn-shenzhen-finance-1: Finance Cloud in Shenzhen • me-east-1: Alibaba Cloud in Dubai • en-central-1: Alibaba Cloud in Frankfurt

Name	Type	Required	Description
MigrationJobClass	String	Yes	The migration task type. Valid values: small, medium, and large. For the testing performance of each instance type, see Data migration types.
ClientToken	String	No	The token used to ensure the idempotence of the request. It can be any string.
OwnerID	String	No	If a RAM user is used to call this operation, the value of OwnerID is the UID of the Apsara Stack tenant account to which this RAM user belongs.

4.1.2.2.3. Response parameters

Name	Type	Description
MigrationJobId	String	The migration task ID.

4.1.2.2.4. Examples

The following code provides an example of how to use the Java SDK to purchase a migration instance.

```
private static String createMigrationJob(String region,String migrationJobClass) {
        CreateMigrationJobRequest request=new CreateMigrationJobRequest();
        request.setRegion("cn-hangzhou");
        request.setMigrationJobClass("small");
        try {
           CreateMigrationJobResponse response = client.getAcsResponse(request);
           String MigrationJobId = response.getMigrationJobId();
           System.out.println("Create Migration Job succeed. The Migration Job Id is "+MigrationJobId);
           return MigrationJobId;
        } catch (Exception e) {
           // TODO: handle exception
           System.out.println("Create Migration Job failed!" +e.toString());
           return null;
        }
     }
```

4.1.2.3. DeletetMigrationJob

4.1.2.3.1. Overview

You can call this operation to delete a migration task regardless of its status. After a migration task is deleted, you cannot query or manage it through OpenAPI or the console.

4.1.2.3.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DeletetMigrationJob.
MigrationJobId	String	Yes	The migration task ID.

4.1.2.3.3. Response parameters

Name	Type	Description
Common response parameters	-	The request ID.

4.1.2.3.4. Examples

The following code provides an example of how to use the Java SDK to delete a migration task.

```
private static void deleteMigrationJob(String MigrationJobId) {
      DeleteMigrationJobRequest request = new DeleteMigrationJobRequest();
      DeleteMigrationJobResponse response = new DeleteMigrationJobResponse();
      request.setMigrationJobId(MigrationJobId);
      try {
         response = client.getAcsResponse(request);
         System.out.println("Delete Migration Job "+MigrationJobId+" Succeed!") ;
      } catch (Exception e) {
         // TODO: handle exception
         System.out.println("Delete Migration Job "+MigrationJobId+" Failed!") ;
         System.out.println(e.toString());
      }
   }
```

4.1.2.4. DescribeMigrationJobDetail

4.1.2.4.1. Overview

When a migration task is in the Migrating, MigrationFailed, Suspending, or Finished status, you can call this operation to query data migration details, including the migration status and progress of each object.

4.1.2.4.2. Request parameters

Description	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeMigrationJobDetail.
MigrationJobId	String	Yes	The migration task ID.
MigrationMode.StructureInitialization	Boolean	No	The schema migration details.
MigrationMode.DataInitialization	Boolean	No	The full data migration details.
MigrationMode.DataSynchronization	Boolean	No	The incremental data migration details.

Description	Type	Required	Description
PageSize	Integer	No	• The number of records on each page. Valid values: 30, 50, and 100. Default value: 30. • This parameter is required when only one of these three parameters is set to true: MigrationMode.StructureInitialization, MigrationMode.DataInitialization, and MigrationMode.DataSynchronization.
PageNum	Integer	No	• The page number. It must be greater than 0 and smaller than or equal to the maximum value of Integer. Default value: 1. • This parameter is required when only one of these three parameters is set to true: MigrationMode.StructureInitialization, MigrationMode.DataInitialization, and MigrationMode.DataSynchronization.

4.1.2.4.3. Response parameters

Name	Type	Description
StructureInitializationDetails	List	The schema migration details. For more information, see StructureInitializationDetail parameter definitions.
DataInitializationDetails	List	The full data migration details. For more information, see DataInitializationDetail parameter definitions.

Name	Type	Description
DataSynchronizationDetails	List	The incremental data migration details. For more information, see DataSynchronizationDetail parameter definitions.

4.1.2.4.4. StructureInitializationDetail parameter definitions

Name	Type	Description
ObjectName	String	The name of the object to be migrated.
ObjectType	String	The type of the object to be migrated. Valid values: Table, Constraint, Index, View, MaterializeView, Type, Synonym, Trigger, Function, Procedure, Package, Default, Rule, and Plan Guide.
SourceOwnerDBName	String	The name of the source database from which the object is to be migrated.
DestinationOwnerDBName	String	The schema of the destination database to which the object is to be migrated.
Status	String	The migration status of the object to be migrated.
ErrorMessage	String	The reason why the object failed to be migrated.
ObjectDefinition	String	The schema definition of the object to be migrated.
Constraints	List	Constraints in which the object to be migrated participates. Constraints are active only when the object to be migrated is a table. The list of constraints includes indexes and foreign keys.

4.1.2.4.5. DataInitializationDetail parameter definitions

Name	Type	Description
TableName	String	The name of the table to be migrated.
SourceOwnerDBName	String	The name of the source database from which the table is to be migrated.
DestinationOwnerDBName	String	The name of the destination database to which the table is to be migrated.
Status	String	The migration status of the table.
ErrorMessage	String	The reason why the table failed to be migrated.
TotalRowNum	String	The total number of records to be migrated in the table.
FinishRowNum	String	The number of records that have been migrated in the table.
UsedTime	String	The time that has been spent on migrating the table. Unit: second.

4.1.2.4.6. DataSynchronizationDetail parameter definitions

Name	Type	Description
TableName	String	The name of the table to be migrated.
SourceOwnerDBName	String	The name of the source database from which the table is to be migrated.
DestinationOwnerDBName	String	The name of the destination database to which the table is to be migrated.
Status	String	The migration status of the table to be migrated.
ErrorMessage	String	The reason why the table failed to be migrated.

4.1.2.4.7. Examples

The following code provides an example of how to use the Java SDK to query migration task details.

```
private static void describeMigrationJobDetail(String MigrationJobId) {
        DescribeMigrationJobDetailRequest request = new DescribeMigrationJobDetailRequest();
        request.setMigrationJobId(MigrationJobId);
        request.setMigrationModeStructureInitialization(true);
        request.setMigrationModeDataInitialization(true);
        request.setMigrationModeDataSynchronization(true);
        try {
            DescribeMigrationJobDetailResponse response = client.getAcsResponse(request);
            System.out.println("Get Migration Detail of Migration Job " +MigrationJobId+" Succeed!") ;
            List<StructureInitializationDetail> struInitDetails= response.getStructureInitializationDetailList
();
            System.out.println("Structure Initialization Detail:");
            for (int i = 0; i < struInitDetails.size(); i++) {
                String objectName=struInitDetails.get(i).getObjectName();
                String SrcDBName=struInitDetails.get(i).getSourceOwnerDBName();
                String DestDBName = struInitDetails.get(i).getDestinationOwnerDBName();
                String Status=struInitDetails.get(i).getStatus();
                String errorMsg=struInitDetails.get(i).getErrorMessage();
                System.out.println("ObjectName:"+objectName+"; SRCDBName:"+SrcDBName+"; DESTDBName
:"
                        +DestDBName+"; Status:"+Status+"errorMsg:"+errorMsg);
            }
            List<DataInitializationDetail> dataInitDetails = response.getDataInitializationDetailList();
            System.out.println("Data Initialization Detail:");
            for (int i = 0; i < dataInitDetails.size(); i++) {
                String tblName=dataInitDetails.get(i).getTableName();
                String tbl_srcDBName=dataInitDetails.get(i).getSourceOwnerDBName();
                String tbl_destDBName=dataInitDetails.get(i).getDestinationOwnerDBName();
                String dataInitStatus=dataInitDetails.get(i).getStatus();
                String dataInitSumRows=dataInitDetails.get(i).getTotalRowNum();
                String dataInitFinishRows=dataInitDetails.get(i).getFinishRowNum();
                String dataUsedTime=dataInitDetails.get(i).getMigrationTime();
                String dataInitErrMSG = dataInitDetails.get(i).getErrorMessage();
                System.out.println("TableName:"+tblName+"; SRCDBName:"+tbl_srcDBName
                        +"; DESTDBName:"+tbl_destDBName+"; SumRows:"+dataInitSumRows+"; FinishRows"
                        +"; UsedTime:"+dataUsedTime+"; ErrorMSG:"+dataInitErrMSG);
```

```
        }
        List<DataSynchronizationDetail> dataSyncDetails = response.getDataSynchronizationDetailList
();
        System.out.println("Data Synchronization Detail");
        for (int i = 0; i < dataSyncDetails.size(); i++) {
          String dataSyncTblName=dataSyncDetails.get(i).getTableName();
          String dataSyncSRCDBName=dataSyncDetails.get(i).getSourceOwnerDBName();
          String dataSyncDESTDBName = dataSyncDetails.get(i).getDestinationOwnerDBName();
          String dataSyncStatus = dataSyncDetails.get(i).getStatus();
          String dataSyncErrMsg = dataSyncDetails.get(i).getErrorMessage();
          System.out.println("TableName:"+dataSyncTblName+"; SRCDBName:"+dataSyncSRCDBName
              +"; DESTDBName:"+dataSyncDESTDBName+"; Status:"+dataSyncStatus+"; ErrorMSG:"+d
ataSyncErrMsg);
        }
      } catch (Exception e) {
        // TODO: handle exception
        System.out.println("Get Migration Detail of Migration Job " +MigrationJobId+" Failed!") ;
        System.out.println(e.toString());
      }
    }
```

4.1.2.5. DescribeMigrationJobStatus

4.1.2.5.1. Overview

You can call this operation to query the data migration status regardless of the migration task status. You can query the overall migration status of the task, the migration status of each migration stage (schema migration, full data migration, and incremental data migration), and the precheck status.

4.1.2.5.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeMigrationJobStatus.
MigrationJobId	String	Yes	The migration task ID.

4.1.2.5.3. Response parameters

Name	Type	Description
MigrationJobId	String	The ID of the migration task.
MigrationJobName	String	The name of the migration task.
PayType	String	The billing method.
MigrationJobClass	String	The type of the migration instance.
SourceEndpoint	Endpoint	The IP address for connecting to the source instance. For more information, see the parameter definition in section 1.2.5.4.
DestinationEndpoint	Endpoint	The IP address for connecting to the destination instance. For more information, see the parameter definition in section 1.2.5.4.
MigrationMode	MigrationMode	The migration type. For more information, see the parameter definition in section 1.2.5.5.
MigrationObjects	List	The objects to be migrated.
MigrationJobStatus	String	The status of the migration task. Valid values: • NotStarted: The migration task is not started. • Prechecking: The migration task is being prechecked. • PrecheckFailed: The precheck fails. • Migrating: Data is being migrated. • Suspending: The migration task is suspended. • MigrationFailed: The migration task fails. • Finished: The migration task is complete.
PrecheckStatus	PrecheckStatus	The precheck status including the check progress and the specific state. For more information, see the parameter definition in section 1.2.5.6.

Name	Type	Description
StructureInitializationStatus	StructureInitializationStatus	The status of schema migration including the migration progress and the specific state. For more information, see the parameter definition in section 1.2.5.8.
DataInitializationStatus	DataInitializationStatus	The status of full data migration including the migration progress and the specific state. For more information, see the parameter definition in section 1.2.5.9.
DataSynchronizationStatus	DataSynchronizationStatus	The status of incremental data migration including the migration progress and the specific state. For more information, see the parameter definition in section 1.2.5.10.

4.1.2.5.4. Endpoint parameter definitions

Name	Type	Description
InstanceType	String	The instance type.
Region	String	The region of the source instance.
InstanceId	String	The instance ID.
EngineName	String	The database type.
IP	String	The database connection address.
Port	String	The listening port of the database.
UserName	String	The username to access the database.
OracleSID	String	The Oracle SID.
DatabaseName	String	The database name.

4.1.2.5.5. MigrationMode parameter definitions

Name	Type	Description
StructureInitialization	String	Indicates whether to perform schema migration.
DataInitialization	String	Indicates whether to perform full data migration.
DataSynchronization	String	Indicates whether to perform incremental data migration.

4.1.2.5.6. PrecheckStatus parameter definitions

Name	Type	Description
Status	String	The precheck status.
Percent	String	The precheck progress. Unit: %.
Detail	List	The check details of each precheck item. For more information, see CheckItem parameter definitions.

4.1.2.5.7. CheckItem parameter definitions

Name	Type	Description
ItemName	String	The check item name.
CheckStatus	String	The check status. Valid values: • NotStarted • Checking • Failed • Finished
ErrorMessage	String	The cause of the error.
RepairMethod	String	The repair method.

4.1.2.5.8. StructureInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.2.5.9. DataInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.2.5.10. DataSynchronizationStatus parameter definitions

Name	Type	Description

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Delay	String	The migration latency.

4.1.2.5.11. Examples

The following code provides an example of how to use the Java SDK to query the status of a migration task.

```
private static void describeMigrationJobStatus(String MigrationJobId){
    DescribeMigrationJobStatusRequest request = new DescribeMigrationJobStatusRequest();
    request.setMigrationJobId(MigrationJobId);
    try {
      DescribeMigrationJobStatusResponse response = client.getAcsResponse(request);
      System.out.println("Get Status of Migration Job "+MigrationJobId+" Succeed!") ;
      String JobStatus=response.getMigrationJobStatus();
      /*Get the basic information of the migration job*/
      System.out.println("MigrationJobName:"+response.getMigrationJobName()
          +"MigrationJobId:"+response.getMigrationJobId()
          +"MigrationObjects:"+response.getMigrationObject()
          +"; MigrationMode.StructureInitialization:" + response.getMigrationMode().getStructureInit
ialization()
          +"; MigrationMode.DataInitialization:"+response.getMigrationMode().getDataInitialization(
)
          +"; MigrationMode.DataSynchronization:"+response.getMigrationMode().getDataSynchroni
zation()
          +"; MigrationJobStatus:"+response.getMigrationJobStatus());
      /*Get the source endpoint of the migration job*/
      System.out.println("SourceEndpoint.InstanceType:"+response.getSourceEndpoint().getInstance
Type()
          +"; SourceEndpoint.InstanceId:"+response.getSourceEndpoint().getInstanceId()
```

```
            +"; SourceEndpoint.EngineName:"+response.getSourceEndpoint().getEngineName()
            +"; SourceEndpoint.IP:"+response.getSourceEndpoint().getIP()
            +"; SourceEndpoint.Port"+response.getSourceEndpoint().getPort()
            +"; SourceEndpoint.UserName:"+response.getSourceEndpoint().getUserName()
            +"; SourceEndpoint.DatabaseName:"+response.getSourceEndpoint().getDatabaseName()
            +"; SourceEndpoint.OracleSID:"+response.getSourceEndpoint().getOracleSID());
        /*Get the destination endpoint of the migration job*/
        System.out.println("DestinationEnpoint.InstanceType:"+response.getDestinationEndpoint().get
InstanceType()
            +"DestinationEndpoint.InstanceId:"+response.getDestinationEndpoint().getInstanceId()
            +"DestinationEndpint.EngineName:"+response.getDestinationEndpoint().getEngineName()
            +"DestinationEndpoint.IP:"+response.getDestinationEndpoint().getIP()
            +"DestinationEndpoint:Port:"+response.getDestinationEndpoint().getPort()
            +"DestinationEndpoint.UserName:"+response.getDestinationEndpoint().getUserName()
            +"DestinationEndpoint.DatabaseName"+response.getDestinationEndpoint().getDatabaseN
ame()
            +"DestinationEndpoint.OracleSID:"+response.getDestinationEndpoint().getOracleSID());
        /*Get Status of Precheck*/
        try {
          com.aliyuncs.dts.model.v20160801. DescribeMigrationJobStatusResponse.PrecheckStatus prec
heckStatus = response.getPrecheckStatus();
          System.out.println("Get  Precheck Status of Migration Job "+MigrationJobId +" Succeed!") ;
          System.out.println("Precheck Status:"+ precheckStatus.getStatus()+";Precheck Percent:"+pr
echeckStatus.getPercent());
          List<CheckItem> checkItems = precheckStatus.getDetail();
          for (int i = 0; i < checkItems.size(); i++) {
            String checkItem=checkItems.get(i).getItemName();
            String checkStatus = checkItems.get(i).getCheckStatus();
            String repairMethod = checkItems.get(i).getRepairMethod();
            String errorMessage =checkItems.get(i).getErrorMessage();
            System.out.println("CheckItem:"+checkItem+"; CheckStatus:"+checkStatus
                +"errorMessage:"+errorMessage+"; repairMethod:"+repairMethod);
          }
        } catch (Exception e) {
          // TODO: handle exception
          System.out.println("Get  Precheck Status of Migration Job "+MigrationJobId +" Failed!") ;
          System.out.println(e.toString());
        }
        /*Get Status of Structure Migration*/
        try {
          StructureInitializationStatus strucInitStatus = new StructureInitializationStatus();
```

```
            strucInitStatus = response.getStructureInitializationStatus();
            String status = strucInitStatus.getStatus();
            String percent = strucInitStatus.getPercent();
            String progress = strucInitStatus.getProgress();
            String errorMsg = strucInitStatus.getErrorMessage();
            System.out.println("Get Structure Initialization Status of Migration Job " +MigrationJobId+" Su
cceed!") ;
            System.out.println("Status:"+status+";percent:"+percent+"; progress:"+progress+"; errorMs
g:"+errorMsg);
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Get Structure Initialization Status of Migration Job " +MigrationJobId+" Fai
led!") ;
            System.out.println(e.toString());
        }
        /*Get Status of Data Migration*/
        try {
            DataInitializationStatus dataInitStatus = new DataInitializationStatus();
            dataInitStatus = response.getDataInitializationStatus();
            String datainitstatus=dataInitStatus.getStatus();
            String datainitpercent = dataInitStatus.getPercent();
            String datainitprogress = dataInitStatus.getProgress();
            String datainiterrormsg = dataInitStatus.getErrorMessage();
            System.out.println("Get Data Initialization Status of Migration Job " +MigrationJobId+" Succee
d!") ;
            System.out.println("Status:"+datainitstatus+";percent:"+datainitpercent+"; progress:"+datai
nitprogress+"; errorMsg:"+datainiterrormsg);
         } catch (Exception e) {
            // TODO: handle exception
             System.out.println("Get Data Initialization Status of Migration Job " +MigrationJobId+" Failed!
") ;
             System.out.println(e.toString());
        }
        /*Get Status of Data Synchronization*/
        try {
            DataSynchronizationStatus dataSyncStatus = new DataSynchronizationStatus();
            dataSyncStatus = response.getDataSynchronizationStatus();
            String datasynstatus=dataSyncStatus.getStatus();
            String datasyncpercent=dataSyncStatus.getPercent();
            String datasyncdelay=dataSyncStatus.getDelay();
            String datasyncerrormsg=dataSyncStatus.getErrorMessage();
```

```
            String datasyncerrormsg=datasyncstatus.getErrorMessage();
            System.out.println("Get Data Synchronization Status of Migration Job " +MigrationJobId+" Succeed!") ;
            System.out.println("Status:"+datasynstatus+";percent:"+datasyncpercent+"; delay:"+datasyncdelay+"; errorMsg:"+datasyncerrormsg);
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Get Data Synchronization Status of Migration Job " +MigrationJobId+" Failed!") ;
            System.out.println(e.toString());
        }
    } catch (Exception e) {
        // TODO: handle exception
        System.out.println("Get Status of Migration Job "+MigrationJobId+" Failed");
        System.out.println(e.toString());
    }
  }
```

4.1.2.6. StartMigrationJob

4.1.2.6.1. Overview

You can call this operation to start a migration task. When a migration task is in the NotStarted or MigrationFailed status, you can call this operation to start or restart it.

4.1.2.6.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to StartMigrationJob.
MigrationJobId	String	Yes	The migration task ID.

4.1.2.6.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.2.6.4. Examples

The following code provides an example of how to use the Java SDK to start a migration task.

```
private static void startMigrationJob(String MigrationJobId){
    StartMigrationJobRequest request = new StartMigrationJobRequest();
    request.setMigrationJobId(MigrationJobId);
    try {
       StartMigrationJobResponse response = client.getAcsResponse(request);
       System.out.println("Start "+MigrationJobId+" Succeed");
    } catch (Exception e) {
       // TODO: handle exception
       System.out.println("Start "+MigrationJobId + " Failed");
       System.out.println(e.toString());
    }
  }
```

4.1.2.7. StopMigrationJob

4.1.2.7.1. Overview

You can call this operation to stop a migration task when it is in the **Migrating** status.

> **Note** You cannot start a task that has been stopped.

4.1.2.7.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to StopMigrationJob.
MigrationJobId	String	Yes	The migration task ID.

4.1.2.7.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.2.7.4. Examples

The following code provides an example of how to use the Java SDK to stop a migration task.

```
private static void stopMigrationJob(String MigrationJobId) {
    StopMigrationJobRequest request = new StopMigrationJobRequest();
    try {
      request.setMigrationJobId(MigrationJobId);
      StopMigrationJobResponse response = client.getAcsResponse(request);
      System.out.println("Stop Migration Job "+MigrationJobId+" Succeed!") ;
    } catch (Exception e) {
      // TODO: handle exception
      System.out.println("Stop Migration Job "+MigrationJobId+" Failed!") ;
      System.out.println(e.toString());
    }
  }
```

4.1.2.8. SuspendMigrationJob

4.1.2.8.1. Overview

You can call this operation to suspend a migration task when it is in the Migrating status. Note that if a task has been suspended for more than 6 hours after it enters the incremental data migration stage, you may fail to restart it through the StartMigrationJob operation.

4.1.2.8.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to SuspendMigrationJob.
MigrationJobId	String	Yes	The migration task ID.

4.1.2.8.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.2.8.4. Examples

The following code provides an example of how to use the Java SDK to suspend a migration task.

```
private static void suspendMigrationJob(String MigrationJobId){
    SuspendMigrationJobRequest request = new SuspendMigrationJobRequest();
    request.setMigrationJobId(MigrationJobId);
    try {
       SuspendMigrationJobResponse response = client.getAcsResponse(request);
       System.out.println("Suspend "+ MigrationJobId + " Succeed");
    } catch (Exception e) {
       // TODO: handle exception
       System.out.println("Suspend "+MigrationJobId +"Failed");
       System.out.println(e.toString());
    }
  }
```

4.1.3. Data synchronization APIs

4.1.3.1. ConfigureSynchronizationJob

4.1.3.1.1. Overview

After you call the CreateSynchronizationJob operation to purchase a synchronization instance, you can call this operation to configure a synchronization task, provided that the synchronization task is not configured.

4.1.3.1.2. Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ConfigureSynchronizationJob.
SynchronizationJobId	String	Yes	The ID of the data synchronization task.
SynchronizationJobName	String	No	The name of the data synchronization task.
SourceEndpoint.InstanceId	String	Yes	The ID of the source instance.

Parameter	Type	Required	Description
SynchronizationDirection	String	No	The synchronization direction. This parameter is used only for two-way synchronization tasks. Valid values: Forward and Reverse. Default value: Forward.
SourceEndpoint.OwnerID	String	No	The ID of the Apsara Stack tenant account to which the source instance belongs. This parameter is available if the source and destination instances belong to different Apsara Stack tenant accounts.
SourceEndpoint.Role	String	No	The RAM role that the Apsara Stack tenant account of the source instance assigns to the Apsara Stack tenant account of the destination instance. This parameter is available if the source and destination instances belong to different Apsara Stack tenant accounts. For more information about role permissions and how to grant permissions to a role, see Configure RAM authorization for cross-account data migration and synchronization.
DestinationEndpoint.InstanceId	String	Yes	The ID of the destination instance. If the destination instance is an AnalyticDB instance, the parameter indicates the database name.
SynchronizationObjects	List	Yes	The objects that you want to synchronize.
StructureInitialization	Boolean	No	Specifies whether to perform initial schema synchronization. Valid values: • true: yes • false: no Default value: true.
DataInitialization	Boolean	No	Specifies whether to perform initial full data synchronization. Valid values: • true: yes • false: no Default value: true.
PartitionKey.ModifyTime_Day	Boolean	No	Specifies whether the incremental data table is partitioned by day. This parameter is available if the destination instance is a MaxCompute instance. Valid values: • true: The incremental data table supports daily partitions. • false: The incremental data table does not support daily partitions. Default value: true.

4.1.3.1.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.3.1.4. Example

This topic describes how to use the Java SDK to configure a synchronization task.

```
public void configureSyncJobRDS2RDS(String SyncJobId,DefaultAcsClient client) {
    ConfigureSynchronizationJobRequest request = new ConfigureSynchronizationJobRequest();
    request.setSynchronizationJobId(SyncJobId);
    request.setSynchronizationJobName("jiangliu_test_rds2rds_sync");
    request.setSourceEndpointInstanceId("rm-axxxxxxxxxxxxxxxxx");
    request.setDestinationEndpointInstanceId("rm-bxxxxxxxxxxxxxxxxx");
    request.setStructureInitialization(true);
    request.setDataInitialization(true);
    String SyncObject="[{\"DBName\":\"jiangliutest\",\"NewDBName\":\"jiangliutest\","
        + "\"TableIncludes\":[{\"TableName\":\"t1\",\"NewTableName\":\"t1\"}]}]";
    request.setSynchronizationObjects(SyncObject);
    ConfigureSynchronizationJobResponse response = new ConfigureSynchronizationJobResponse();
    try {
      response = client.getAcsResponse(request);
      System.out.println("Configure Sync Job Succeed!") ;
    } catch (Exception e) {
      // TODO: handle exception
      System.out.println("Configure Sync Job Failed");
      System.out.println(e.toString());
    }
  }
```

4.1.3.2. CreateSynchronizationJob

4.1.3.2.1. Overview

You can call this operation to purchase a data synchronization instance. After the purchase, you need to call the ConfigureSynchronizationJob operation to configure the instance. If your synchronization instance adopts the Pay-As-You-Go billing method, you will be charged only when you have called the ConfigureSynchronizationJob operation and started the synchronization task.

4.1.3.2.2. Request parameters

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateSynchronizationJob.
SourceRegion	String	Yes	The region where the source instance resides. • cn-hangzhou: Alibaba Cloud in the China (Hangzhou) region • cn-shanghai: Alibaba Cloud in the China (Shanghai) region • cn-qingdao: Alibaba Cloud in the China (Qingdao) region • cn-beijing: Alibaba Cloud in the China (Beijing) region • cn-zhangbei: Alibaba Cloud in the China (Zhangjiakou-Beijing Winter Olympics) region • cn-shenzhen: Alibaba Cloud in the China (Shenzhen) region • cn-hongkong: Alibaba Cloud in the China (Hong Kong) region • us-west-1: Alibaba Cloud in the US (Silicon Valley) region • us-east-1: Alibaba Cloud in the US (Virginia) region • ap-southeast-1: Alibaba Cloud in the Singapore (Singapore) region • cn-hangzhou-finance-1: Finance Cloud in the China (Hangzhou) region • cn-qingdao-finance-1: Finance Cloud in the China (Qingdao) region • cn-shanghai-finance-1: Finance Cloud in the China (Shanghai) region • cn-shenzhen-finance-1: Finance Cloud in the China (Shenzhen) region • me-east-1: Alibaba Cloud in the UAE (Dubai) region • en-central-1: Alibaba Cloud in the Germany (Frankfurt) region
SourceEndpoint.InstanceType	String	No	The type of the source instance. Valid values: RDS and DRDS. Default value: RDS.
DestinationEndpoint.InstanceType	String	No	The type of the destination instance. Valid values: RDS, DRDS, and AnalyticDB. Default value: RDS.

Parameter	Type	Required	Description
SourceRegion	String	Yes	The region where the destination instance resides. • cn-hangzhou: Alibaba Cloud in the China (Hangzhou) region • cn-shanghai: Alibaba Cloud in the China (Shanghai) region • cn-qingdao: Alibaba Cloud in the China (Qingdao) region • cn-beijing: Alibaba Cloud in the China (Beijing) region • cn-zhangbei: Alibaba Cloud in the China (Zhangjiakou-Beijing Winter Olympics) region • cn-shenzhen: Alibaba Cloud in the China (Shenzhen) region • cn-hongkong: Alibaba Cloud in the China (Hong Kong) region • us-west-1: Alibaba Cloud in the US (Silicon Valley) region • us-east-1: Alibaba Cloud in the US (Virginia) region • ap-southeast-1: Alibaba Cloud in the Singapore (Singapore) region • cn-hangzhou-finance-1: Finance Cloud in the China (Hangzhou) region • cn-qingdao-finance-1: Finance Cloud in the China (Qingdao) region • cn-shanghai-finance-1: Finance Cloud in the China (Shanghai) region • cn-shenzhen-finance-1: Finance Cloud in the China (Shenzhen) region • me-east-1: Alibaba Cloud in the UAE (Dubai) region • en-central-1: Alibaba Cloud in the Germany (Frankfurt) region
NetworkType	String	No	The type of the data transmission network when you synchronize data across regions. Valid values: Internet and Intranet. Default value: Internet.
SynchronizationJobClass	String	No	The specification of the data synchronization instance. Valid values: micro, small, medium, and large. Default value: small.
PayType	String	Yes	The billing method. Valid values: Postpaid and Prepaid.

Parameter	Type	Required	Description
Period	String	No	Specifies whether to purchase a yearly subscription instance or a monthly subscription instance. This parameter is required if the PayType parameter is set to Prepaid. Valid values: Year and Month.
UsedTime	String	No	The duration of the subscription, in months. This parameter is required if the Period parameter is set to Month. Set the value to an integer greater than 0.
ClientToken	String	No	The client token that is used to ensure the idempotence of the request.

4.1.3.2.3. Response parameters

Name	Type	Description
SynchronizationJobId	String	The GUID of the synchronization task.

4.1.3.2.4. Examples

The following code provides an example of how to use the Java SDK to purchase a synchronization instance.

```
public class DataSyncTest {
    public String createSyncJob(DefaultAcsClient client) {
        CreateSynchronizationJobRequest request = new CreateSynchronizationJobRequest();
        request.setSourceRegion("cn-hangzhou");
        request.setDestRegion("cn-hangzhou");
        request.setSynchronizationJobClass("micro");
        request.setPayType("Postpaid");
        request.setClientToken("createsyncjob_1");
        CreateSynchronizationJobResponse response = new CreateSynchronizationJobResponse();
        try {
            response = client.getAcsResponse(request);
            String DataSyncId  = response.getSynchronizationJobId();
            System.out.println("Create Synchronization Job Succeed! Synchronization Job Id is "+DataSyncId
);
            return DataSyncId;
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Create Synchronization Job Failed!") ;
            return null;
        }
    }
```

4.1.3.3. DeleteSynchronizationJob

4.1.3.3.1. Overview

You can call this operation to delete a synchronization task regardless of the synchronization task status. After a synchronization task is released, you cannot query or manage it through OpenAPI or the console.

4.1.3.3.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DeleteSynchronizationJob.
SynchronizationJobId	String	Yes	The synchronization task ID.

4.1.3.3.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.3.3.4. Examples

The following code provides an example of how to use the Java SDK to delete a synchronization task.

```
public void deleteSyncJob(String SyncJobId,DefaultAcsClient client) {
    DeleteSynchronizationJobRequest request = new DeleteSynchronizationJobRequest();
    request.setSynchronizationJobId(SyncJobId);
    DeleteSynchronizationJobResponse response = new DeleteSynchronizationJobResponse();
    try {
      response = client.getAcsResponse(request);
      System.out.println("Delete Sync Job "+SyncJobId+ " Succeed!") ;
    } catch (Exception e) {
      // TODO: handle exception
      System.out.println("Delete Sync Job "+SyncJobId+ " Failed!") ;
    }
  }
```

4.1.3.4. DescribeDataLoadStatus

4.1.3.4.1. Overview

You can call this operation to query initial synchronization details regardless of the synchronization task status.

4.1.3.4.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeDataLoadStatus.
SynchronizationJobId	String	Yes	The synchronization task ID.

Name	Type	Required	Description
StructureInitialization	Boolean	No	The schema initialization details.
DataInitialization	Boolean	No	The full data initialization details.
DataSynchronization	Boolean	No	The incremental data initialization details.
PageSize	Integer	No	The number of records on each page. Valid values: 30, 50, and 100. Default value: 30. This parameter is required when only one of these three parameters is set to true: StructureInitialization, DataInitialization, and DataSynchronization.
PageNum	Integer	No	The page number. It must be greater than 0 and smaller than or equal to the maximum value of Integer. Default value: 1. This parameter is required when only one of these three parameters is set to true: StructureInitialization, DataInitialization, and DataSynchronization.

4.1.3.4.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.3.4.4. Examples

The following code provides an example of how to use the Java SDK to query the initial synchronization details.

```
public void descInitStatus(String SyncJobId, DefaultAcsClient client) {
```

```
        DescribeInitializationStatusRequest request = new DescribeInitializationStatusRequest();
        request.setSynchronizationJobId(SyncJobId);
        DescribeInitializationStatusResponse response = new DescribeInitializationStatusResponse();
        try {
            response = client.getAcsResponse(request);
            System.out.println("Describe Initialization Detail of Synchronization Job "+SyncJobId+" Succeed!"
) ;
            List<StructureInitializationDetail> strucInitDetails  = response.getStructureInitializationDetails(
);
            List<DataInitializationDetail> dataInitDetails  = response.getDataInitializationDetails();
            List<com.aliyuncs.dts.model.v20160801. DescribeInitializationStatusResponse.DataSynchronizat
ionDetail> dataSynchronizationDetails = response.getDataSynchronizationDetails();
            System.out.println("Structure Initialization Detail:");
            for (int i = 0; i < strucInitDetails.size(); i++) {
                System.out.println("ObjectName:"+strucInitDetails.get(i).getObjectName()
                        +"; ObjectType:"+strucInitDetails.get(i).getObjectType()
                        +"; SRCDBName:"+strucInitDetails.get(i).getSourceOwnerDBName()
                        +"; DESTDBName:"+strucInitDetails.get(i).getDestinationOwnerDBName()
                        +"; Status:"+strucInitDetails.get(i).getStatus()
                        +"; ErrMSG:"+strucInitDetails.get(i).getErrorMessage());
            }
            System.out.println("Data Initialization Detail:");
            for (int i = 0; i < dataInitDetails.size(); i++) {
                System.out.println("TableName:"+dataInitDetails.get(i).getTableName()
                        +"; SRCDBName:"+dataInitDetails.get(i).getSourceOwnerDBName()
                        +"; DESTDBName:"+dataInitDetails.get(i).getDestinationOwnerDBName()
                        +"; Status:"+dataInitDetails.get(i).getStatus()
                        +"; TotalRows:"+dataInitDetails.get(i).getTotalRowNum()
                        +"; FinishedRows:"+dataInitDetails.get(i).getFinishRowNum()
                        +"; UsedTime:"+dataInitDetails.get(i).getUsedTime()
                        +"; ErroMSG:"+dataInitDetails.get(i).getErrorMessage());
            }
            System.out.println("Data Synchronization Detail:");
            for (int i = 0; i < dataSynchronizationDetails.size(); i++) {
                System.out.println("TableName:"+dataSynchronizationDetails.get(i).getTableName()
                        +"; SRCDBName:"+dataSynchronizationDetails.get(i).getSourceOwnerDBName()
                        +"; DESTDBName:"+dataSynchronizationDetails.get(i).getDestinationOwnerDBName()
                        +"; Status:"+dataSynchronizationDetails.get(i).getStatus()
                        +"; ErrMSG:"+dataSynchronizationDetails.get(i).getErrorMessage());
            }
        } catch (Exception e) {
```

```
        // TODO: handle exception
        System.out.println("Describe Initialization Status of Synchronization Job "+SyncJobId+" Failed!")
;
        System.out.println(e.toString());
    }
  }
```

4.1.3.5. DescribeSynchronizationJobStatus

4.1.3.5.1. Overview

You can call this operation to query the status and running details of a synchronization task regardless of its status.

4.1.3.5.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSynchronizationJobStatus.
SynchronizationJobId	String	Yes	The ID of the synchronization instance.
SynchronizationDirection	String	No	The synchronization direction. Valid values: Forward and Reverse. Default value: Forward.

4.1.3.5.3. Response parameters

Name	Type	Description
SynchronizationJobId	String	The synchronization task ID.
SynchronizationJobName	String	The synchronization task name.
SynchronizationJobClass	String	The synchronization task type.
PayType	String	The billing method.

Name	Type	Description
ExpireTime	String	The expiration time, in the format of YYYY-MM-DDT hh:mm:ssZ, such as 2011-06-09T 12:11:11Z.
SourceEndpoint	SourceEndpoint	The endpoint to access the source instance specified in the synchronization task.
DestinationEndpoint	DestinationEndpoint	The endpoint to access the destination instance specified in the synchronization task.
SynchronizationObjects	List	The objects to be synchronized.
StructureInitialization	String	Indicates whether to perform schema initialization.
DataInitialization	String	Indicates whether to perform full data initialization.
Status	String	The synchronization task status. Valid values: NotStarted, Prechecking, PrecheckFailed, Initializating, InitializeFailed, Synchronizing, Failed, Suspending, Modifying, and Finished.
ErrorMessage	String	The cause of the synchronization error.
PrecheckStatus	PrecheckStatus	The precheck status. For more information, see PrecheckStatus parameter definitions.
StructureInitializationStatus	StructureInitializationStatus	The schema initialization status. For more information, see StructureInitializationStatus parameter definitions.
DataInitializationStatus	DataInitializationStatus	The full data initialization status. For more information, see DataInitializationStatus parameter definitions.
DataSynchronizationStatus	DataSynchronizationStatus	The incremental data initialization status. For more information, see DataSynchronizationStatus parameter definitions.
Delay	String	The synchronization latency. Unit: second.

Name	Type	Description
Performance	String	The synchronization performance. For more information, see Performance parameter definitions.

4.1.3.5.4. PrecheckStatus parameter definitions

Name	Type	Description
Status	String	The precheck status.
Percent	String	The precheck progress. Unit: %.
Detail	List	The check details of each precheck item. For more information, see CheckItem parameter definitions.

4.1.3.5.5. CheckItem parameter definitions

Name	Type	Description
ItemName	String	The check item name.
CheckStatus	String	The check status. Valid values: • NotStarted • Checking • Failed • Finished
ErrorMessage	String	The cause of the error.
RepairMethod	String	The repair method.

4.1.3.5.6. StructureInitializationStatus parameter definitions

Name	Type	Description

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.3.5.7. DataInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.3.5.8. DataSynchronizationStatus parameter definitions

Name	Type	Description

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Delay	String	The migration latency. Unit: second.

4.1.3.5.9. Performance parameter definitions

Name	Type	Description
TPS	String	The number of records that are synchronized per second.
Flow	String	The data traffic that is synchronized per second. Unit: Mbit/s.

4.1.3.5.10. Examples

The following code provides an example of how to use the Java SDK to query the status of a synchronization task.

```
public void descSyncJobStatus(String SyncJobId,DefaultAcsClient client) {
    DescribeSynchronizationJobStatusRequest request = new DescribeSynchronizationJobStatusRequ
est();
    request.setSynchronizationJobId(SyncJobId);
    DescribeSynchronizationJobStatusResponse response = new DescribeSynchronizationJobStatusR
esponse();
    try {
      response = client.getAcsResponse(request);
      /*Get Sync Configure of the Sync Job*/
      System.out.println("SyncJobId:"+response.getSynchronizationJobId()
          +"SyncJobName:"+response.getSynchronizationJobName()
          +"; PayType:"+response.getPayType()+"; ExpireTime:"+response.getExpireTime()
```

```
                +"; SyncClass:"+response.getSynchronizationJobClass()
                +"; StrucInit:"+response.getStructureInitialization()
                +"; DataInit:"+response.getDataInitialization()+"; Status:"+response.getStatus()
                +"; Delay:"+response.getDelay()
                +"; ErrMSG:"+response.getErrorMessage());
        /*Get Sync Objects of the Sync Job*/
        System.out.println("Sync Objects:");
        List<SynchronizationObject> syncObjects = response.getSynchronizationObjects();
        /*Get the Source Endpoint of the synchronization job*/
        SourceEndpoint sourceEndpoint = response.getSourceEndpoint();
        System.out.println("InstanceType:"+sourceEndpoint.getInstanceType()
            +"InstanceId:"+sourceEndpoint.getInstanceId());
        /*Get the Destination Endpoint of the synchronization job*/
        DestinationEndpoint destinationEndpoint = new DestinationEndpoint();
        System.out.println("Instance Type:"+destinationEndpoint.getInstanceType()
            +destinationEndpoint.getInstanceId());
        // Get Sync Object of the synchronization job
        System.out.println("SyncObject:"+org.json.JSONObject.valueToString(syncObjects));
        // Get Precheck Status of the synchronization job
        try {
            com.aliyuncs.dts.model.v20160801. DescribeSynchronizationJobStatusResponse.PrecheckStat
us precheckStatus = response.getPrecheckStatus();
            System.out.println("Get Precheck Status for Sync Job Succeed");
            System.out.println("Precheck Status:"+response.getPrecheckStatus().getStatus()
                +";  Precheck Percent:"+response.getPrecheckStatus().getPercent());
            List<com.aliyuncs.dts.model.v20160801. DescribeSynchronizationJobStatusResponse.Precheck
Status.CheckItem> precheckDetails= precheckStatus.getDetail();
            for (int i = 0; i < response.getPrecheckStatus().getDetail().size(); i++) {
                System.out.println("CheckItem:"+response.getPrecheckStatus().getDetail().get(i).getCheck
Status()
                    +"; ErrorMSG:"+response.getPrecheckStatus().getDetail().get(i).getErrorMessage()
                    +"; RepairMethod:"+response.getPrecheckStatus().getDetail().get(i).getRepairMethod
());
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Get Precheck Status for Sync Job Failed");
        }
        // Get Structure Initialization Status of the synchronization job
        try {
            System.out.println("Get Structure Initialization Status Succeed!") ;
```

```
            System.out.println("Structure Init Status:"+response.getStructureInitializationStatus().getSt
atus()
                    +"; Percent:"+response.getStructureInitializationStatus().getPercent()
                    +"; Progress:"+response.getStructureInitializationStatus().getProgress()
                    +"; ErrMSG:"+response.getStructureInitializationStatus().getErrorMessage());
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Get Structure Init Status Failed!") ;
        }
        // Get Data Initialization Status of the synchronization job
        try {
            System.out.println("Get Data Initialization Status Succeed!") ;
            System.out.println("Data Init Status:"+response.getDataInitializationStatus().getStatus()
                    +"; Percent:"+response.getDataInitializationStatus().getPercent()
                    +"; Progress:"+response.getDataInitializationStatus().getProgress()
                    +"; ErrMSG:"+response.getDataInitializationStatus().getErrorMessage());
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Get Data Init Status Failed!") ;
            System.out.println(e.toString());
        }
        //Get Data Sync Status
        try {
            com.aliyuncs.dts.model.v20160801. DescribeSynchronizationJobStatusResponse.DataSynchron
izationStatus dataSyncStatus = response.getDataSynchronizationStatus();
            System.out.println("Get Data Synchronization Status Succeed!") ;
            System.out.println("Data Sync Status:"+response.getDataSynchronizationStatus().getStatus
()
                    +"; Percent:"+response.getDataSynchronizationStatus().getPercent()
                    +"; Delay:"+response.getDataSynchronizationStatus().getDelay()
                    +"; ErrMSG:"+response.getDataSynchronizationStatus().getErrorMessage());
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Get Data Sync Status Failed!") ;
            System.out.println(e.toString());
        }
        //Get Performance
        System.out.println("TPS:"+response.getPerformance().getRPS()
              +"Flow:"+response.getPerformance().getRPS());
      } catch (Exception e) {
        // TODO: handle exception
```

```
            // TODO: handle exception
            System.out.println("Describe Status of Data Sync Job "+ SyncJobId + "Failed");
            System.out.println(e.toString());
        }
    }
```

4.1.3.6. DescribeSynchronizationJobs

4.1.3.6.1. Overview

You can call this operation to query the status and details of all synchronization tasks under the current account.

4.1.3.6.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSynchronizationJobs.
PageSize	Integer	No	The number of records on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNum	Integer	No	The page number. It must be greater than 0 and smaller than or equal to the maximum value of Integer. Default value: 1.

4.1.3.6.3. Response parameters

Name	Type	Description
PageNumber	Integer	The page number.
TotalRecordCount	Integer	The total number of records.
PageRecordCount	Integer	The number of records on the page.

Name	Type	Description
SynchronizationInstances	List	The synchronization task status. For more information, see SynchronizationInstance parameter definitions.

4.1.3.6.4. SynchronizationInstance parameter definitions

Name	Type	Description
SynchronizationJobId	String	The synchronization task ID.
SynchronizationJobName	String	The synchronization task name.
SynchronizationJobClass	String	The synchronization task type.
PayType	String	The billing method.
ExpireTime	String	The expiration time, in the format of YYYY-MM-DDT hh:mm:ssZ, such as 2011-06-09T 12:11:11Z.
SourceEndpoint	SourceEndpoint	The endpoint to access the source instance specified in the synchronization task.
DestinationEndpoint	DestinationEndpoint	The endpoint to access the destination instance specified in the synchronization task.
SynchronizationObjects	List	The objects to be synchronized.
StructureInitialization	String	Indicates whether to perform schema initialization.
DataInitialization	String	Indicates whether to perform full data initialization.

Name	Type	Description
Status	String	The synchronization task status. Valid values: • NotStarted • Prechecking • PrecheckFailed • Initializating • InitializeFailed • Synchronizing • Failed • Suspending • Modifying • Finished
ErrorMessage	String	The cause of the synchronization error.
PrecheckStatus	PrecheckStatus	The precheck status. For more information, see PrecheckStatus parameter definitions.
StructureInitializationStatus	StructureInitializationStatus	The schema initialization status. For more information, see StructureInitializationStatus parameter definitions.
DataInitializationStatus	DataInitializationStatus	The full data initialization status. For more information, see DataInitializationStatus parameter definitions.
DataSynchronizationStatus	DataSynchronizationStatus	The incremental data initialization status. For more information, see DataSynchronizationStatus parameter definitions.
Delay	String	The synchronization latency. Unit: second.
Performance	String	The synchronization performance. For more information, see Performance parameter definitions.

4.1.3.6.5. PrecheckStatus parameter definitions

Name	Type	Description
Status	String	The precheck status.
Percent	String	The precheck progress. Unit: %.
Detail	List	The check details of each precheck item. For more information, see CheckItem parameter definitions.

4.1.3.6.6. CheckItem parameter definitions

Name	Type	Description
ItemName	String	The check item name.
CheckStatus	String	The check status. Valid values: • NotStarted • Checking • Failed • Finished
ErrorMessage	String	The cause of the error.
RepairMethod	String	The repair method.

4.1.3.6.7. StructureInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.

Name	Type	Description
Progress	String	The number of tables that have been migrated.

4.1.3.6.8. DataInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.3.6.9. DataSynchronizationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Delay	String	The migration latency. Unit: second.

4.1.3.6.10. Performance parameter definitions

Name	Type	Description
TPS	String	The number of records that are synchronized per second.
Flow	String	The data traffic that is synchronized per second. Unit: Mbit/s.

4.1.3.6.11. Examples

The following code provides an example of how to use the Java SDK to query the list of synchronization tasks.

```
public void descSyncJobs(DefaultAcsClient client) {
    DescribeSynchronizationJobsRequest request = new DescribeSynchronizationJobsRequest();
    DescribeSynchronizationJobsResponse response = new DescribeSynchronizationJobsResponse();
    try {
      response = client.getAcsResponse(request);
      System.out.println("Describe all Synchronization Jobs Succeed!") ;
      List<SynchronizationInstance> syncInstances=response.getSynchronizationInstances();
      System.out.println("Total num of Synchronization Jobs:"+syncInstances.size());
      for (SynchronizationInstance syncInstance : syncInstances) {
        System.out.println("SyncJobId:"+syncInstance.getSynchronizationJobId()
            +"; SyncJobName:"+syncInstance.getSynchronizationJobName()
            +"; PayType:"+syncInstance.getPayType()
            +"; InstanceClass:"+syncInstance.getSynchronizationJobClass()
            +"; StrucInit:"+syncInstance.getStructureInitialization()
            +"; DataInit:"+syncInstance.getDataInitialization()
            +"; Status:"+syncInstance.getStatus()
            +"; ErrMSG:"+syncInstance.getErrorMessage()
            +"; Delay:"+syncInstance.getDelay());
        System.out.println("SourceEndpointInstanceType:"+syncInstance.getSourceEndpoint().getIns
tanceType()
            +"SourceEnpointInstanceId:"+syncInstance.getSourceEndpoint().getInstanceId());
        System.out.println("DestEndpointInstanceType:"+syncInstance.getDestinationEndpoint().getI
nstanceType()
            +"DestEndpointInstanceId:"+syncInstance.getDestinationEndpoint().getInstanceId());
        // Get Precheck Status
        try {
          com.aliyuncs.dts.model.v20160801. DescribeSynchronizationJobsResponse.SynchronizationI
```

```
nstance.PrecheckStatus precheckStatus = syncInstance.getPrecheckStatus();
                System.out.println("Get Precheck Status for Sync Job Succeed");
                System.out.println("Precheck Status:"+syncInstance.getPrecheckStatus().getStatus()
                    +"Precheck Percent:"+syncInstance.getPrecheckStatus().getPercent());
                for (int j = 0; j < syncInstance.getPrecheckStatus().getDetail().size(); j++) {
                  System.out.println("CheckItem:"+syncInstance.getPrecheckStatus().getDetail().get(j).get
CheckStatus()
                    +"; ErrorMSG:"+syncInstance.getPrecheckStatus().getDetail().get(j).getErrorMessag
e()
                    +"; RepairMethod:"+syncInstance.getPrecheckStatus().getDetail().get(j).getRepairM
ethod());
                }
            } catch (Exception e) {
                // TODO: handle exception
                System.out.println("Get Precheck Status for Sync Job Failed");
            }
            // Get Structure Initialization Status
            try {
                System.out.println("Get Structure Initialization Status Succeed!") ;
                System.out.println("Structure Init Status:"+syncInstance.getStructureInitializationStatus().
getStatus()
                    +"; Percent:"+syncInstance.getStructureInitializationStatus().getPercent()
                    +"; Progress:"+syncInstance.getStructureInitializationStatus().getProgress()
                    +"; ErrMSG:"+syncInstance.getStructureInitializationStatus().getErrorMessage());
            } catch (Exception e) {
                // TODO: handle exception
                System.out.println("Get Structure Initialization Status Failed!") ;
            }
            // Get Data Initialization Status
            try {
                System.out.println("Get Data Init Status Succeed!") ;
                System.out.println("Data Init Status:"+syncInstance.getDataInitializationStatus().getStatu
s()
                    +"; Percent:"+syncInstance.getDataInitializationStatus().getPercent()
                    +"; Progress:"+syncInstance.getDataInitializationStatus().getProgress()
                    +"; ErrMSG:"+syncInstance.getDataInitializationStatus().getErrorMessage());
            } catch (Exception e) {
                // TODO: handle exception
                System.out.println("Get Data Init Status Failed!") ;
                System.out.println(e.toString());
            }
```

```
            }
            //Get Data Synchronization Status
            try {
               System.out.println("Get Data Synchronization Status Succeed!") ;
               System.out.println("Data Sync Status:"+syncInstance.getDataSynchronizationStatus().getS
tatus()
                     +"; Percent:"+syncInstance.getDataSynchronizationStatus().getPercent()
                     +"; Delay:"+syncInstance.getDataSynchronizationStatus().getDelay()
                     +"; ErrMSG:"+syncInstance.getDataSynchronizationStatus().getErrorMessage());
            } catch (Exception e) {
               // TODO: handle exception
               System.out.println("Get Data Sync Status Failed!") ;
               System.out.println(e.toString());
            }
         }
      } catch (Exception e) {
         // TODO: handle exception
         System.out.println("Desc all Sync Jobs Failed!") ;
         System.out.println(e.toString());
      }
   }
```

4.1.3.7. DescribeObjectModifyStatus

4.1.3.7.1. Overview

After you call the ModifySynchronizationObject operation to modify the list of objects to be synchronized, you can call this operation to query the modification status and progress.

4.1.3.7.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeObjectModifyStatus.

Name	Type	Required	Description
TaskId	String	Yes	The ID of the task for modifying the list of objects to be synchronized. It is the response parameter of the ModifySynchronizationObject operation.

4.1.3.7.3. Response parameters

Name	Type	Description
Status	String	The modification status of the list of objects to be synchronized. Valid values: • NotStarted • Prechecking • PrecheckFailed • Migrating • Failed • Finished
ErrorMessage	String	The reason why the list of objects to be synchronized failed to be modified.
PrecheckStatus	PrecheckStatus	The precheck status. For more information, see PrecheckStatus parameter definitions.
StructureInitializationStatus	StructureInitializationStatus	The schema initialization status. For more information, see StructureInitializationStatus parameter definitions.
DataInitializationStatus	DataInitializationStatus	The full data initialization status. For more information, see DataInitializationStatus parameter definitions.
DataSynchronizationStatus	DataSynchronizationStatus	The incremental data initialization status. For more information, see DataSynchronizationStatus parameter definitions.

4.1.3.7.4. PrecheckStatus parameter definitions

Name	Type	Description
Status	String	The precheck status.

Name	Type	Description
Percent	String	The precheck progress. Unit: %.
Detail	List	The check details of each precheck item. For more information, see CheckItem parameter definitions.

4.1.3.7.5. CheckItem parameter definitions

Name	Type	Description
ItemName	String	The check item name.
CheckStatus	String	The check status. Valid values: • NotStarted • Checking • Failed • Finished
ErrorMessage	String	The cause of the error.
RepairMethod	String	The repair method.

4.1.3.7.6. StructureInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.3.7.7. DataInitializationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Progress	String	The number of tables that have been migrated.

4.1.3.7.8. DataSynchronizationStatus parameter definitions

Name	Type	Description
Status	String	The migration status. Valid values: • NotStarted • Migrating • Failed • Finished
ErrorMessage	String	The cause of the migration error.
Percent	String	The migration progress. Unit: %.
Delay	String	The migration latency.

4.1.3.7.9. Examples

The following code provides an example of how to use the Java SDK to query the status of a task for modifying the list of objects to be synchronized.

```
public void descObjectModifyStatus(String SyncJobId,String TaskId,DefaultAcsClient client) {
        DescribeSynchronizationObjectModifyStatusRequest request = new DescribeSynchronizationObje
```

```
DescribeSynchronizationObjectModifyStatusRequest request = new DescribeSynchronizationObje
ctModifyStatusRequest();
    request.setTaskId(TaskId);
    DescribeSynchronizationObjectModifyStatusResponse response = new DescribeSynchronizationO
bjectModifyStatusResponse();
    try {
      response = client.getAcsResponse(request);
      System.out.println("Describe Sync Object Modify Status Succeed!") ;
      String status=response.getStatus();
      PrecheckStatus precheckStatus = response.getPrecheckStatus();
      StructureInitializationStatus strucInitStatus=response.getStructureInitializationStatus();
      DataInitializationStatus dataInitStatus=response.getDataInitializationStatus();
      DataSynchronizationStatus dataSyncStatus = response.getDataSynchronizationStatus();
      String errorMSG=response.getErrorMessage();
      //Get Precheck Status
      System.out.println("The precheck status of SYNC Job "+SyncJobId);
      String precheck_status=precheckStatus.getStatus();
      String precheck_percent=precheckStatus.getPercent();
      System.out.println("Precheck Status:"+precheckStatus.getStatus()
          +"; Precheck Percent:"+precheckStatus.getPercent());
      System.out.println("Precheck Detail:");
      for (int i = 0; i < precheckStatus.getDetail().size(); i++) {
        String ItemName=precheckStatus.getDetail().get(i).getItemName();
        String CheckStatus=precheckStatus.getDetail().get(i).getCheckStatus();
        String CheckErrorMsg=precheckStatus.getDetail().get(i).getErrorMessage();
        String RepairMethod = precheckStatus.getDetail().get(i).getRepairMethod();
        System.out.println("CheckItme:"+ItemName+"; Status:"+CheckStatus+"; ErrorMSG:"+CheckErr
orMsg
            +"; RepairMethod:"+RepairMethod);
      }
      // Get the status of the structure Initialization
      String strucInit_Status=strucInitStatus.getStatus();
      String strucInit_percent=strucInitStatus.getPercent();
      String strucInit_progress=strucInitStatus.getProgress();
      String strucInit_ErrMsg=strucInitStatus.getErrorMessage();
      System.out.println("The Structure Init Status of SYNC Job "+SyncJobId);
      System.out.println("Status:"+strucInit_Status+"; Percent:"+strucInit_percent
          +"; Progress:"+strucInit_progress+"; ErrMSG:"+strucInit_ErrMsg);
      // Get the status of the data Initialization
      String dataInit_Status=dataInitStatus.getStatus();
      String dataInit_Percent = dataInitStatus.getPercent();
```

```
        String dataInit_Progress = dataInitStatus.getProgress();
        String dataInit_ErrMSG = dataInitStatus.getErrorMessage();
        System.out.println("The data Init Status of SYNC Job "+SyncJobId);
        System.out.println("Status:"+dataInit_Status+"; Percent:"+dataInit_Percent
            +"; Progress:"+dataInit_Progress+"; ErrMSG:"+dataInit_ErrMSG);
        // Get the status of the data synchronization
        String dataSync_Status=dataSyncStatus.getStatus();
        String dataSync_Percent=dataSyncStatus.getPercent();
        String dataSync_Delay=dataSyncStatus.getDelay();
        String dataSync_ErrMSG=dataSyncStatus.getErrorMessage();
        System.out.println("The data Init Status of SYNC Job "+SyncJobId);
        System.out.println("Status:"+dataSync_Status+"; Percent:"+dataSync_Percent
            +"; Delay:"+dataSync_Delay+"; ErrMSG:"+dataSync_ErrMSG);
    } catch (Exception e) {
        // TODO: handle exception
        System.out.println("Describe Synchronization Objects Modify Status Failed");
        System.out.println(e.toString());
    }
}
```

4.1.3.8. ModifySynchronizationObject

4.1.3.8.1. Overview

When a synchronization task is in the NotStarted, Synchronizing, Failed, or Suspending status, you can call this operation to modify the list of objects to be synchronized. After that, you can call the DescribeObjectModifyStatus operation to query the detailed modification status of the list of objects to be synchronized.

4.1.3.8.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifySynchronizationObject.
SynchronizationJobId	String	Yes	The ID of the synchronization task.
SynchronizationObjects	List	Yes	The list of modified objects to be synchronized.

4.1.3.8.3. Response parameters

Name	Type	Description
TaskId	String	The ID of the task for modifying the list of objects to be synchronized. You can call the DescribeObjectModifyStatus operation based on this ID to query the modification status.

4.1.3.8.4. Examples

The following code provides an example of how to use the Java SDK to modify the list of objects to be synchronized.

```
public String modifySyncObjects(String SyncJobId,DefaultAcsClient client) {
    String SyncObject="[{\"DBName\":\"jiangliutest\",\"NewDBName\":\"jiangliutest\","
        + "\"TableIncludes\":[{\"TableName\":\"t2\",\"NewTableName\":\"t2\"}]}]";
    ModifySynchronizationObjectRequest request = new ModifySynchronizationObjectRequest();
    request.setSynchronizationJobId(SyncJobId);
    request.setSynchronizationObjects(SyncObject);
    ModifySynchronizationObjectResponse response = new ModifySynchronizationObjectResponse();
    try {
      response = client.getAcsResponse(request);
      String TaskId=response.getTaskId();
      System.out.println("Modify Sync Job "+SyncJobId+" Succeed! Modify task Id:"+TaskId);
      return TaskId;
    } catch (Exception e) {
      // TODO: handle exception
      System.out.println("Modify Sync Job "+ SyncJobId +" Failed!") ;
      System.out.println(e.toString());
      return null;
    }
  }
```

4.1.3.9. StartSynchronizationJob

4.1.3.9.1. Overview

When a synchronization task is in the NotStarted, Failed, or Suspending status, you can call this operation to start the synchronization task.

4.1.3.9.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to StartSynchronizationJob.
SynchronizationJobId	String	Yes	The synchronization task ID.

4.1.3.9.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.3.9.4. Examples

The following code provides an example of how to use the Java SDK to start a synchronization task.

```
public void startSyncJob(String SyncJobId,DefaultAcsClient client) {
      StartSynchronizationJobRequest request  = new StartSynchronizationJobRequest();
      request.setSynchronizationJobId(SyncJobId);
      StartSynchronizationJobResponse response = new StartSynchronizationJobResponse();
      try {
         response = client.getAcsResponse(request);
         System.out.println("Start Sync Job "+SyncJobId+" Succeed!") ;
      } catch (Exception e) {
         // TODO: handle exception
         System.out.println("Start Sync Job "+SyncJobId+ " Failed");
         System.out.println(e.toString());
      }
   }
```

4.1.3.10. SuspendSynchronizationJob

4.1.3.10.1. Overview

When a synchronization task is in the Synchronizing status, you can call this operation to suspend the synchronization task. If a synchronization task has been suspended for over 6 hours, it may fail to be started.

4.1.3.10.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to SuspendSynchronizationJob.
SynchronizationJobId	String	Yes	The synchronization task ID.

4.1.3.10.3. Response parameters

Name	Type	Description
<Common response parameters>	-	The request ID.

4.1.3.10.4. Examples

The following code provides an example of how to use the Java SDK to suspend a synchronization task.

```
public void suspendSyncJob(String SyncJobId,DefaultAcsClient client) {
    SuspendSynchronizationJobRequest request = new SuspendSynchronizationJobRequest();
    request.setSynchronizationJobId(SyncJobId);
    SuspendSynchronizationJobResponse response = new SuspendSynchronizationJobResponse();
    try {
       response = client.getAcsResponse(request);
       System.out.println("Suspend Sync Job "+SyncJobId+ " Succeed!") ;
    } catch (Exception e) {
       // TODO: handle exception
       System.out.println("Suspend Sync Job "+SyncJobId+" Failed");
       System.out.println(e.toString());
    }
  }
```

4.1.4. Data subscription APIs

4.1.4.1. ConfigureSubscriptionInstance

4.1.4.1.1. Overview

You can call this operation to modify the configurations of a change tracking task, such as instance connection details and objects for change tracking. You can call this operation multiple times to modify the configurations of a task before the task is started. Only the changes of RDS for MySQL instances can be tracked.

4.1.4.1.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ConfigureSubscriptionInstance.
SubscriptionInstanceId	String	Yes	The ID of the subscription channel to be configured.
SubscriptionInstanceName	String	No	The name of a subscription channel that can distinguish it from other subscription channels.
SourceEndpoint.InstanceType	String	No	The type of the subscription instance. Valid value: RDS. Default value: RDS.
SourceEndpoint.InstanceId	String	Yes	The RDS instance ID.
SubscriptionDataType.DDL	Boolean	Yes	Indicates whether to subscribe to schema updates made through DDL. Valid values: true (yes) and false (no). Default value: true.
SubscriptionDataType.DML	Boolean	Yes	Indicates whether to subscribe to data updates made through DML. Valid values: true (yes) and false (no). Default value: true.

Name	Type	Required	Description
SubscriptionObjects	List	Yes	The listening port of the source instance. This parameter is required when the source instance is a user-created database. For more information, see SubscriptionObject parameter definitions.

4.1.4.1.3. SubscriptionObject parameter definitions

The SubscriptionObject parameters are defined as JSON strings. Certain regular expressions can be used in the object configuration.

The SubscriptionObject parameter definitions are as follows:

```
SubscriptionObject:{
    DBName:"The name of the database to be subscribed",
    TableIncludes:[{
        TableName: "The name of the table to be migrated"
    }]
}
```

SubscriptionObject indicates the whitelist of objects to be subscribed.

You can use the following regular expressions to define different object names:

- A period (.) matches any single character except '\r\n'.
- An asterisk (*) matches zero or more occurrences of a subexpression which occurs before. h.*llo matches strings such as hllo and heeeello.
- A question mark (?) matches zero or one occurrence of a subexpression which occurs before. h.? llo matches hllo and hello, but not haello.
- The [characters] character set matches any character included in the braces. For example, h[aello] matches hallo and hello.
- The [^characters] negative character set does not match any character included in the braces. For example, h[^ae]llo matches hcllo and hdllo, but not hallo or hello.
- The [character1-character2] character range matches any character included in the range from character1 to character2, such as [0-9] and [a-z].

4.1.4.1.4. Response parameters

Name	Type	Description
RequestId	String	The GUID of the request.

Name	Type	Description
Success	String	Indicates whether the request is successful. Valid values: true and false.
ErrCode	String	The error code.
ErrMessage	String	The detailed error message.

4.1.4.1.5. Example

This topic describes how to use the Java SDK to configure a migration task.

```
private static void configureSbeJob(String MigrationJobId){
        String MigrationJobName="jiangliutest";
        /*Source Endpoint*/
        String SourceEndpoint_InstanceId="rm-xxxxxxxxxxxxxxxxx";
        String SourceEndpoint_InstanceType="rds";
        Boolean MigrationMode_SubscriptionDataTypeDML=true;
        Boolean MigrationMode_SubscriptionDataTypeDDL=true;
        String MigrationObjects="[{\"DBName\":\"jiangliutest\","
                + "\"TableIncludes\":[{\"TableName\":\"t1\"}]}]";
        ConfigureSubscriptionInstanceRequest request = new ConfigureSubscriptionInstanceRequest();
        request.setMigrationJobId(MigrationJobId);
        request.setMigrationJobName(MigrationJobName);
        request.setSubscriptionInstanceId(jobId);
        request.setSubscriptionDataTypeDDL(MigrationMode_SubscriptionDataTypeDDL);
        request.setSubscriptionDataTypeDML(MigrationMode_SubscriptionDataTypeDML);
        request.setSourceEndpointInstanceType(SourceEndpoint_InstanceType);
        request.setSourceEndpointInstanceID(SourceEndpoint_InstanceId);
        request.setMigrationObject(MigrationObjects);
        try {
           ConfigureSubscriptionInstanceResponse response = client.getAcsResponse(request);
           System.out.println("Configure "+ MigrationJobId+ " Succeed");
        } catch (Exception e) {
           // TODO: handle exception
           System.out.println("Configure "+MigrationJobId + " Failed!") ;
           System.out.println(e.toString());
        }
    }
```

4.1.4.2. CreateSubscriptionInstance

4.1.4.2.1. Overview

You can call this operation to purchase a database incremental subscription instance. After the purchase, you need to call the ConfigureSubscriptionInstance operation to configure a subscription task.

4.1.4.2.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to CreateSubscriptionInstance.
Region	String	Yes	The region of the subscription instance. Valid values: • cn-hangzhou: Alibaba Cloud in Hangzhou • cn-shanghai: Alibaba Cloud in Shanghai • cn-qingdao: Alibaba Cloud in Qingdao • cn-beijing: Alibaba Cloud in Beijing • cn-zhangbei: Alibaba Cloud in Zhangbei • cn-shenzhen: Alibaba Cloud in Shenzhen • cn-hongkong: Alibaba Cloud in Hong Kong • us-west-1: Alibaba Cloud in Western US • us-east-1: Alibaba Cloud in Eastern US • ap-southeast-1: Alibaba Cloud in Singapore • cn-hangzhou-finance-1: Finance Cloud in Hangzhou • cn-qingdao-finance-1: Finance Cloud in Qingdao • cn-shanghai-finance-1: Finance Cloud in Shanghai • cn-shenzhen-finance-1: Finance Cloud in Shenzhen • me-east-1: Alibaba Cloud in Dubai • en-central-1: Alibaba Cloud in Frankfurt
ClientToken	String	No	The token used to ensure the idempotence of the request. It can be any string.
OwnerID	String	No	If a RAM user is used to call this operation, the value of OwnerID is the UID of the Apsara Stack tenant account to which this RAM user belongs.

4.1.4.2.3. Response parameters

Name	Type	Description
SubscriptionInstanceId	String	The subscription channel ID. This is the GUID of a subscription channel.

4.1.4.2.4. Examples

The following code provides an example of how to use the Java SDK to purchase a migration instance.

```
private String create() throws Exception {
        CreateSubscriptionInstanceRequest request = new CreateSubscriptionInstanceRequest();
        request.setRegion("cn-hangzhou");
        request.setPayType("postpaid");
        CreateSubscriptionInstanceResponse response = client.getAcsResponse(request);
        if(response == null){
           throw new Exception("create fail");
        }
        System.out.println("CreateSubscriptionInstanceRequest, result :"+response.getSubscriptionInsta
nceId());
        return response.getSubscriptionInstanceId();
    }
```

4.1.4.3. DeleteSubscriptionInstance

4.1.4.3.1. Overview

You can call this operation to delete a subscription channel regardless of its status. After a subscription task is deleted, you cannot query or manage it through OpenAPI or the console.

4.1.4.3.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DeleteSubscriptionInstance.
SubscriptionInstanceId	String	Yes	The ID of the subscription channel to be released.

4.1.4.3.3. Response parameters

Name	Type	Description
RequestId	String	The GUID of the request.
Success	String	Indicates whether the request is successful. Valid values: true and false.
ErrCode	String	The error code.
ErrMessage	String	The detailed error message.

4.1.4.3.4. Examples

The following code provides an example of how to use the Java SDK to delete a subscription task.

```
private static void deleteMigrationJob(String MigrationJobId) {
      DeleteMigrationJobRequest request = new DeleteMigrationJobRequest();
      DeleteMigrationJobResponse response = new DeleteMigrationJobResponse();
      request.setMigrationJobId(MigrationJobId);
      try {
         response = client.getAcsResponse(request);
         System.out.println("Delete Migration Job "+MigrationJobId+" Succeed!") ;
      } catch (Exception e) {
         // TODO: handle exception
         System.out.println("Delete Migration Job "+MigrationJobId+" Failed!") ;
         System.out.println(e.toString());
      }
   }
```

4.1.4.4. DescribeSubscriptionInstances

4.1.4.4.1. Overview

You can call this operation to query the list of subscription channels that your Apsara Stack tenant account has access to. Data subscription channels in any status are returned.

4.1.4.4.2. Request parameters

Description	Type	Required	Description

Description	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSubscriptionInstances.
PageSize	Integer	No	The number of records on each page. Valid values: 30, 50, and 100. Default value: 30.
PageNum	Integer	No	The page number. It must be greater than 0 and smaller than or equal to the maximum value of Integer. Default value: 1.

4.1.4.4.3. Response parameters

Name	Type	Description
SubscriptionInstances	List	The subscription channel details. For more information, see DescribeSubscriptionInstanceStatus.
PageNumber	Integer	The page number.
TotalRecrodCount	Integer	The total number of records.
PageRecordCount	Integer	The number of records on the page.

4.1.4.4.4. Examples

The following code provides an example of how to use the Java SDK to query the list of subscription instances.

```
public List<SubscriptionInstance> list(String jobName) throws ClientException {
    DescribeSubscriptionInstancesRequest request = new DescribeSubscriptionInstancesRequest();
    request.setPageNum(1);
    request.setPageSize(20);
    if(jobName! =null) {
       request.setSubscriptionInstanceName(jobName);
    }
    DescribeSubscriptionInstancesResponse response = client.getAcsResponse(request);
    System.out.println("page no: "+response.getPageNumber());
    System.out.println("page size: "+response.getPageRecordCount());
    System.out.println("size: "+ response.getSubscriptionInstances().size());
    return response.getSubscriptionInstances();
  }
```

4.1.4.5. DescribeSubscriptionInstanceStatus

4.1.4.5.1. Overview

You can call this operation to query the status of a subscription channel regardless of the subscription task status. You can query the channel configuration information, channel status, and downstream consumption status.

4.1.4.5.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to DescribeSubscriptionInstanceStatus.
SubscriptionInstanceId	String	Yes	The subscription channel ID.

4.1.4.5.3. Response parameters

Name	Type	Description
SubscriptionInstanceId	String	The subscription channel ID.
SubscriptionInstanceName	String	The subscription channel name.
PayType	String	The billing method.

Name	Type	Description
SourceEndpoint	Endpoint	The endpoint to access the source instance specified in the subscription task. For more information, see Endpoint parameter definitions.
SubscriptionDataType.DDL	Boolean	Indicates whether the data to be subscribed contains schema updates made through DDL. Valid values: true (yes) and false (no).
SubscriptionDataType.DML	Boolean	Indicates whether the data to be subscribed contains data updates made through DML. Valid values: true (yes) and false (no).
SubscriptionObjects	List	The objects to be subscribed, listed in a SubscriptionObject array. For more information, see SubscriptionObject parameter definitions.
Status	String	The subscription channel status. Valid values: NotStarted, Prechecking, PrecheckFailed, Starting, Normal, and Abnormal.
ErrorMessage	String	The cause of the channel error.
ConsumptionCheckpoint	String	The consumption checkpoint of the downstream SDK connected to the subscription channel. The date format follows the ISO8601 standard. It is in the UTC format of YYYY-MM-DDThh:mm:ssZ. For example, 2017-08-15T12:00:00Z (or 20:00:00, August 15, 2017, China Standard Time).
BeginTimestamp	String	The start time of the timestamp range of the data stored in the subscription channel. The date format follows the ISO8601 standard. It is in the UTC format of YYYY-MM-DDThh:mm:ssZ. For example, 2017-08-15T12:00:00Z (or 20:00:00, August 15, 2017, China Standard Time).
EndTimestamp	String	The end time of the timestamp range of the data stored in the subscription channel. The date format follows the ISO8601 standard. It is in the UTC format of YYYY-MM-DDThh:mm:ssZ. For example, 2017-08-15T12:00:00Z (or 20:00:00, August 15, 2017, China Standard Time).

Name	Type	Description
ConsumptionClient	String	The downstream client information, in the following format: IP address of the downstream client:Random ID generated by DTS.

4.1.4.5.4. Endpoint parameter definitions

Name	Type	Description
InstanceType	String	The instance type.
Region	String	The region of the source instance.
InstanceId	String	The instance ID.
EngineName	String	The database type.
IP	String	The connection address of the database.
Port	String	The listening port of the database.
UserName	String	The username to access the database.
OracleSID	String	The Oracle SID.
DatabaseName	String	The database name.

4.1.4.5.5. SubscriptionObject parameter definitions

Name	Type	Description
databaseName	String	The name of the database from which the list of objects are to be subscribed.
tableList	List	The names of the tables to be subscribed.
wholeDatabase	Boolean	Indicates whether to subscribe to the whole database.

4.1.4.5.6. Examples

The following code provides an example of how to use the Java SDK to query the status of a subscription channel.

```
private void describe(String jobId) throws ClientException {
    DescribeSubscriptionInstanceStatusRequest request = new DescribeSubscriptionInstanceStatusR
equest();
    request.setSubscriptionInstanceId(jobId);
    DescribeSubscriptionInstanceStatusResponse response = client.getAcsResponse(
      request);
    System.out.println(response.getStatus());
    System.out.println(response.getBeginTimestamp());
    System.out.println(response.getEndTimestamp());
    System.out.println(response.getConsumptionCheckpoint());
  }
```

4.1.4.6. ModifyComsumptionTimestamp

4.1.4.6.1. Overview

You can call this operation to modify the consumption checkpoint only when the subscription channel is in the Normal status and the consumption client is stopped.

4.1.4.6.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifyComsumptionTimestamp.
SubscriptionInstanceId	String	Yes	The ID of the subscription channel for which the consumption checkpoint needs to be modified.
ConsumptionTimestamp	String	Yes	The timestamp of the modified consumption checkpoint. The timestamp format follows the ISO8601 standard. The UTC time is used. Format: yyyy-MM-ddTHH:mm:ssZ. For example, 2013-08-15T12:00:00Z indicates 20:00:00 on Tuesday, August 15, 2013 Beijing Time).

4.1.4.6.3. Response parameters

Name	Type	Description
RequestId	String	The GUID of the request.
Success	String	Indicates whether the request is successful. Valid values: true and false.
ErrCode	String	The error code.
ErrMessage	String	The detailed error message.

4.1.4.6.4. Examples

The following code provides an example of how to use the Java SDK to modify the consumption checkpoint.

```
private void modifyTimestamp(String jobId,String ts) throws ClientException {
    ModifyConsumptionTimestampRequest request = new ModifyConsumptionTimestampRequest();
    request.setSubscriptionInstanceId(jobId);
    //ts
    request.setConsumptionTimestamp(ts);
    ModifyConsumptionTimestampResponse response = client.getAcsResponse(request);
    System.out.println("modify timestamp ,result:" + response.getSuccess());
    }
```

4.1.4.7. ModifySubscriptionObject

4.1.4.7.1. Overview

When the subscription channel is in the NotStarted, Failed, Normal or Abnormal status, you can call this operation to modify the list of objects to be subscribed.

- If the subscription channel is in the Normal status, you call this operation to modify the list of objects to be subscribed. After that, DTS calls the StartSubscriptionInstance operation directly to start the subscription channel.
- If the subscription channel is in the NotStarted, Failed, or Abnormal status, you call this operation to modify the list of objects to be subscribed. After that, you need to call the StartSubscriptionInstance operation to start the subscription channel if you want to.

4.1.4.7.2. Request parameters

Description	Type	Required	Description

Description	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to ModifySubscriptionObject.
SubscriptionInstanceId	String	Yes	The ID of the subscription channel for the list of objects to be modified.
SubscriptionObjects	List	Yes	The new list of objects to be subscribed after the modification. For more information, see SubscriptionObject parameter definitions.

4.1.4.7.3. SubscriptionObject parameter definitions

The SubscriptionObject parameters are defined as JSON strings. Certain regular expressions can be used in the object configuration.

The SubscriptionObject parameter definitions are as follows:

```
SubscriptionObject:{
    DBName:"The name of the database to be subscribed",
    TableIncludes:[{
        TableName: "The name of the table to be migrated"
    }]
}
```

SubscriptionObject indicates the whitelist of objects to be subscribed.

You can use the following regular expressions to define different object names:

- A period (.) matches any single character except '\r\n'.
- An asterisk (*) matches zero or more occurrences of a subexpression which occurs before. h.*llo matches strings such as hllo and heeeello.
- A question mark (?) matches zero or one occurrence of a subexpression which occurs before. h.? llo matches hllo and hello, but not haello.
- The [characters] character set matches any character included in the braces. For example, h[aello] matches hallo and hello.
- The [^characters] negative character set does not match any character included in the braces. For example, h[^ae]llo matches hcllo and hdllo, but not hallo or hello.
- The [character1-character2] character range matches any character included in the range from character1 to character2, such as [0-9] and [a-z].

4.1.4.7.4. Response parameters

Name	Type	Description
RequestId	String	The GUID of the request.
Success	String	Indicates whether the request is successful. Valid values: true and false.
ErrCode	String	The error code.
ErrMessage	String	The detailed error message.

4.1.4.7.5. Examples

The following code provides an example of how to use the Java SDK to modify the list of objects to be subscribed.

```
private void modifyObject(String jobId) throws ClientException {
    ModifySubscriptionObjectRequest request = new ModifySubscriptionObjectRequest();
    request.setSubscriptionInstanceId(jobId);
    String sbeObject = "[{\n"
                + "\"DBName\":\"miyuntest\","
                + "\"TableIncludes\":["
                + "{\"TableName\":\"t1\"},"
                + "{\"TableName\":\"t2\"},"
                + "{\"TableName\":\"t3\"}"
                + "]"
                + "}]";
    request.setSubscriptionObject(sbeObject);
    ModifySubscriptionObjectResponse response = client.getAcsResponse(request);
    System.out.println(response.getSuccess());
  }
```

4.1.4.8. StartSubscriptionInstance

4.1.4.8.1. Overview

You can call this operation to start a subscription channel only when the channel is in the NotStarted, Failed, or Suspending status.

4.1.4.8.2. Request parameters

Name	Type	Required	Description
Action	String	Yes	The operation that you want to perform. Set the value to StartSubscriptionInstance.
SubscriptionInstanceId	String	Yes	The ID of the subscription channel to be started.

4.1.4.8.3. Response parameters

Name	Type	Description
RequestId	String	The GUID of the request.
Success	String	Indicates whether the request is successful. Valid values: true and false.
ErrCode	String	The error code.
ErrMessage	String	The detailed error message.

4.1.4.8.4. Examples

The following code provides an example of how to use the Java SDK to start a subscription task.

```
private static void startSbeJob(String MigrationJobId){
    StartSubscriptionInstanceRequest request = new StartSubscriptionInstanceRequest();
    request.setSubscriptionInstanceId(jobId);
    try {
       StartSubscriptionInstanceResponse response = client.getAcsResponse(request);
       System.out.println("Start "+MigrationJobId+" Succeed");
    } catch (Exception e) {
       // TODO: handle exception
       System.out.println("Start "+MigrationJobId + " Failed");
       System.out.println(e.toString());
    }
  }
```

4.2. Call the Java SDK

4.2.1. Quick start

This topic describes how to obtain and call the *DTS* Java SDK.

Install the Apsara Stack Java SDK

If you use Apache Maven to manage Java projects, you only need to import the JAR file to *pom.xml* of the project.

To use the development kit of Alibaba Cloud products, you must install the SDK core library and the product SDK.

The SDK core library version is 3.2.8, and the *DTS* SDK version is *2.0.3*.

```
<dependency>
   <groupId>com.aliyun</groupId>
   <artifactId>aliyun-java-sdk-core</artifactId>
   <version>3.2.8</version>
</dependency>
<dependency>
   <groupId>com.aliyun</groupId>
   <artifactId>aliyun-java-sdk-dts</artifactId>
   <version>2.0.3-SNAPSHOT</version>
</dependency>
```

Use the Java SDK

The DTS SDK includes two core kits: aliyun-java-sdk-core-inner and aliyun-java-sdk-dts. The aliyun-java-sdk-core-inner kit automatically routes requests to the DTS endpoint. The aliyun-java-sdk-dts kit includes the related classes of all DTS functions.

The following uses CreateMigrationJob, the method that uses the DTS Java SDK to create a data migration task, to describe the complete process of using the SDK. The aliyun-java-sdk-dts kit includes all classes used.

1. Create an object of IClientProfile named profile. This object stores the Access Key ID, Access Key Secret, and default region information.

   ```
   IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", accsessKeyId, accessSecret);
   ```

2. Create the DefaultAcsClient object. All subsequent responses are obtained from DefaultAcsClient.

   ```
   client = new DefaultAcsClient(profile);
   ```

3. Create a request for the corresponding method. The request class is named in the format of API method name + Request. For example, the name of the API method used to create a migration task is CreateMigrationJob, so the corresponding request class name is CreateMigrationJobRequest. You can use a constructor to create a default class.

   ```
   CreateMigrationJobRequest request=new CreateMigrationJobRequest();
   ```

4. Use the setXxx method of the request class to configure information that must be provided in API parameters. For example, configure the region and specification of a migration task.

```
request.setRegion("cn-hangzhou");
request.setMigrationJobClass("small");
```

5. Use the DefaultAcsClient object to obtain a response to the corresponding request.

```
ConfigureMigrationJobResponse response = client.getAcsResponse(request);
```

6. Call the getXxx method in the response to obtain a returned parameter value, such as the migration task ID.

```
String MigrationJobId = response.getMigrationJobId();
```

7. The call process is now completed.

4.2.2. User manual

4.2.2.1. Install the Java SDK

This topic describes how to install and use the *DTS* Java SDK.

The Apsara Stack Java SDK supports JDK V1.6 and later. You can install the latest JDK version by importing the JAR file.

You can click here to download the *DTS* JAR file.

If you are using Eclipse or IntelliJ, you can install the *DTS* Java SDK directly by importing the JAR file.

The installation method is as follows:

- In Eclipse

  To install the DTS Java SDK in Eclipse, follow these steps:

  i. Copy the downloaded `aliyun-java-sdk-dts-2.0.3.jar` file to your project folder.

  ii. In Eclipse, right click the project, and then select **Properties**.

  iii. In the dialog box that appears, choose **Java Build Path** > **Libraries** > **Add JARs** to add the downloaded JAR files.

  iv. Click **Apply and Close**.

- In IntelliJ

  To install the DTS Java SDK in IntelliJ, follow these steps:

  i. Copy the downloaded `aliyun-java-sdk-dts-2.0.3.jar` file to your project folder.

  ii. Open your project in IntelliJ, and choose **File** > **Project Structure** .

  iii. Click **Apply** and then **OK**.

4.2.2.2. Configure identity verification credentials

When using the Apsara Stack SDK to access Apsara Stack services, you must provide an authorized account for identity verification.

Currently, the Java SDK supports identity verification based on the AccessKey.

When an Access Key is used as the access credential, you need to configure the credential during client initialization.

> **Notice** Do not disclose any code that may contain your AccessKey. For example, do not commit the code to public GitHub projects. Otherwise, your Apsara Stack tenant account may be compromised.

```
DefaultProfile profile = DefaultProfile.getProfile(
   "<your-region-id>",        // The region ID.
   "<your-access-key-id>",    // The AccessKey ID of the authorized account.
   "<your-access-key-secret>"); // The AccessKey Secret of the authorized account.
```

4.2.2.3. Initiate a call

This topic describes how to use the *DTS* Java SDK to initiate a request.

Procedure

1. Create an AcsClient instance.

   ```
   IAcsClient client = new DefaultAcsClient(profile);
   ```

2. Create a request.

   A request class name is in the format of `${apiName}Request`, where ${apiName} indicates the API name, such as *CreateMigrationJobRequest*.

   When the SDKs of multiple services are used, different requests may have the same name. Distinguish the requests by service package.

   ```
   CreateMigrationJobRequest request=new CreateMigrationJobRequest();
   ```

3. Initiate a call and handle the response.

   ```
   try {
                   ConfigureMigrationJobResponse response = client.getAcsResponse(request);
                   System.out.println("Configure "+ MigrationJobId+ " Succeed");
                   } catch (Exception e) {
                   // TODO: handle exception
                   System.out.println("Configure "+MigrationJobId + " Failed!") ;
                   System.out.println(e.toString());
                   }
   ```

In normal cases, all response fields are deserialized into the response. You can directly call `response.getXXX()` to obtain the response fields.

However, if an exception occurs or you want to obtain the original HTTP response, you can use the `doAction()` method to obtain the original response.

```
HttpResponse response = client.doAction(request);
```

4.2.2.4. Troubleshooting

The Java SDK returns corresponding errors when errors occur on the server side or the SDK side. These errors contain detailed error information, including the error code and the error message.

In normal cases, you do not need to solve the errors returned by the Java SDK. You only need to handle the errors returned by the server.

- `ServerException` is the error returned by the server.
- `ClientException` is the error returned by the SDK side.

For example, when the following error occurs, you have to modify the AccessKey ID based on the error message.

```
com.aliyuncs.exceptions.ClientException: InvalidAccessKeyId.NotFound : Specified access key is not fo
und.
```

If you want to handle the errors returned by the Java SDK, refer to the following example to write your code:

```
try {
    FooResponse response = client.getAcsResponse(request);
    // The following code is used to solve returned errors:
    // ...
}catch (ServerException e){
    // You can add your own error handling logic here
    // For example, display the error message
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
    // If you cannot solve the error, submit a ticket and provide the request ID to Alibaba technical support personnel.
    System.out.println("ResponseId=" + e.getRequestId());
}catch (ClientException e){
    // You can add your own error handling logic here.
    // For example, display the error message.
    System.out.println("ErrorCode=" + e.getErrCode());
    System.out.println("ErrorMessage=" + e.getErrMsg());
}
```

4.2.2.5. Errors and troubleshooting

The following are some common errors you may experience when using the Apsara Stack SDK.

Error code	Error message	Cause	Solution
SDK.CanNotResolveEndpoint	Can not resolve endpoint, please check the user guide	The SDK cannot automatically obtain the endpoint of the called product in the specified region.	Check whether the provided region ID and endpoint are correct. After obtaining the correct endpoint, run the following code to set the endpoint: `DefaultProfile.addEndpoint("your-region-id", "your-region-id", "product", "endpoint"");`

Error code	Error message	Cause	Solution
SDK.JsonUnmarshalError	Failed to unmarshal response	SDK response deserialization fails. In most cases, it is because the response structure received by the SDK does not conform to the API metadata. For example, the fields do not match or the format is incorrect.	You can use the client.doAction(request) method to obtain the original HTTP response.
SDK.TimeoutError	*The request timed out 4 times(3 for retry), perhaps we should have the threshold raised a little?*	The request has timed out and all retry attempts have failed.	In scenarios such as low network quality, we recommend that you increase the timeout value or the allowed number of retries.
SDK.ServerError:InvalidProtocol.NeedSsl	Your request is denied as lack of ssl protect.Recommend:https://error-center.aliyun.com/status/search?Keyword=InvalidProtocol.NeedSsl&source=PopGw	The operation only accepts HTTPS requests but not HTTP requests.	Add the following code before sending a request: request.setProtocol(ProtocolType.HTTPS)

5.Cloud Native Distributed Database PolarDB-X

5.1. API reference

5.1.1. List of operations by function

The following tables list API operations available for use in Cloud Native Distributed Database PolarDB-X.

Database management

API	Description
ChangeAccountPassword	Resets the password of a PolarDB-X database.
CheckDrdsDbName	Checks whether the name of a PolarDB-X database is valid.
DescribeRdsPerformanceSummary	Queries the performance metrics of ApsaraDB RDS for MySQL instances for a PolarDB-X instance, including the CPU utilization, input/output operations per second (IOPS), and active sessions.
DescribeRecycleBinStatus	Queries the status of the table recycle bin for a PolarDB-X database.
DescribeRecycleBinTables	Queries the tables that can be recycled for a PolarDB-X database.
DescribeRdsSuperAccountInstances	Queries the privileged accounts of ApsaraDB RDS for MySQL instances for a PolarDB-X instance.
DescribeDrdsSlowSqls	Queries slow SQL statements of a PolarDB-X database.
DescribeDrdsDbInstance	Queries the details about an ApsaraDB RDS for MySQL instance of a PolarDB-X database.
DescribeDrdsDbInstances	Queries the ApsaraDB RDS for MySQL instances of a PolarDB-X database.
DescribeDrdsDbRdsNameList	Queries the storage instances of a PolarDB-X database.
DescribeDrdsDbRdsRelationInfo	Queries the structure information of storage instances for a PolarDB-X database.

API	Description
DescribeDrdsDbTasks	Queries incomplete tasks for a PolarDB-X database.
DescribeDrdsInstanceDbMonitor	Queries the performance monitoring data of a PolarDB-X database.
SubmitCleanTask	Submits a cleanup task for smooth scale-out of a PolarDB-X database.
SubmitSmoothExpandPreCheckTask	Submits a precheck task for smooth scale-out of a PolarDB-X database.
ValidateShardTask	Verifies whether a partitioning task of a PolarDB-X database is valid.
SubmitSmoothExpandPreCheck	Submits a precheck task for smooth scale-out of a PolarDB-X database.
StartRestore	Starts a backup-based recovery task for a PolarDB-X instance.
RefreshDrdsAtomUrl	Refreshes the connection between the PolarDB-X instance and the ApsaraDB RDS for MySQL instance whose network type is changed to ensure that the connection is smooth.
ModifyRdsReadWeight	Modifies the read weights of ApsaraDB RDS for MySQL instances.
	Removes database creation failure records from a PolarDB-X instance.
RemoveDrdsDb	Removes a database from a PolarDB-X instance.
RemoveBackupsSet	Removes a backup set from a PolarDB-X instance.
ReleaseHiStoreInstance	Releases a columnar storage instance, namely, a HiStore instance of a PolarDB-X instance.
DescribeBroadcastTables	Queries the broadcast tables in a PolarDB-X database.
RemoveRecycleBinTable	Removes tables from the recycle bin for a PolarDB-X database.
DescribeBackMenu	Queries whether a PolarDB-X instance supports backup and recovery.
DescribeDrdsSqlAuditStatus	Queries the details about SQL audit for a PolarDB-X instance.

API	Description
DescribeTables	Queries the details about tables in a PolarDB-X database.
DescribeBackupTimes	Queries the point in time to which a PolarDB-X instance can be restored.
DescribeGlobalBroadcastType	Queries whether the table in a PolarDB-X database is a broadcast table.
DescribeHotDbList	Queries the available databases for hot-spot scale-out of a PolarDB-X instance.
DescribeExpandLogicTableInfoList	Queries the shard key of a PolarDB-X database.
DescribeTableListByType	Queries type-specific tables of a PolarDB-X database.
DisableSqlAudit	Disables SQL audit of a PolarDB-X database.
DescribeShardTaskList	Queries the partitioning tasks of a PolarDB-X database.
DescribeTable	Queries the table schema of a PolarDB-X database.
DescribeInstDbLogInfo	Queries the time range of historical SQL data that can be imported to a PolarDB-X database.
ResetDrdsToRdsConnections	Resets the number of connections from the PolarDB-X instance to ApsaraDB RDS for MySQL instances.
DatalinkReplicationPrecheck	Prechecks a partitioning task of a PolarDB-X database.
DescribeRDSPerformance	Queries the performance data of ApsaraDB RDS for MySQL instances for a PolarDB-X instance.
CreateShardTask	Creates a partitioning task for a PolarDB-X database.
DescribeShardTaskInfo	Queries details about a partitioning task of a PolarDB-X database.
DescribeBackupDbs	Queries the databases that can be restored for a PolarDB-X instance.
DeleteShardTasks	Deletes partitioning tasks from a PolarDB-X database.
ModifyDrdsIpWhiteList	Modifies the IP address whitelist of a PolarDB-X database.

API	Description
PutStartBackup	Submits a data backup task for a PolarDB-X instance.
DescribeSqlFlashbakTask	Queries the flashback tasks of a PolarDB-X database.
CheckExpandStatus	Verifies whether scale-out operations such as smooth scale-out can be performed on a PolarDB-X database.
CheckSqlAuditEnableStatus	Verifies whether SQL audit is enabled for a PolarDB-X logical database.
SetupBroadcastTables	Configures a broadcast table for a PolarDB-X database.
SetupRecycleBinStatus	Enables the table recycle bin for a PolarDB-X database.
SetupTable	Enables full table scan for a PolarDB-X database.
ModifyAccountPrivilege	Modifies account permissions for a PolarDB-X instance.
DescribeDrdsShardingDbs	Queries details about a horizontally partitioned PolarDB-X database.
CreateDrdsDB	Creates a PolarDB-X database.
DescribeDbInstances	Queries the storage instances of a PolarDB-X instance.
DescribeDrdsDB	Queries the details about a database on a PolarDB-X instance.
DescribeDbInstanceDbs	Queries the databases on a storage instance of a PolarDB-X instance.
SubmitSmoothExpandTask	Submits a smooth scale-out task for a PolarDB-X database.
SubmitSwitchTask	Submits a switchover task for smooth scale-out of a PolarDB-X database.

Instance management

API	Description
DescribeDrdsDBs	Queries the details about databases on a PolarDB-X instance.
DescribeDrdsInstance	Queries the details about a PolarDB-X instance.

API	Description
DescribeDrdsInstanceLevelTasks	Queries incomplete tasks for a PolarDB-X instance.
DescribeDrdsInstanceMonitor	Queries the monitoring data of a PolarDB-X instance.
DescribeDrdsInstances	Queries PolarDB-X instances that meet specific criteria in a region.
DescribeDrdsInstanceVersion	Queries the versions of a PolarDB-X instance.
DescribeDrdsParams	Queries the parameters of a PolarDB-X instance or database.
RollbackInstanceVersion	Rolls back the version of a PolarDB-X instance.
UpgradeInstanceVersion	Upgrades the version of the PolarDB-X instance.
SubmitRollbackTask	Submits a smooth scale-out rollback task for a PolarDB-X database.
SwitchGlobalBroadcastType	Switches the broadcast table in a PolarDB-X database from the multi-write mode to the asynchronous link mode.
SubmitSqlFlashbackTask	Submits an SQL flashback task for a PolarDB-X instance.
RemoveDrdsInstance	Releases a PolarDB-X instance.
DescribeDrdsTasks	Queries the change tasks of a PolarDB-X instance.
RestartDrdsInstance	Restarts a PolarDB-X instance.
DescribePreCheckResult	Queries precheck results of a PolarDB-X instance.
DescribeHiStoreInstanceInfo	Queries the details about a HiStore instance of a PolarDB-X instance.
DescribeStorageInstanceSubDbInfo	Queries the details about a storage instance of a PolarDB-X database.
RemoveInstanceAccount	Deletes the instance management account of a PolarDB-X instance.
DescribeInstanceMenuSwitch	Queries the switch configuration of a PolarDB-X instance.
DescribeInstanceAccounts	Queries the account information of a PolarDB-X instance.

API	Description
DescribeInstDbSlsInfo	Queries the names of the Log Service project and Logstore used by SQL audit.
SetBackupPolicy	Configures a backup policy for a PolarDB-X instance.
EnableSqlAudit	Enables SQL audit for a PolarDB-X database.
EnableSqlFlashbackMatchSwitch	Enables SQL flashback matching. After it is enabled, the query information is added to the ApsaraDB RDS for MySQL binary logs that are used by the PolarDB-X database to precisely restore data.
SetupDrdsParams	Configures parameters for a PolarDB-X instance or database.
FlashbackRecycleBinTable	Restores a logical table in the recycle bin.
ModifyAccountDescription	Modifies the account description of a PolarDB-X instance.
ModifyDrdsInstanceDescription	Modifies the description of a PolarDB-X instance.
CreateInstanceAccount	Creates an account for a PolarDB-X instance.

5.1.2. Make API requests

To send a Cloud Native Distributed Database PolarDB-X API request, you must send an HTTP GET request to the Cloud Native Distributed Database PolarDB-X endpoint. You must add the request parameters that correspond to the API operation being called. After you call the API operation, the system returns a response. The request and response are encoded in UTF-8.

Request syntax

Cloud Native Distributed Database PolarDB-X API operations use the RPC protocol. You can call Cloud Native Distributed Database PolarDB-X API operations by sending HTTP GET requests.

The request syntax is as follows:

```
http://Endpoint/?Action=xx&Parameters
```

- Endpoint: the endpoint of Cloud Native Distributed Database PolarDB-X. For more information about how to obtain the endpoint, see Obtain the endpoint of ASAPI.
- Action: the name of the operation being performed. For example, to query tables in a Cloud Native Distributed Database PolarDB-X database, you must set the Action parameter to DescribeTables.
- Version: the version of the API. For example, the version of the Cloud Native Distributed Database PolarDB-X API is `2019-01-23`.

- Parameters: the request parameters for the operation. Separate multiple parameters with ampersands (&).

Request parameters include both common parameters and operation-specific parameters. Common parameters are used for all PolarDB-X API calls regardless of the operation. For more information, see Common parameters.

The following example demonstrates how to call the DescribeTables operation in Cloud Native Distributed Database PolarDB-X:

```
https://drds.aliyuncs.com/?Action=DescribeTables
&Format=xml
&Version=2019-01-23
&Signature=xxxx%xxxx%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15xxxxxxxxx396
&SignatureVersion=1.0
&AccessKeyId=key-test
&TimeStamp=2012-06-01T12:00:00Z
...
```

5.1.3. Common parameters

Common request parameters

Parameter	Type	Required	Description
Format	String	No	The format in which the response is returned. Valid values: JSON (default) and XML.
Version	String	Yes	The version number of the API, in the format of YYYY-MM-DD. Valid value: 2019-01-23.
AccessKeyId	String	Yes	The AccessKey ID provided to you by Apsara Stack.
Signature	String	Yes	The signature string of the current request.

Parameter	Type	Required	Description
SignatureMethod	String	Yes	The encryption method of the signature string. Set the value to HMAC-SHA1.
Timestamp	String	Yes	The time at which the quest is signed. Specify the UTC time in the ISO 8601 standard in the YYYY-MM-DDThh:mm:ssZ format. For example, 20:00:00 on January 10, 2013 (UTC+8) is written as 2013-01-10T12:00:00Z.
SignatureVersion	String	Yes	The version of the signature encryption algorithm. Set the value to 1.0.
SignatureNonce	String	Yes	A unique and random number used to prevent replay attacks. You must use different numbers for different requests.
ResourceOwnerAccount	String	No	The owner account (the logon username) of the resource you want to access by using this API request.

Common response parameters

Responses can be returned in either the JSON or XML format. You can specify the response format in the request. The default response format is XML. Each time you send a request to call an API operation, the system returns a unique identifier RequestId, regardless of whether the request is successful or not.

- An HTTP status code of `2xx` indicates that the call succeeds.
- An HTTP status code of `4xx` or `5xx` indicates that the call failed.

The following are sample responses:

- XML format

```
<? xml version="1.0" encoding="utf-8"? >
  <!—Result root node-->
  <API operation name+response>
    <!—Returned request tag-->
    <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
    <!—Response data-->
  </API name+Response>
```

- JSON format

```
{
  "RequestId":"4C467B38-3910-447D-87BC-AC049166F216",
  /*Response data*/
  }
```

5.1.4. Request signature

This topic describes how to sign the request when you send an API request.

AccessKey ID and AccessKey secret

- PolarDB-X verifies the identity of each API request sender. Therefore, whether you send an HTTP request or an HTTPS request, the request must contain the signature generated by an AccessKey ID and an AccessKey secret.
- An AccessKey ID and an AccessKey secret are issued to visitors by Alibaba Cloud. Visitors can apply for and manage an AccessKey pair on the Alibaba Cloud official website. The AccessKey ID is used to verify the identity of the visitor.
- The AccessKey secret is used to encrypt the signature string on the client and verify the signature string on the server. You must keep your AccessKey secret strictly confidential. For more information about how to obtain an AccessKey pair, see Obtain an AccessKey pair.

Signature method

When you send an API request, take the following steps to sign the request:

1. Create a canonicalized query string by arranging the request parameters.
   i. Arrange the request parameters (including all common and operation-specific parameters except Signature) in alphabetical order.

   > **Notice** When you use the GET method to submit requests, these parameters constitute the parameter field of the request URI. They come after the question mark (?) in the request URI and are connected by ampersands (&).

ii. Encode the name and value of each request parameter. The parameter names and values must be URL encoded by using UTF-8. The URL encoding rules are described as follows:

- Uppercase letters, lowercase letters, digits, and some special characters such as hyphens (-), underscores (_), periods (.), and tildes (~) do not need to be encoded.
- Other characters must be percent encoded in %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 %XY%ZA... format.
- Spaces must be encoded as %20. Do not encode spaces as plus signs (+).

> **Note** Libraries that support URL encoding, such as the java.net.URLEncoder class in Java, encode strings in compliance with the requirements of the "application/x-www-form-urlencoded" Multipurpose Internet Mail Extensions (MIME) type. If you use this encoding method, you can replace plus signs (+) with %20, asterisks (*) with %2A, and %7E with tildes (~) in the encoded string to obtain the required string.

iii. Connect the encoded parameter names and values with equal signs (=).

iv. Sort the parameter name and value pairs connected by equal signs (=) in alphabetical order and connect the pairs with ampersands (&).

2. Use the created canonicalized query string to construct a string-to-sign based on the following rules:

```
StringToSign=
 HTTPMethod + "&" +
 percentEncode("/") + "&" +
 percentEncode(CanonicalizedQueryString)
```

- HTTPMethod specifies the HTTP method that is used to submit a request, such as GET.
- percentEncode("/") specifies the encoded value (%2F) for the forward slash (/) based on the URL encoding rules described in step 1.b.
- percentEncode(CanonicalizedQueryString) is the string as a result of encoding the canonicalized query string created in step 1 based on the URL encoding rules described in step 1.b.

3. Follow RFC 2104 to calculate the HMAC value of the string-to-sign by using the AccessKey secret as the key.

> **Notice** The key used for calculating the signature is your AccessKey secret followed by an ampersand (&). The SHA1 algorithm is used for the 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.

> **Notice** When the signature string is submitted to the PolarDB-X server as the final request parameter value, the string must be URL encoded like other parameters based on the rules defined in RFC 3986.

Example

DescribeDrdsInstances is used in this example. The canonicalized query string is as follows:

```
AccessKeyId=testid&Action=DescribeDrdsInstances&Format=XML&RegionId=cn-hangzhou&SignatureMethod=HMAC-SHA1&SignatureNonce=ae5bdbeb-9b44-40a1-8bb4-b40784bff686&SignatureVersion=1.0&Timestamp=2016-01-20T14%3A26%3A15Z&Version=2015-04-13
```

Encode the canonicalized query string in UTF-8 to create the following StringToSign:

```
GET&%2F&AccessKeyId%3Dtestid&Action%3DDescribeDrdsInstances&Format%3DXML&RegionId%3Dcn-hangzhou&SignatureMethod%3DHMAC-SHA1&SignatureNonce%3Dae5bdbeb-9b44-40a1-8bb4-b40784bff686&SignatureVersion%3D1.0&Timestamp%3D2016-01-20T14%253A26%253A15Z&Version%3D2015-04-13
```

Assume that the AccessKey ID is testid, the AccessKey secret is testsecret, and the key used for HMAC calculation is testsecret&. The calculated signature value is as follows:

```
h/ka/jNO+WZv8Tqgo4a75sp6eTs=
```

The signed request URL that includes the Signature parameter is as follows:

```
http://drds.example.com/?AccessKeyId=testid&Action=DescribeDrdsInstances&Format=XML&RegionId=cn-hangzhou&SignatureMethod=HMAC-SHA1&SignatureNonce=ae5bdbeb-9b44-40a1-8bb4-b40784bff686&SignatureVersion=1.0&Timestamp=2016-01-20T14%3A26%3A15Z&Version=2015-04-13&Signature=h%xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

An example of calling the DescribeDrdsInstances operation in Java is as follows:

```
public static void drdsOpenAPI() throws NoSuchAlgorithmException, InvalidKeyException, IOException {
    //AccessKey ID and AccessKey secret
    String accessKey = "testid";
    String accessSecret = "testsecret";
    //Common parameters
    Map<String, String> parameters = new TreeMap<String, String>();
    parameters.put("Format", "JSON");
    parameters.put("Action", "DescribeDrdsInstances");//Call the DescribeDrdsInstances operation
    parameters.put("Version", "2015-04-13");
    parameters.put("AccessKeyId", accessKey);
    parameters.put("SignatureMethod", "HMAC-SHA1");
```

```
        parameters.put("Timestamp", getISO8601Time());
        parameters.put("SignatureVersion", "1.0");
        parameters.put("SignatureNonce", UUID.randomUUID().toString());
        parameters.put("RegionId", "cn-hangzhou");
        StringBuilder paramStr = new StringBuilder();
        //Concatenate request parameters
        for(Map.Entry<String, String> entry : parameters.entrySet()) {
          paramStr.append(percentEncode(entry.getKey())).append("=").append(percentEncode(entry.get
Value())).append("&");
        }
        paramStr.deleteCharAt(paramStr.length()-1);
        //Calculate the signature
        StringBuilder stringToSign = new StringBuilder();
           stringToSign.append("GET").append("&").append(percentEncode("/"))
               .append("&").append(percentEncode(paramStr.toString()));
           Mac mac = Mac.getInstance("HmacSHA1");
        mac.init(new SecretKeySpec((accessSecret + "&").getBytes("UTF-8"),"HmacSHA1"));
        byte[] signData = mac.doFinal(stringToSign.toString().getBytes("UTF-8"));
        String signStr = Base64Helper.encode(signData);
        //Concatenate the request URL
        String requestUrl = "http://drds.example.com/?" + paramStr.toString() +
           "&Signature=" +  percentEncode(signStr);
        //Prepare to send HTTP requests
        URL url = new URL(requestUrl);
        HttpURLConnection httpConn = (HttpURLConnection)url.openConnection();
        httpConn.setRequestMethod("GET");
        httpConn.setDoOutput(true);
        httpConn.setDoInput(true);
        httpConn.setUseCaches(false);
        httpConn.connect();
        InputStream content = httpConn.getInputStream();
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
        byte[] buff = new byte[1024];
        while(true) {
          final int read = content.read(buff);
          if(read == -1) break;
          outputStream.write(buff,0,read);
        }
        System.out.println(new String(outputStream.toByteArray()));;
    }
```

```
  public static String percentEncode(String value) throws UnsupportedEncodingException{
      return value ! = null ? URLEncoder.encode(value, "UTF-8").replace("+", "%20")
            .replace("*", "%2A").replace("%7E", "~") : null;
    }
  static String getISO8601Time() {
      SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'");
      df.setTimeZone(new SimpleTimeZone(0, "GMT"));
      return df.format(new Date());
  }
}
```

5.1.5. Database management

5.1.5.1. ChangeAccountPassword

Resets the password of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
AccountName	String	Yes	test	The name of the account.
Action	String	Yes	ChangeAccountPassword	The operation that you want to perform. Set the value to ChangeAccountPassword.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
Password	String	Yes	Test123	The new password.
AccessKeyId	String	No	sdfgssdsf	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ChangeAccountPassword
&AccountName=test
&DrdsInstanceId=drdsxxxxxxxxxx
&Password=Test123
&<Common request parameters>
```

Sample success responses

XML format

```
<ChangeAccountPasswordResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
</ChangeAccountPasswordResponse>
```

JSON format

```
{
  "RequestId":"DSSDF-SEWE-23ERW",
  "Success":true
}
```

Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.2. CheckDrdsDbName

Checks whether the name of a PolarDB-X database is valid.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CheckDrdsDbName	The operation that you want to perform. Set the value to CheckDrdsDbName.
DbName	String	Yes	testdb	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Result	Boolean	true	Indicates whether the name of the PolarDB-X database is valid. Valid values: • true: valid • false: invalid
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CheckDrdsDbName
&DbName=testdb
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<CheckDrdsDbNameResponse>
    <Result>true</Result>
    <RequestId>CF38538C-68BD-4278-B58F-EDE96FA8A5CE</RequestId>
    <Success>true</Success>
</CheckDrdsDbNameResponse>
```

JSON format

```
{
  "Result":true,
  "RequestId":"CF38538C-68BD-4278-B58F-EDE96FA8A5CE",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.3. CheckExpandStatus

Verifies whether scale-out operations such as smooth scale-out can be performed on a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CheckExpandStatus	The operation that you want to perform. Set the value to CheckExpandStatus.
DbName	String	Yes	testdb	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The verification result.

Parameter	Type	Example	Description
IsActive	Boolean	true	Indicates whether scale-out is supported. The value true indicates that scale-out is supported. The value false indicates that scale-out is not supported.
Msg	String	success	The additional information.
RequestId	String	drds_sample_request_id	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CheckExpandStatus
&DbName=testdb
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<CheckExpandStatus>
    <Data>
        <IsActive>true</IsActive>
        <Msg></Msg>
    </Data>
    <RequestId>7CDBA7D5-8D62-4D24-9C65-510D621278DB</RequestId>
    <Success>true</Success>
</CheckExpandStatus>
```

JSON format

```
{
  "Data":{
    "IsActive":true,
    "Msg":""
  },
  "RequestId":"7CDBA7D5-8D62-4D24-9C65-510D621278DB",
  "Success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.4. CheckSqlAuditEnableStatus

Verifies whether SQL audit is enabled for a PolarDB-X logical database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CheckSqlAuditEnableStatus	The operation that you want to perform. Set the value to CheckSqlAuditEnableStatus.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
RequestId	String	drds_sample_request_id	The ID of the request.
Status	String	enabled	Indicates whether SQL audit was enabled for the logical database. Valid values: enabled and disabled.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CheckSqlAuditEnableStatus
&DbName=drds_db
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<CheckSqlAuditEnableStatus>
    <Status>"disabled"</Status>
    <RequestId>FF13E47D-4E38-4A5A-BA68-32A554E6FE10</RequestId>
    <Success>true</Success>
</CheckSqlAuditEnableStatus>
```

JSON format

```
{
  "Status":"\"disabled\"",
  "RequestId":"FF13E47D-4E38-4A5A-BA68-32A554E6FE10",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.5. CreateDrdsDB

Creates a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDrdsDB	The operation that you want to perform. Set the value to CreateDrdsDB.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
DbInstType	String	No	RDS	The type of the storage instance. Only ApsaraDB RDS for MySQL instances are supported.
DbInstanceIsCreating	Boolean	No	false	Indicates whether ApsaraDB RDS for MySQL instances involved in database creation are being created.
DbName	String	No	drds_db	The name of the PolarDB-X database.
Encode	String	No	drds_sample_encode	The database encoding.
Password	String	No	drds_sample_password	The logon password of the PolarDB-X database.
RdsInstance.N	RepeatList	No	["rds_id1", "rds_id2"]	This parameter is required only for horizontal partitioning. It is a JSON array string.
RdsSuperAccount.N.AccountName	String	No	rdsuser	The name of the ApsaraDB RDS for MySQL privileged account.

Parameter	Type	Required	Example	Description
RdsSuperAccount.N.DbInstanceId	String	No	rds_id	The ID of the ApsaraDB RDS for MySQL instance.
RdsSuperAccount.N.Password	String	No	rds_password	The password of the ApsaraDB RDS for MySQL privileged account.
Type	String	No	HORIZONTAL	The partition mode of the database. Set the value to HORIZONTAL.

Response parameters

Parameter	Type	Example	Description
RequestId	String	drds_sample_request_id	The ID of the request.
Success	Boolean	true	Indicates whether the PolarDB-X database was created.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CreateDrdsDB
&DrdsInstanceId=drdsxxxxxxxxxxx
&DbName=drds_db
&Encode=utf8mb4
&Password=drds_sample_password
&RdsInstance=["rds_id1", "rds_id2"]
&Type=HORIZONTAL
&DbInstType=RDS
&DbInstanceIsCreating=false
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateDrdsDB>
    <RequestId>FF13E47D-4E38-4A5A-BA68-32A554E610EF</RequestId>
    <Success>true</Success>
</CreateDrdsDB>
```

JSON format

```
{
  "RequestId":"FF13E47D-4E38-4A5A-BA68-32A554E610EF",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.6. CreateShardTask

Creates a partitioning task for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateShardTask	The operation that you want to perform. Set the value to CreateShardTask.
DbName	String	Yes	test_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
SourceTableName	String	Yes	student_tb1	The name of the source table.
TargetTableName	String	Yes	student_tb2	The name of the destination table.

Parameter	Type	Required	Example	Description
TaskType	String	Yes	SINGLE_TO_SHARD	The type of the task. Valid values: SHARD_TO_SINGLE, SINGLE_TO_SHARD, and SHARD_TO_SHARD.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the partitioning task was created.
RequestId	String	F8997D95-94AD-416A-AE70-E24D087045ED	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CreateShardTask
&DbName=test_db
&DrdsInstanceId=drdsxxxxxxxxxxx
&SourceTableName=student_tb1
&TargetTableName=student_tb2
&TaskType=SINGLE_TO_SHARD
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateShardTaskResponse>
    <Data>true</Data>
    <RequestId>F8997D95-94AD-416A-AE70-E24D087045ED</RequestId>
    <Success>true</Success>
</CreateShardTaskResponse>
```

JSON format

```
{
  "Data":true,
  "RequestId":"F8997D95-94AD-416A-AE70-E24D087045ED",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.7. DatalinkReplicationPrecheck

Prechecks a partitioning task of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DatalinkReplicationPrecheck	The operation that you want to perform. Set the value to DatalinkReplicationPrecheck.
DbName	String	Yes	test_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxx	The ID of the PolarDB-X instance.
DstTableName	String	Yes	student_1	The name of the destination table.
SrcTableName	String	Yes	student_2	The name of the source table.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	XXXXXX	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	Long	20190886	The ID of the precheck task.
RequestId	String	F8997D95-94AD-416A-AE70-E24D087045ED	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DatalinkReplicationPrecheck
&DbName=test_db
&DrdsInstanceId=drdsxxxxxxxx
&DstTableName=student_1
&SrcTableName=student_2
&<Common request parameters>
```

Sample success responses

JSON format

```
<DatalinkReplicationPrecheckResponse>
    <Data>20190886</Data>
    <RequestId>F8997D95-94AD-416A-AE70-E24D087045ED</RequestId>
    <Success>true</Success>
</DatalinkReplicationPrecheckResponse>
```

JSON format

```
{
  "Data":20190886,
  "RequestId":"F8997D95-94AD-416A-AE70-E24D087045ED",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.8. DeleteShardTasks

Deletes partitioning tasks from a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteShardTasks	The operation that you want to perform. Set the value to DeleteShardTasks.
DbName	String	Yes	test_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.
TableName.N.SourceTableName	String	No	student_1	The name of the source table.
TableName.N.TargetTableName	String	No	student_2	The name of the destination table.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether partitioning tasks were deleted.

Parameter	Type	Example	Description
RequestId	String	F8997D95-94AD-416A-AE70-E24D087045ED	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DeleteShardTasks
&DbName=test_db
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DeleteShardTasksResponse>
    <Data>true</Data>
    <RequestId>F8997D95-94AD-416A-AE70-E24D087045ED</RequestId>
    <Success>true</Success>
</DeleteShardTasksResponse>
```

JSON format

```
{
  "Data":true,
  "RequestId":"F8997D95-94AD-416A-AE70-E24D087045ED",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.9. DescribeBackMenu

Queries whether a PolarDB-X instance supports backup and recovery.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeBackMenu	The operation that you want to perform. Set the value to DescribeBackMenu.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	xxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
List	-	-	The list of supported items.
list	-	-	The details about a supported item.
MenuName	String	phy	• The mode of the backup. Valid values: logic: logical backup • phy: physical backup
Support	Boolean	true	Indicates whether the PolarDB-X instance supports backup and recovery.
RequestId	String	60C21BE4-EDFE-454C-95ED-3A5C7493D74A	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeBackMenu
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeBackMenuResponse>
    <List>
        <MenuName>phy</MenuName>
        <Support>true</Support>
    </List>
    <List>
        <MenuName>logic</MenuName>
        <Support>true</Support>
    </List>
    <RequestId>60C21BE4-EDFE-454C-95ED-3A5C7493D74A</RequestId>
    <Success>true</Success>
</DescribeBackMenuResponse>
```

JSON format

```
{
  "RequestId":"60C21BE4-EDFE-454C-95ED-3A5C7493D74A",
  "Success":true,
  "List":[
    {
      "MenuName":"phy",
      "Support":true
    },
    {
      "MenuName":"logic",
      "Support":true
    }
  ]
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.10. DescribeBackupDbs

Queries the databases that can be restored for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeBackupDbs	The operation that you want to perform. Set the value to DescribeBackupDbs.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	xxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
BackupId	String	No	201908367	Query by backup set.
PreferredRestoreTime	String	No	1568632541236	Query by restoration time.

Response parameters

Parameter	Type	Example	Description
DbNames	-	["db1","db2"]	The list of returned databases.
dbName	-	-	The details about a database.
RequestId	String	9EF10719-C34F-4E64-8260-D29C1A78B5DD	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeBackupDbs
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeBackupDbsResponse>
    <DbNames></DbNames>
    <RequestId>9EF10719-C34F-4E64-8260-D29C1A78B5DD</RequestId>
    <Success>true</Success>
</DescribeBackupDbsResponse>
```

JSON format

```
{
  "RequestId":"9EF10719-C34F-4E64-8260-D29C1A78B5DD",
  "DbNames":{
    "dbName":[]
  },
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.11. DescribeBackupTimes

Queries the point in time to which a PolarDB-X instance can be restored.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeBackupTimes	The operation that you want to perform. Set the value to DescribeBackupTimes.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	xxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	4780A19F-5ECB-4C56-AD20-966A3FF99772	The ID of the request.
RestoreTime	-	-	The point in time to which the PolarDB-X instance can be restored.
EndTime	String	1568636922671	The timestamp that specifies the end of the time range to query.
StartTime	String	1568632853000	The timestamp that specifies the beginning of the time range to query.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeBackupTimes
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeBackupTimesResponse>
    <RequestId>4780A19F-5ECB-4C56-AD20-966A3FF99772</RequestId>
    <Success>true</Success>
    <RestoreTime>
        <EndTime>1568636922671</EndTime>
        <StartTime>1568632853000</StartTime>
    </RestoreTime>
</DescribeBackupTimesResponse>
```

JSON format

```
{
  "RestoreTime":{
    "EndTime":"1568636922671",
    "StartTime":"1568632853000"
  },
  "RequestId":"4780A19F-5ECB-4C56-AD20-966A3FF99772",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.12. DescribeBroadcastTables

Queries the broadcast tables in a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeBroadcastTables	The operation that you want to perform. Set the value to DescribeBroadcastTables.
DbName	String	Yes	test_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
Query	String	Yes	tb1	The query content.
CurrentPage	Integer	No	1	The page number of the current page.
PageSize	Integer	No	10	The number of entries to return on each page.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
IsShard	Boolean	true	Indicates whether the table was partitioned.
List	-	-	The list of broadcast tables.
Broadcast	Boolean	true	Indicates whether the table is a broadcast table.
BroadcastType	String	1	The type of the broadcast table. Valid values: • 1: multi-write mode • 2: synchronous link mode
DbInstType	Integer	0	The type of the storage instance. The value 0 indicates RDS.
IsShard	Boolean	false	Indicates whether the table was partitioned.
Status	Integer	1	The status of the broadcast table. Valid values: • 1: activated • 2: activating • 3: activation exception

Parameter	Type	Example	Description
Table	String	nation	The name of the table.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	40	The number of entries returned per page.
RequestId	String	86E420ED-43F2-4788-A58C-9218491FFB09	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Total	Integer	2	The total number of entries.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeBroadcastTables
&DbName=test_db
&DrdsInstanceId=drdsxxxxxxxxxx
&Query=tb1
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeBroadcastTablesResponse>
    <PageSize>40</PageSize>
    <RequestId>86E420ED-43F2-4788-A58C-9218491FFB09</RequestId>
    <PageNumber>1</PageNumber>
    <Total>2</Total>
    <List>
        <Status>1</Status>
        <DbInstType>0</DbInstType>
        <IsShard>false</IsShard>
        <Table>nation</Table>
        <Broadcast>true</Broadcast>
        <BroadcastType>1</BroadcastType>
    </List>
    <List>
        <Status>1</Status>
        <DbInstType>0</DbInstType>
        <IsShard>false</IsShard>
        <Table>region</Table>
        <Broadcast>true</Broadcast>
        <BroadcastType>1</BroadcastType>
    </List>
    <Success>true</Success>
</DescribeBroadcastTablesResponse>
```

JSON format

```
{
  "PageNumber":1,
  "PageSize":40,
  "RequestId":"86E420ED-43F2-4788-A58C-9218491FFB09",
  "Success":true,
  "List":[
    {
      "Status":1,
      "IsShard":false,
      "BroadcastType":1,
      "Broadcast":true,
      "Table":"nation",
      "DbInstType":0
    },
    {
      "Status":1,
      "IsShard":false,
      "BroadcastType":1,
      "Broadcast":true,
      "Table":"region",
      "DbInstType":0
    }
  ],
  "Total":2
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.13. DescribeDbInstanceDbs

Queries the databases on a storage instance of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDbInstanceDbs	The operation that you want to perform. Set the value to DescribeDbInstanceDbs.

Parameter	Type	Required	Example	Description
DbInstanceId	String	Yes	pc-bpxxxxxxxxxxxxx	The ID of the storage instance.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	xxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud. For more information, see Common parameters.
AccountName	String	No	supper_account	The username of the privileged account. Leave this parameter blank if no privileged account exists.
DbInstType	String	No	RDS	The type of the storage instance. Valid values: POLARDB and RDS.
Password	String	No	Pass_123456	The password of the privileged account. Leave this parameter blank if no privileged account exists.

Response parameters

Parameter	Type	Example	Description
Databases	-	-	The list of returned databases.
Database	-	-	The details about a database.
DbName	String	db_newsudbv	The name of the database.

Parameter	Type	Example	Description
Description	String	Created by PolarDB-X, do NOT change password	The description of the database.
Status	Integer	1	The status of the database. • 0: creating • 1: available • 3: deleting
RequestId	String	E9F3D991-08DE-4B74-BE0E-06B80958F5DC	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Total	String	1	The total number of databases.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDbInstanceDbs
&DbInstanceId=pc-bpxxxxxxxxxxxxxx
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDbInstanceDbsResponse>
    <Databases>
        <Database>
            <Status>1</Status>
            <Description>Created by PolarDB-X, do NOT change password</Description>
            <DbName>db_newsudbv</DbName>
        </Database>
    </Databases>
    <total>1</total>
    <RequestId>E9F3D991-08DE-4B74-BE0E-06B80958F5DC</RequestId>
    <Success>true</Success>
</DescribeDbInstanceDbsResponse>
```

JSON format


```
{
  "total":1,
  "Databases":{
    "Database":[
      {
        "Status":1,
        "Description":"Created by PolarDB-X, do NOT change password",
        "DbName":"db_newsudbv"
      }
    ]
  },
  "RequestId":"E9F3D991-08DE-4B74-BE0E-06B80958F5DC",
  "Success":true
}
```


Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.14. DescribeDbInstances

Queries the storage instances, namely, ApsaraDB RDS for MySQL instances of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDbInstances	The operation that you want to perform. Set the value to DescribeDbInstances.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	xxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
DbInstType	String	No	RDS	The type of the storage instance. Set the value to RDS.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.
Search	String	No	pc-bp1xxxxxxxxxxx	Query by storage instance ID.

Response parameters

Parameter	Type	Example	Description
Items	-	-	The list of returned storage instances.
DBInstance	-	-	The details about a storage instance.
DBInstanceDescription	String	pc-bp1xxxxxxxxxxxxx	The description of the storage instance.
DBInstanceId	String	pc-bp1xxxxxxxxxxxxx	The ID of the storage instance.

Parameter	Type	Example	Description
DBInstanceStatus	Integer	1	The status of the storage instance. Valid values: • 0: creating • 1: running • 3: deleting • 5: restarting • 6: changing specifications • 7: recovering • 8: switching between the classic network and VPC network
DBInstanceType	String	RDS	The type of the storage instance. Example: RDS.
Engine	String	mysql	The engine that the storage instance runs.
EngineVersion	String	5.6	The version of the engine that the storage instance runs.
InstanceNetworkType	String	VPC	The network type of the storage instance. Valid values: • CLASSIC: classic network • VPC: Virtual Private Cloud (VPC) network Note Apsara Stack does not support VPC networks.
ReadOnlyDBInstanceId	-	[]	The list of read-only storage instances.
ReadOnlyDBInstanceId	-	-	The details about a read-only storage instance.

Parameter	Type	Example	Description
RegionId	String	cn-hangzhou	The ID of the region where the storage instance is located.
ZoneId	String	cn-hangzhou-a	The ID of the zone where the storage instance is located.
RequestId	String	05670C01-093E-4DF4-99FA-0C63FF60250D	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDbInstances
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDbInstancesResponse>
    <RequestId>05670C01-093E-4DF4-99FA-0C63FF60250D</RequestId>
    <Items>
        <DBInstance>
            <EngineVersion>5.6</EngineVersion>
            <DBInstanceStatus>1</DBInstanceStatus>
            <DBInstanceId>pc-bp1xxxxxxxxxxxxxx</DBInstanceId>
            <ReadOnlyDBInstanceId></ReadOnlyDBInstanceId>
            <DBInstanceType>Primary</DBInstanceType>
            <RegionId>cn-hangzhou</RegionId>
            <DBInstanceDescription>pc-bp1xxxxxxxxxxxxx</DBInstanceDescription>
            <Engine>POLARDB</Engine>
            <InstanceNetworkType>VPC</InstanceNetworkType>
        </DBInstance>
    </Items>
</DescribeDbInstancesResponse>
```

JSON format

```
{
  "Items":{
    "DBInstance":[
      {
        "RegionId":"cn-hangzhou",
        "DBInstanceType":"Primary",
        "DBInstanceId":"pc-bp1xxxxxxxxxxxxx",
        "InstanceNetworkType":"VPC",
        "DBInstanceDescription":"pc-bp1xxxxxxxxxxxxx",
        "EngineVersion":"5.6",
        "DBInstanceStatus":1,
        "Engine":"POLARDB",
        "ReadOnlyDBInstanceId":{
          "ReadOnlyDBInstanceId":[]
        }
      }
    ]
  },
  "RequestId":"05670C01-093E-4DF4-99FA-0C63FF60250D"
}
```


Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.15. DescribeDrdsDB

Queries the details about a database on a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsDB	The operation that you want to perform. Set the value to DescribeDrdsDB.
DbName	String	Yes	db_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxx	The ID of PolarDB-X instance.
AccessKeyId	String	No	xxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The details about the database.
CreateTime	String	1568949855000	The timestamp when the database was created.
DbInstType	String	RDS	The type of the storage instance. Example: RDS.
DbName	String	db_0	The name of the database.
InstRole	String	MASTER	The type of the storage instance. Valid values: • MASTER: primary instance. • SLAVE: read-only instance.
Mode	String	HORIZONTAL	The partition mode of the database. Set the value to HORIZONTAL.
Schema	String	db_0_xxxxxxxxxxxxwpyh	The name of a physical database after partitioning.

Parameter	Type	Example	Description
Status	String	NORMAL	The status of the database. Valid values: • TO_BE_INIT: creating • NORMAL: running
RequestId	String	05203D07-8447-48D0-BED5-2F00BFD40630	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDB
&DbName=db_test
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDBResponse>
    <RequestId>45203D07-8447-48D0-BED5-2F00BFD40630</RequestId>
    <Data>
        <Status>NORMAL</Status>
        <DbInstType>RDS</DbInstType>
        <InstRole>MASTER</InstRole>
        <Schema>db_0_xxxxxxxxxxxxwpyh</Schema>
        <CreateTime>1568949855000</CreateTime>
        <Mode>HORIZONTAL</Mode>
        <DbName>db_0</DbName>
    </Data>
    <Success>true</Success>
</DescribeDrdsDBResponse>
```

JSON format


```
{
  "Data":{
    "Status":"NORMAL",
    "DbName":"db_0",
    "CreateTime":1568949855000,
    "Schema":"db_0_xxxxxxxxxxxxwpyh",
    "DbInstType":"RDS",
    "Mode":"HORIZONTAL",
    "InstRole":"MASTER"
  },
  "RequestId":"45203D07-8447-48D0-BED5-2F00BFD40630",
  "Success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.16. DescribeDrdsDbInstance

Queries the details about an ApsaraDB RDS for MySQL instance of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsDbInstance	The operation that you want to perform. Set this parameter to DescribeDrdsDbInstance.
DbInstanceId	String	Yes	rm-bpxxxxxxxxxxx	The ID of ApsaraDB RDS for MySQL instance of the PolarDB-X instance.
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	rm-drdsxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	ASU678xxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
DbInstance	-	-	The details about the ApsaraDB RDS for MySQL instance of the PolarDB-X database.
ConnectUrl	String	rm-bp1uxxxxxxxxxxx.mysql.rds.aliyuncs.com	The connection URL.
DBInstanceId	String	rm-bp1xxxxxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance.
DBInstanceStatus	String	1	The status of the ApsaraDB RDS for MySQL instance.
DbInstType	String	Primary	The type of the ApsaraDB RDS for MySQL instance.
Engine	String	MySQL	The engine that the ApsaraDB RDS for MySQL instance runs.
EngineVersion	String	5.7	The version of the engine that the ApsaraDB RDS for MySQL instance runs.
ExpireTime	String	12341434315	The timestamp when the ApsaraDB RDS for MySQL instance expires.
NetworkType	String	VPC	The network type of the ApsaraDB RDS for MySQL instance.
PayType	String	Postpaid	The billing method of the ApsaraDB RDS for MySQL instance.

Parameter	Type	Example	Description
Port	Integer	3306	The port of the ApsaraDB RDS for MySQL instance.
RdsInstType	String	RDS	The type of the ApsaraDB RDS for MySQL instance.
ReadOnlyInstances	-	-	The list of read-only ApsaraDB RDS for MySQL instances.
ReadOnlyInstance	-	-	The details about a read-only ApsaraDB RDS for MySQL instance.
ConnectUrl	String	rm-bp1xxxxxxxxxxxxx.mysql.rds.aliyuncs.com	The connection URL.
DBInstanceId	String	rm-bp1xxxxxxxxxxxx	The ID of the read-only ApsaraDB RDS for MySQL instance.
DBInstanceStatus	String	1	The status of the read-only ApsaraDB RDS for MySQL instance.
DbInstType	String	RDS	The type of the read-only ApsaraDB RDS for MySQL instance.
Engine	String	MySQL	The engine that the read-only ApsaraDB RDS for MySQL instance runs.
EngineVersion	String	5.7	The version of the engine that the read-only ApsaraDB RDS for MySQL instance runs.
ExpireTime	String	1823487328173	The timestamp when the read-only ApsaraDB RDS for MySQL instance expires.

Parameter	Type	Example	Description
NetworkType	String	VPC	The network type of the read-only ApsaraDB RDS for MySQL instance.
PayType	String	Postpaid	The billing method of the read-only ApsaraDB RDS for MySQL instance.
Port	Integer	3306	The port of the read-only ApsaraDB RDS for MySQL instance.
RdsInstType	String	RDS	The type of the read-only ApsaraDB RDS for MySQL instance.
ReadWeight	Integer	30	The read weight of the read-only ApsaraDB RDS for MySQL instance.
RemainDays	String	0	The remaining number of days before the read-only ApsaraDB RDS for MySQL instance expires.
VersionAction	Integer	0	An invalid parameter.
ReadWeight	Integer	70	The read weight of the read-only ApsaraDB RDS for MySQL instance.
RemainDays	String	0	The remaining number of days before the read-only ApsaraDB RDS for MySQL instance expires.
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDbInstance
&DbInstanceId=rm-bp1xxxxxxxxxxxx
&DbName=drds_test
&DrdsInstanceId=rm-drdsxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDbInstance>
    <RequestId>2F7F8080-9132-4279-85D0-B7E5C4305162</RequestId>
    <DbInstance>
        <DbInstType>RDS</DbInstType>
        <ReadOnlyInstances></ReadOnlyInstances>
        <EngineVersion>5.7</EngineVersion>
        <DBInstanceStatus>1</DBInstanceStatus>
        <Port>3306</Port>
        <PayType>Postpaid</PayType>
        <ReadWeight>100</ReadWeight>
        <ConnectUrl>rm-bp1xxxxxxxxxxxx.mysql.rds.aliyuncs.com</ConnectUrl>
        <DBInstanceId>rm-bp1xxxxxxxxxxxx</DBInstanceId>
        <NetworkType>VPC</NetworkType>
        <RdsInstType>Primary</RdsInstType>
        <ExpireTime></ExpireTime>
        <RemainDays>0</RemainDays>
        <Engine>MySQL</Engine>
    </DbInstance>
    <Success>true</Success>
</DescribeDrdsDbInstance>
```

JSON format

```
{
  "DbInstance":{
    "RemainDays":0,
    "ReadOnlyInstances":{
      "ReadOnlyInstance":[]
    },
    "DBInstanceId":"rm-bp1ub71ct9skc3y",
    "ReadWeight":100,
    "RdsInstType":"Primary",
    "DbInstType":"RDS",
    "ConnectUrl":"rm-bp1ub71ct9skc3y.mysql.rds.aliyuncs.com",
    "Engine":"MySQL",
    "NetworkType":"VPC",
    "Port":3306,
    "ExpireTime":"",
    "EngineVersion":"5.7",
    "DBInstanceStatus":1,
    "PayType":"Postpaid"
  },
  "RequestId":"2F7F8080-9132-4279-85D0-B7E5C4305162",
  "Success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.17. DescribeDrdsDbInstances

Queries the ApsaraDB RDS for MySQL instances of a PolarDB-X database.

When you query the primary ApsaraDB RDS for MySQL instances of a PolarDB-X database, the ReadOnlyInstances attribute of each primary ApsaraDB RDS for MySQL instance specifies the list of read-only ApsaraDB RDS for MySQL instances used by this primary ApsaraDB RDS for MySQL instance for the current database.

Request parameters

Parameter	Type	Required	Example	Description
DbName	String	Yes	dbname	The name of the PolarDB-X database.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	ASU6xxxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	30	The number of entries to return on each page.

Response parameters

DbInstances	-	-	The list of ApsaraDB RDS for MySQL instances of the PolarDB-X database.
DbInstance	-	-	The details about a ApsaraDB RDS for MySQL instance in the PolarDB-X database.
ConnectUrl	String	rm-m5xxxxxxxxx82.mysql.rds.aliyuncs.com	The URL of the ApsaraDB RDS for MySQL instance, which is returned when the corresponding API operation of the ApsaraDB RDS for MySQL is called.
DBInstanceId	String	rm-m5xxxxxxxxx82	The ID of the ApsaraDB RDS for MySQL instance.

DBInstanceStatus	String	1	The status of the ApsaraDB RDS for MySQL instance. Valid values: • 0: creating • 1: running • 3: deleting • 5: restarting • 6: changing specifications • 7: backing up • 8: switching between classic and Virtual Private Cloud (VPC) networks • 9: migrating • 11: migrating data • 12: generating a disaster recovery instance • 13: importing data • 14: Data is being imported from other ApsaraDB RDS for MySQL instances. • 15: switching for disaster recovery • 16: creating a temporary instance • 17: creating the network • 18: cloning the instance • 19: switching the link • 20: migrating the read-only instance
DbInstType	String	RDS	The type of the ApsaraDB RDS for MySQL instance. Currently, only ApsaraDB for RDS instances are supported.

Engine	String	MySQL	The engine that the ApsaraDB RDS for MySQL instance runs. Valid value: MySQL.
EngineVersion	String	5.7	The version of the engine that the storage instance runs. Valid values: MySQL 5.7, MySQL 5.6, and MySQL 5.5.
ExpireTime	String	1237486127634	The time when the ApsaraDB RDS for MySQL instance expires.
NetworkType	String	VPC	The network type of the ApsaraDB RDS for MySQL instance. Note Apsara Stack does not support VPC networks.
Port	Integer	3306	The port of the ApsaraDB RDS for MySQL instance.
RdsInstType	String	Primary	Indicates whether the ApsaraDB RDS for MySQL instance is a primary instance or a read-only instance.
ReadOnlyInstances	-	-	The list of read-only ApsaraDB RDS for MySQL instances.
ReadOnlyInstance	-	-	The details about a read-only ApsaraDB RDS for MySQL instance.
ConnectUrl	String	rm-m5xxxxxxxxx82.mysql.rds.aliyuncs.com	The URL of the read-only ApsaraDB RDS for MySQL instance, which is returned when the corresponding API operation of the ApsaraDB RDS for MySQL is called.

DBInstanceStatus	String	1	The status of the read-only ApsaraDB RDS for MySQL instance.
DbInstType	String	RDS	The type of the read-only ApsaraDB RDS for MySQL instance.
Engine	String	MySQL	The engine that the read-only ApsaraDB RDS for MySQL instance runs.
EngineVersion	String	5.7	The version of the engine that the read-only ApsaraDB RDS for MySQL instance runs.
ExpireTime	String	123421352351234	The timestamp when the read-only ApsaraDB RDS for MySQL instance expires.
InstanceName	String	rm-m5xxxxxxxxx82	An invalid parameter.
NetworkType	String	VPC	The network type of the read-only ApsaraDB RDS for MySQL instance.
PayType	String	Postpaid	The billing method of the read-only ApsaraDB RDS for MySQL instance.
Port	Integer	3306	The port of the read-only ApsaraDB RDS for MySQL instance.
RdsInstType	String	RDS	The type of the read-only ApsaraDB RDS for MySQL instance.
ReadWeight	Integer	70	The read weight of the read-only ApsaraDB RDS for MySQL instance.
RemainDays	Integer	0	The remaining number of days before the read-only ApsaraDB RDS for MySQL instance expires.

ReadWeight	Integer	30	The read weight of the read-only ApsaraDB RDS for MySQL instance in the PolarDB-X database.
RemainDays	Integer	0	The remaining number of days before the prepaid read-only ApsaraDB RDS for MySQL instance expires.
PageNumber	String	1	The number of the page returned.
PageSize	String	10	The number of entries returned per page.
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Total	String	1	The total number of primary read-only ApsaraDB RDS for MySQL instances.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDbInstances
&DbName=dbname
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDbInstances>
    <PageSize>10</PageSize>
    <RequestId>658B6F25-442A-4996-8AAE-F77DAFBC9BAA</RequestId>
    <PageNumber>1</PageNumber>
    <Total>1</Total>
    <DbInstances>
        <DbInstance>
            <DbInstType>RDS</DbInstType>
            <ReadOnlyInstances></ReadOnlyInstances>
            <EngineVersion>5.7</EngineVersion>
            <DBInstanceStatus>1</DBInstanceStatus>
            <Port>3306</Port>
            <PayType>Postpaid</PayType>
            <ReadWeight>100</ReadWeight>
            <ConnectUrl>rm-m5xxxxxxxxx82.mysql.rds.aliyuncs.com</ConnectUrl>
            <DBInstanceId>rm-m5xxxxxxxxx82</DBInstanceId>
            <NetworkType>VPC</NetworkType>
            <RdsInstType>Primary</RdsInstType>
            <ExpireTime></ExpireTime>
            <RemainDays>0</RemainDays>
            <Engine>MySQL</Engine>
        </DbInstance>
    </DbInstances>
    <Success>true</Success>
</DescribeDrdsDbInstances>
```

JSON format

```
{
  "PageNumber":1,
  "PageSize":10,
  "RequestId":"658B6F25-442A-4996-8AAE-F77DAFBC9BAA",
  "DbInstances":{
    "DbInstance":[
      {
        "RemainDays":0,
        "ReadOnlyInstances":{
          "ReadOnlyInstance":[]
        },
        "DBInstanceId":"rm-m5xxxxxxxxx82",
        "ReadWeight":100,
        "RdsInstType":"Primary",
        "DbInstType":"RDS",
        "ConnectUrl":"rm-m5xxxxxxxxx82.mysql.rds.aliyuncs.com",
        "Engine":"MySQL",
        "NetworkType":"VPC",
        "Port":3306,
        "ExpireTime":"",
        "EngineVersion":"5.7",
        "DBInstanceStatus":1,
        "PayType":"Postpaid"
      }
    ]
  },
  "Success":true,
  "Total":1
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.18. DescribeDrdsDbRdsNameList

Queries the IDs of the storage instances of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsDbRdsNameList	The operation that you want to perform. Set the value to DescribeDrdsDbRdsNameList.
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	ASU67xxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
InstanceNameList	-	rm-m5xxxxxxxxx82	The IDs of storage instances.
InstanceName	-	-	The ID of a storage instance.
RequestId	String	37537B38-A663-45CB-A81F-E3B202BB2415	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDbRdsNameList
&DbName=drds_test
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDbRdsNameList>
    <RequestId>37537B38-A663-45CB-A81F-E3B202BB2415</RequestId>
    <InstanceNameList>
        <InstanceName>rm-m5xxxxxxxxx82</InstanceName>
    </InstanceNameList>
    <Success>true</Success>
</DescribeDrdsDbRdsNameList>
```

JSON format

```
{
  "InstanceNameList":{
    "InstanceName":[
      "rm-m5xxxxxxxxx82"
    ]
  },
  "RequestId":"37537B38-A663-45CB-A81F-E3B202BB2415",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.19. DescribeDrdsDbRdsRelationInfo

Queries the structure information of storage instances for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsDbRdsRelationInfo	The operation that you want to perform. Set the value to DescribeDrdsDbRdsRelationInfo.
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	ASU67xxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The structure information about the storage instances of the PolarDB-X database. Each item corresponds to a primary storage instance.
Data	-	-	The details about a primary storage instance of the PolarDB-X database.
RdsInstanceId	String	rm-m5xxxxxxxxx82	The ID of the primary storage instance.
ReadOnlyInstanceInfo	-	rm-m5xxxxxxxxx82	The information about read-only storage instances of the primary storage instance.
ReadOnlyInstanceIds	-	-	The IDs of the read-only storage instances.
UsedInstanceId	String	rm-m5xxxxxxxxx82	The ID of the used storage instance. For the primary PolarDB-X instance, the used storage instance is the primary storage instance. For the read-only PolarDB-X instance, the used storage instance is the optional storage instance.
UsedInstanceType	String	RDS	The type of the used storage instance.

Parameter	Type	Example	Description
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDbRdsRelationInfo
&DbName=drds_test
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDbRdsRelationInfo>
    <RequestId>E827BEE8-A8BE-4E86-992A-E6400B263CB1</RequestId>
    <Data>
        <Data>
            <ReadOnlyInstanceInfo>
                <ReadOnlyInstanceIds>rm-m5xxxxxxxxx82</ReadOnlyInstanceIds>
            </ReadOnlyInstanceInfo>
            <UsedInstanceId>rm-m5xxxxxxxxx82</UsedInstanceId>
            <UsedInstanceType>RDS</UsedInstanceType>
            <RdsInstanceId>rm-m5xxxxxxxxx82</RdsInstanceId>
        </Data>
    </Data>
    <Success>true</Success>
</DescribeDrdsDbRdsRelationInfo>
```

JSON format

```
{
  "Data":{
    "Data":[
      {
        "UsedInstanceType":"RDS",
        "RdsInstanceId":"rm-m5xxxxxxxxx82",
        "UsedInstanceId":"rm-m5xxxxxxxxx82",
        "ReadOnlyInstanceInfo":{
          "ReadOnlyInstanceIds":[
            "rm-m5xxxxxxxxx82"
          ]
        }
      }
    ]
  },
  "RequestId":"E827BEE8-A8BE-4E86-992A-E6400B263CB1",
  "Success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.20. DescribeDrdsDbTasks

Queries incomplete tasks for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsDbTasks	The operation that you want to perform. Set the value to DescribeDrdsDbTasks.
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	ASU678Uxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
TaskType	String	No	create_db	The type of the task. If this parameter is not specified, all types of tasks are queried.

Response parameters

Parameter	Type	Example	Description
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Tasks	-	-	The list of returned tasks.
Task	-	-	The details about a task.
AllowCancel	Boolean	false	Indicates whether the task can be canceled.
ExpandType	String	hot_expand	The type of the scale-out task. Valid values: smooth_expand and hot_expand.
GmtCreate	Long	123978138412341	The timestamp when the task was created.
Label	String	create_db	The type of the task.
ParentJobId	String	123214	The ID of the parent task.
Progress	Integer	99	The task progress. Valid values: 0 to 100.
ShowProgress	Boolean	true	Indicates whether the task progress was displayed.

Parameter	Type	Example	Description
TargetId	Long	12312	The ID of the incomplete task.
TaskDetail	String	{}	The details about the task. Example: the details about a smooth scale-out task.
TaskName	String	Database creation task	The name of the incomplete task.
TaskPhase	String	1	The current phase of the incomplete task.
TaskStatus	Integer	1	The status of the incomplete task. Valid values: • 0: running • 1: success • 2: failed • 3: canceled
TaskType	Integer	8	The type of the incomplete task.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDbTasks
&DbName=drds_test
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDbTasks>
    <Tasks>
        <Task>
            <AllowCancel>false</AllowCancel>
            <ExpandType></ExpandType>
            <GmtCreate>1568696422000</GmtCreate>
            <Label>create_db</Label>
            <ParentJobId>123</ParentJobId>
            <Progress>90</Progress>
            <ShowProgress>true</ShowProgress>
            <TargetId>154</TargetId>
            <TaskDetail></TaskDetail>
            <TaskName>Database creation task</TaskName>
            <TaskPhase>1</TaskPhase>
            <TaskStatus>0</TaskStatus>
            <TaskType>8</TaskType>
        </Task>
    </Tasks>
    <RequestId>7E54C57C-DF5A-4FE8-8E25-DC888817B54D</RequestId>
    <Success>true</Success>
</DescribeDrdsDbTasks>
```

JSON format

```
{
  "RequestId":"7E54C57C-DF5A-4FE8-8E25-DC888817B54D",
  "Tasks":{
    "Task":[
      {
        "ParentJobId":123,
        "AllowCancel":false,
        "GmtCreate":1568696422000,
        "TaskName":"Database creation task",
        "Progress":90,
        "TargetId":154,
        "ExpandType":"",
        "ShowProgress":true,
        "TaskDetail":"",
        "TaskPhase":1,
        "TaskStatus":0,
        "Label":"create_db",
        "TaskType":8
      }
    ]
  },
  "Success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.21. DescribeDrdsInstanceDbMonitor

Queries the performance monitoring data of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsInstanceDbMonitor	The operation that you want to perform. Set the value to DescribeDrdsInstanceDbMonitor.

Parameter	Type	Required	Example	Description
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
EndTime	Long	Yes	1568705520000	The timestamp that specifies the end of the time range to query.
Key	String	Yes	qps	The monitoring metric.
StartTime	Long	Yes	1568701920000	The timestamp that specifies the beginning of the time range to query.
AccessKeyId	String	No	ASU67xxxxxxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The list of returned monitoring data.
Key	String	qps	The monitoring metric.
Unit	String	qps	The unit of the monitoring metric.
Values	-	-	The obtained values of the monitoring metric.
Date	Long	1568705520000	The timestamp when the value was obtained.

Parameter	Type	Example	Description
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsInstanceDbMonitor
&DbName=drds_test
&DrdsInstanceId=drdsxxxxxxxxxxx
&EndTime=1568705520000
&Key=qps
&StartTime=1568701920000
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsInstanceDbMonitor>
    <RequestId>D6A88429-BA1E-4294-BE37-E82167355122</RequestId>
    <Data>
        <Values>
            <Date>1568705520000</Date>
            <Value>10000</Value>
        </Values>
        <Unit>qps</Unit>
        <Key>qps</Key>
    </Data>
    <Success>true</Success>
</DescribeDrdsInstanceDbMonitor>
```

JSON format

```
{
  "Data":[
    {
      "Values":[
        {
          "Value":10000,
          "Date":1568705520000
        }
      ],
      "Key":"qps",
      "Unit":"qps"
    }
  ],
  "RequestId":"D6A88429-BA1E-4294-BE37-E82167355122",
  "Success":true
}
```


Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.22. DescribeDrdsShardingDbs

Queries details about a horizontally partitioned PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsShardingDbs	The operation that you want to perform. Set the value to DescribeDrdsShardingDbs.

Parameter	Type	Required	Example	Description
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.
DbNamePattern	String	No	test	The matching pattern of the database name.

Response parameters

Parameter	Type	Example	Description
RequestId	String	509BDE17-505A-4B3B-854D-30D3F092502F	The ID of the request.
ShardingDbs	-	-	The list of database shards.
ShardingDb	-	-	The details about a database shard.
BlockingTimeout	Integer	5000	The length of time that a transaction waits to obtain a lock.
ConnectUrl	String	100.xx.xx.100:11569	The connection URL.
ConnectionProperties	String	connectTimeout=3000;autoReconnect=true;failOverReadOnly=false;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8	The properties of the connection string.
DbInstanceId	String	rm-bp1xxxxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance to which the database shard belongs.
DbStatus	String	running	The status of the database shard.

Parameter	Type	Example	Description
DbType	String	mysql	The type of the database shard.
GroupName	String	TEMP1_1568171495522SABE_KUP4_0000	The name of the table shard.
IdleTimeOut	Integer	30	The idle connection timeout.
MaxPoolSize	Integer	60	The maximum size of the connection pool.
MinPoolSize	Integer	5	The minimum size of the connection pool.
PreparedStatementCacheSize	Integer	0	The size of the result cache.
ShardingDbName	String	temp1_zhk1_0000	The name of the database shard.
UserName	String	pg284mi8	The username used to connect to the ApsaraDB RDS for MySQL instance.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Request parameters

```
http(s)://[Endpoint]/? Action=DescribeDrdsShardingDbs
&DbName=drds_test
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsShardingDbsResponse>
    <RequestId>509BDE17-505A-4B3B-854D-30D3F092502F</RequestId>
    <ShardingDbs>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0000</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0000</GroupName>
```

```
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0001</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0001</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0002</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0002</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
```

```
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0003</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0003</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0004</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0004</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
```

```
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0005</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0005</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0006</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0006</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
        <ShardingDb>
            <ShardingDbName>temp1_zhk1_0007</ShardingDbName>
            <GroupName>TEMP1_1568171495522SABE_KUP4_0007</GroupName>
            <MinPoolSize>5</MinPoolSize>
            <IdleTimeOut>30</IdleTimeOut>
            <DbStatus>RW</DbStatus>
```

```
            <MaxPoolSize>60</MaxPoolSize>
            <UserName>pg284mi8</UserName>
            <ConnectUrl>100.xx.xx.101:11569</ConnectUrl>
            <DbType>mysql</DbType>
            <DbInstanceId>rm-bp1hxxxxxxxxxxx</DbInstanceId>
            <BlockingTimeout>5000</BlockingTimeout>
            <PreparedStatementCacheSize>0</PreparedStatementCacheSize>
            <ConnectionProperties>connectTimeout=3000;autoReconnect=true;failOverReadOnly=false
;socketTimeout=900000;rewriteBatchedStatements=true;characterEncoding=utf8</ConnectionPropert
ies>
        </ShardingDb>
    </ShardingDbs>
    <Success>true</Success>
</DescribeDrdsShardingDbsResponse>
```

Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.23. DescribeDrdsSlowSqls

Queries slow SQL statements of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsSlowSqls	The operation that you want to perform. Set the value to DescribeDrdsSlowSqls.
DbName	String	Yes	tempdb	The name of the PolarDB-X database.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
EndTime	Long	Yes	1568289311	The end of the time range to query.
ExeTime	Long	Yes	1568289311	The SQL execution time.
StartTime	Long	Yes	1568267711	The beginning of the time range to query.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	20	The number of entries to return on each page.

Response parameters

Parameter	Type	Example	Description
Items	-	-	The list of returned slow SQL statements.
Item	-	-	The details about a slow SQL statement.
Host	String	10.xx.xx.26	The IP address of the host where the slow SQL statement is executed.
ResponseTime	Long	1568267711	The response time.
Schema	String	user	The schema of the PolarDB-X database.
SendTime	Long	1568267711	The time when the slow SQL statement was sent.

Parameter	Type	Example	Description
Sql	String	SELECT count(1) from payment_order where order_status = '08';	The content of the slow SQL statement.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	20	The number of entries returned per page.
RequestId	String	509BDE17-505A-4B3B-854D-30D3F092502F	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Total	Integer	1	The total number of slow SQL statements.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsSlowSqls
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsSlowSqlsResponse>
    <Items>
        <Item>
            <Host> 10.xx.xx.26 </Host>
            <ResponseTime>2019-09-12 13:55:00</ResponseTime>
            <Schema>user</Schema>
            <SendTime>2019-09-12 13:55:00</SendTime>
            <Sql>SELECT count(1) from  payment_order where  order_status = '08';</Sql>
        </Item>
    </Items>
    <PageNumber>1</PageNumber>
    <PageSize>30</PageSize>
    <RequestId>B2493C08-AB69-44AB-8C1D-6AB17C2CF48F</RequestId>
    <Success>true</Success>
    <Total>0</Total>
</DescribeDrdsSlowSqlsResponse>
```

Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.24. DescribeDrdsSqlAuditStatus

Queries the details about SQL audit for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsSqlAuditStatus	The operation that you want to perform. Set the value to DescribeDrdsSqlAuditStatus.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
Data	-	-	The SQL audit details about a database.
DbName	String	table1	The name of the database.
Detailed	String	test	The details about SQL audit.
Enabled	String	true	Indicates whether SQL audit was enabled for the database.
RequestId	String	DC3ABA3E-0F8A-4596-9104-F5155C34315B	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsSqlAuditStatus
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeGlobalBroadcastTypeResponse>
    <Data>
        <DbName>table1</DbName>
        <Detailed>test</Detailed>
        <Enabled>true</Enabled>
    </Data>
    <RequestId>DC3ABA3E-0F8A-4596-9104-F5155C34315B</RequestId>
    <Success>true</Success>
</DescribeGlobalBroadcastTypeResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.25. DescribeExpandLogicTableInfoList

Queries the shard keys of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeExpandLogicTableInfoList	The operation that you want to perform. Set the value to DescribeExpandLogicTableInfoList.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.

Parameter	Type	Example	Description
data	-	-	The details about shard keys.
ShardDbKey	String	id	The database shard key.
ShardTbKey	String	address	The table shard key.
TableName	String	employee_split2	The name of the table.
RequestId	String	0B6B7BDC-575D-4A77-A4F8-24B7EF027B75	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeExpandLogicTableInfoList
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeExpandLogicTableInfoListResponse>
    <Data>
        <data>
            <ShardDbKey>id</ShardDbKey>
            <TableName>employee_split2</TableName>
            <ShardTbKey>address</ShardTbKey>
        </data>
    </Data>
    <RequestId>0B6B7BDC-575D-4A77-A4F8-24B7EF027B75</RequestId>
    <Success>true</Success>
</DescribeExpandLogicTableInfoListResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.26. DescribeGlobalBroadcastType

Queries whether the table in a PolarDB-X database is a broadcast table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeGlobalBroadcastType	The operation that you want to perform. Set the value to DescribeGlobalBroadcastType.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
IsSyncType	Boolean	false	Indicates whether the table is a broadcast table.
RequestId	String	DC3ABA3E-0F8A-4596-9104-F5155C34315B	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeGlobalBroadcastType
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeGlobalBroadcastTypeResponse>
    <Data>
        <IsSyncType>false</IsSyncType>
    </Data>
    <RequestId>DC3ABA3E-0F8A-4596-9104-F5155C34315B</RequestId>
    <Success>true</Success>
</DescribeGlobalBroadcastTypeResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.27. DescribeHotDbList

Queries the available databases for hot-spot scale-out of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeHotDbList	The operation that you want to perform. Set the value to DescribeHotDbList.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
List	-	-	The list of returned items.
InstanceDb	-	-	The details about an item.

Parameter	Type	Example	Description
HotDbList	-	test,test	The list of hot-spot databases.
data	-	-	The details about a hot-spot database.
RandomCode	String	989fdsf	The random code.
Msg	String	msg	The message.
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeHotDbList
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeHotDbListResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
    <Msg>msg</Msg>
    <RandomCode>msg</RandomCode>
    <Data>
        <List>
            <HotDbList></HotDbList>
        </List>
    </Data>
</DescribeHotDbListResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.28. DescribeInstDbLogInfo

Queries the time range of historical SQL data that can be imported to a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeInstDbLogInfo	The operation that you want to perform. Set the value to DescribeInstDbLogInfo.
DbName	String	Yes	drds_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
LogTimeRange	-	-	The supported time range.
SupportLatestTime	Long	1568267711	The beginning of the time range.
SupportOldestTime	Long	1568267711	The end of the time range.
RequestId	String	DC3ABA3E-0F8A-4596-9104-F5155C34315B	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeInstDbLogInfo
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeInstDbLogInfoResponse>
    <LogTimeRange>
        <SupportLatestTime>1568267711</SupportLatestTime>
        <SupportOldestTime>1568267711</SupportOldestTime>
    </LogTimeRange>
    <RequestId>DC3ABA3E-0F8A-4596-9104-F5155C34315B</RequestId>
    <Success>true</Success>
</DescribeInstDbLogInfoResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.29. DescribeRDSPerformance

Queries the performance data of ApsaraDB RDS for MySQL instances for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeRDSPerformance	The operation that you want to perform. Set the value to DescribeRDSPerformance.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
Keys	String	Yes	cpu	The performance metrics.
RdsInstanceId	String	Yes	rm-hbxxxxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.
DbInstType	String	No	mysql	The type of the ApsaraDB RDS for MySQL database.

Parameter	Type	Required	Example	Description
EndTime	Long	No	1568267711	The end of the time range to query.
StartTime	Long	No	1568267711	The beginning of the time range to query.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
Key	String	cpuusage	The performance metric.
NodeName	String	cpu	The name of the performance metric.
NodeNum	Integer	1	The number of nodes.
Unit	String	%	The unit of the performance metric.
Values	-	-	The value set.
Date	Long	1568625038000	The time when the value was obtained.
Value	String	0.65	The value of the performance metric.
RequestId	String	F06B92ED-EF1B-4F8E-8ACA-ECEFD5E4271C	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeRDSPerformance
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeRDSPerformanceResponse>
    <Data>
        <Key>cpuusage</Key>
        <NodeName>1</NodeName>
        <Unit>%</Unit>
        <Values>
          <Date>1568625038000</Date>
          <Value>0.65</Value>
        </Values>
    </Data>
    <RequestId>DC3ABA3E-0F8A-4596-9104-F5155C34315B</RequestId>
    <Success>true</Success>
</DescribeRDSPerformanceResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.30. DescribeRdsPerformanceSummary

Queries the performance metrics of ApsaraDB RDS for MySQL instances for a PolarDB-X instance, including CPU, input/output operations per second (IOPS), and active sessions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeRdsPerformanceSummary	The operation that you want to perform. Set the value to DescribeRdsPerformanceSummary.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
RdsInstanceId.N	RepeatList	Yes	['rm-bp1xxxxxxxxxxx','rm-bp1xxxxxxxxxx']	The list of ApsaraDB RDS for MySQL instances.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
RdsPerformanceInfos	-	-	The collection of RdsPerformanceInfos objects.
ActiveSessions	Integer	0	The number of active sessions on the current ApsaraDB RDS for MySQL instance.
Cpu	Float	0.65	The CPU utilization of the ApsaraDB RDS for MySQL instance.
Iops	Float	10.56	The IOPS of the ApsaraDB RDS for MySQL instance.
RdsId	String	rm-bp1xxxxxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance.
TotalSessions	Integer	241	The total number of sessions on the current ApsaraDB RDS for MySQL instance.
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeRdsPerformanceSummary
&DrdsInstanceId=drdsxxxxxxxxxxx
&RdsInstanceId.1=['rm-bp1xxxxxxxxxxxx','rm-bp1xxxxxxxxxxx']
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeRdsPerformanceSummaryResponse>
    <code>200</code>
    <data>
        <RequestId>395533CB-12E8-4FDC-8934-127FD2024C0A</RequestId>
        <RdsPerformanceInfos>
            <TotalSessions>241</TotalSessions>
            <Cpu>0.65</Cpu>
            <Iops>10.56</Iops>
            <RdsId>rm-bp1xxxxxxxxxxx</RdsId>
            <ActiveSessions>0</ActiveSessions>
        </RdsPerformanceInfos>
        <Success>true</Success>
    </data>
    <requestId>395533CB-12E8-4FDC-8934-127FD2024C0A</requestId>
    <successResponse>true</successResponse>
</DescribeRdsPerformanceSummaryResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"395533CB-12E8-4FDC-8934-127FD2024C0A",
  "data":{
    "RequestId":"395533CB-12E8-4FDC-8934-127FD2024C0A",
    "RdsPerformanceInfos":[
      {
        "Cpu":0.65,
        "RdsId":"rm-bp1xxxxxxxxxxx",
        "TotalSessions":241,
        "Iops":10.56,
        "ActiveSessions":0
      }
    ],
    "Success":true
  },
  "code":"200"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.31. DescribeRdsSuperAccountInstances

Queries the privileged accounts of ApsaraDB RDS for MySQL instances for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeRdsSuperAccountInstances	The operation that you want to perform. Set the value to DescribeRdsSuperAccountInstances.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
RdsInstance.N	RepeatList	Yes	['rm-bp1xxxxxxxxxx','rm-bp1xxxxxxxxx']	The list of ApsaraDB RDS for MySQL instances.
AccessKeyId	String	No	****	The AccessKey ID provided to you by Alibaba Cloud.
DbInstType	String	No	RDS	The type of the database instance. Default value: RDS.

Response parameters

Parameter	Type	Example	Description
DbInstances	-	rm-bp1xxxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance.
DbInstance	-	-	The list of privileged accounts.
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeRdsSuperAccountInstances
&DrdsInstanceId=drdsxxxxxxxxxx
&RdsInstance.1=['rm-bp1xxxxxxxxxx','rm-bp1xxxxxxxxxx']
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeRdsSuperAccountInstancesResponse>
    <code>200</code>
    <data>
        <RequestId>62CA96F4-B94F-4059-B23E-A58CCF318D3E</RequestId>
        <DbInstances></DbInstances>
    </data>
    <requestId>62CA96F4-B94F-4059-B23E-A58CCF318D3E</requestId>
    <successResponse>true</successResponse>

</DescribeRdsSuperAccountInstancesResponse>
```

JSON format

```
{
   "successResponse":true,
   "requestId":"62CA96F4-B94F-4059-B23E-A58CCF318D3E",
   "data":{
      "RequestId":"62CA96F4-B94F-4059-B23E-A58CCF318D3E",
      "DbInstances":{
         "DbInstance":[]
      }
   },
   "code":"200"
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.32. DescribeRecycleBinStatus

Queries the status of the table recycle bin for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeRecycleBinStatus	The operation that you want to perform. Set the value to DescribeRecycleBinStatus.
DbName	String	Yes	tpch_db	The name of PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.

Parameter	Type	Example	Description
Status	String	disable	The status of the table recycle bin. Valid values: • disable • enable
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeRecycleBinStatus
&DbName=tpch_db
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeRecycleBinStatusResponse>
    <code>200</code>
    <data>
        <Status>enable</Status>
        <RequestId>E7D9B458-2C2D-44E3-BD55-92AD2172303F</RequestId>
        <Success>true</Success>
    </data>
    <requestId>E7D9B458-2C2D-44E3-BD55-92AD2172303F</requestId>
    <successResponse>true</successResponse>

</DescribeRecycleBinStatusResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"E7D9B458-2C2D-44E3-BD55-92AD2172303F",
  "data":{
    "Status":"enable",
    "RequestId":"E7D9B458-2C2D-44E3-BD55-92AD2172303F",
    "Success":true
  },
  "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.33. DescribeRecycleBinTables

Queries the tables that can be recycled for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeRecycleBinTables	The operation that you want to perform. Set the value to DescribeRecycleBinTables.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.

Parameter	Type	Example	Description
CreateTime	String	2019-09-16 14:42:06	The time when the table was created.
OriginalTableName	String	BIN_T4AG3CY5WWXPKHITCHJY	The original name of the table.
TableName	String	drds_params	The current name of the table.
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeRecycleBinTables
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeRecycleBinTablesResponse>
    <code>200</code>
    <data>
        <RequestId>EC985705-2D6B-4B84-9732-BF8F5DCA247A</RequestId>
        <Data>
            <OriginalTableName>drds_params</OriginalTableName>
            <TableName>BIN_T4AG3CY5WWXPKHITCHJY</TableName>
            <CreateTime>2019-09-16 14:42:06</CreateTime>
        </Data>
        <Success>true</Success>
    </data>
    <requestId>EC985705-2D6B-4B84-9732-BF8F5DCA247A</requestId>
    <successResponse>true</successResponse>

</DescribeRecycleBinTablesResponse>
```

JSON format


```
{
  "successResponse":true,
  "requestId":"EC985705-2D6B-4B84-9732-BF8F5DCA247A",
  "data":{
    "Data":[
      {
        "OriginalTableName":"drds_params",
        "CreateTime":"2019-09-16 14:42:06",
        "TableName":"BIN_T4AG3CY5WWXPKHITCHJY"
      }
    ],
    "RequestId":"EC985705-2D6B-4B84-9732-BF8F5DCA247A",
    "Success":true
  },
  "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.34. DescribeShardTaskInfo

Queries details about a partitioning task of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeShardTaskInfo	The operation that you want to perform. Set the value to DescribeShardTaskInfo.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
SourceTableName	String	Yes	a1	The name of the source table.

Parameter	Type	Required	Example	Description
TargetTableName	String	Yes	a2	The name of the destination table.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
Expired	String	0	The number of remaining days before the task expires.
Full	-	-	The full migration task.
Expired	Integer	0	The number of remaining days before the full migration task expires.
Progress	Integer	100	The full migration progress.
StartTime	String	2019-09-16 15:11:53	The start time of the full migration task.
Total	Integer	2	The total number of full migration tasks.
Tps	Integer	0	The current TPS of the full migration task.
FullCheck	-	-	The full check task.
Expired	Integer	0	The number of remaining days before the full check task expires.
Progress	Integer	0	The full check progress.
StartTime	String	2019-09-16 15:12:53	The start time of the full check task.

Parameter	Type	Example	Description
Total	Integer	0	The total number of full check tasks.
Tps	Integer	0	The current TPS of the full check task.
FullRevise	-	-	The full correction task.
Expired	Integer	0	The number of remaining days before the full correction task expires.
Progress	Integer	0	The full correction progress.
StartTime	String	2019-09-16 15:12:53	The start time of the full correction task.
Total	Integer	0	The total number of full correction tasks.
Tps	Integer	0	The current TPS of the full correction task.
Increment	-	-	The increment.
Delay	Integer	1568617906	The latency of the incremental synchronization.
StartTime	String	2019-09-16 15:12:53	The start time of the incremental synchronization.
Tps	Integer	0	The current TPS of the incremental synchronization.
Progress	String	0	The incremental synchronization progress.
Review	-	-	The review task.
Expired	Integer	0	The number of remaining days before the review task expires.
Progress	Integer	0	The review progress.

Parameter	Type	Example	Description
StartTime	String	2019-09-16 15:12:53	The start time of the review task.
Total	Integer	0	The total number of review tasks.
Tps	Integer	0	The current TPS of the review task.
SourceTableName	String	a1	The name of the source table.
Stage	String	2	The current stage of the task.
Status	String	Incremental synchronization	The status of the task.
TargetTableName	String	a2	The name of the destination table.
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeShardTaskInfo
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeShardTaskInfoResponse>
    <code>200</code>
    <data>
        <RequestId>42A9A85B-60F3-4858-88CD-2EE75251D19F</RequestId>
        <Data>
            <Status>Incremental synchronization</Status>
            <SourceTableName>a1</SourceTableName>
            <Progress>0</Progress>
```

```
            <Progress>0</Progress>
            <FullCheck>
                <Progress>0</Progress>
                <Tps>0</Tps>
                <Total>0</Total>
                <Expired>0</Expired>
            </FullCheck>
            <Increment>
                <Tps>0</Tps>
                <StartTime>2019-09-16 15:12:53</StartTime>
                <Delay>1568617906</Delay>
            </Increment>
            <FullRevise>
                <Progress>0</Progress>
                <Tps>0</Tps>
                <Total>0</Total>
                <Expired>0</Expired>
            </FullRevise>
            <Stage>2</Stage>
            <Expired>0</Expired>
            <Review>
                <Progress>0</Progress>
                <Tps>0</Tps>
                <Total>0</Total>
                <Expired>0</Expired>
            </Review>
            <TargetTableName>a2</TargetTableName>
            <Full>
                <Progress>100</Progress>
                <Tps>0</Tps>
                <Total>2</Total>
                <StartTime>2019-09-16 15:11:53</StartTime>
                <Expired>0</Expired>
            </Full>
        </Data>
        <Success>true</Success>
    </data>
    <requestId>42A9A85B-60F3-4858-88CD-2EE75251D19F</requestId>
    <successResponse>true</successResponse>
</DescribeShardTaskInfoResponse>
```

JSON format


```
{
  "successResponse":true,
  "requestId":"42A9A85B-60F3-4858-88CD-2EE75251D19F",
  "data":{
    "Data":{
      "Status":"Incremental synchronization",
      "FullRevise":{
        "Tps":0,
        "Expired":0,
        "Total":0,
        "Progress":0
      },
      "Full":{
        "Tps":0,
        "Expired":0,
        "StartTime":"2019-09-16 15:11:53",
        "Total":2,
        "Progress":100
      },
      "Increment":{
        "Tps":0,
        "Delay":1568617906,
        "StartTime":"2019-09-16 15:12:53"
      },
      "Expired":"0",
      "SourceTableName":"a1",
      "FullCheck":{
        "Tps":0,
        "Expired":0,
        "Total":0,
        "Progress":0
      },
      "Review":{
        "Tps":0,
        "Expired":0,
        "Total":0,
        "Progress":0
      },
      "TargetTableName":"a2",
```

```
        "Stage":"2",
        "Progress":"0"
      },
      "RequestId":"42A9A85B-60F3-4858-88CD-2EE75251D19F",
      "Success":true
   },
   "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.35. DescribeShardTaskList

Queries the partitioning tasks of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeShardTaskList	The operation that you want to perform. Set the value to DescribeShardTaskList.
DbName	String	Yes	testdb	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
Query	String	Yes	a1	The query criteria.
TaskType	String	Yes	SINGLE_TO_SHARD	The type of the task.
CurrentPage	Integer	No	1	The page number of the current page.
PageSize	Integer	No	10	The number of entries to return on each page.

Parameter	Type	Required	Example	Description
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
List	-	-	The list of returned objects.
Delay	Integer	0	The latency of the task.
Expired	String	0	The time when the task expired.
Expired	Long	0	The number of remaining days before the task expires.
Progress	Integer	0	The progress of the task.
SourceTableName	String	a1	The name of the source table.
Stage	Integer	1	The current stage of the task.
TargetTableName	String	a2	The name of the destination table.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
RequestId	String	890BB376-5746-4145-A088-A455F8D359D0	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.
Total	Integer	1	The total number of tables.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeShardTaskList
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeShardTaskListResponse>
    <code>200</code>
    <data>
        <PageSize>10</PageSize>
        <RequestId>890BB376-5746-4145-A088-A455F8D359D0</RequestId>
        <PageNumber>1</PageNumber>
        <Total>1</Total>
        <List>
            <SourceTableName>a1</SourceTableName>
            <Progress>0</Progress>
            <Stage>1</Stage>
            <Expired>0</Expired>
            <TargetTableName>a2</TargetTableName>
            <Delay>0</Delay>
        </List>
        <Success>true</Success>
    </data>
    <requestId>890BB376-5746-4145-A088-A455F8D359D0</requestId>
    <successResponse>true</successResponse>
</DescribeShardTaskListResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"890BB376-5746-4145-A088-A455F8D359D0",
  "data":{
    "PageNumber":1,
    "PageSize":10,
    "RequestId":"890BB376-5746-4145-A088-A455F8D359D0",
    "Success":true,
    "List":[
      {
        "Delay":0,
        "Expired":0,
        "SourceTableName":"a1",
        "TargetTableName":"a2",
        "Stage":1,
        "Progress":0
      }
    ],
    "Total":1
  },
  "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.36. DescribeSqlFlashbakTask

Queries the flashback tasks of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeSqlFlashbakTask	The operation that you want to perform. Set the value to DescribeSqlFlashbakTask.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	*****	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
SqlFlashbackTasks	-	-	The list of returned flashback tasks.
SqlFlashbackTask	-	-	The details about a flashback task.
DbName	String	drds_flashback	The name of the PolarDB-X database for the flashback task.
DownloadUrl	String	http://...	The download URL.
ExpireTime	Long	1569216270000	The time when the download URL expires.
GmtCreate	Long	1568611408000	The time when the task was created.
GmtModified	Long	1568611469000	The time when the task was modified.
Id	Long	238	The ID of the primary key.
InstId	String	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
RecallProgress	Integer	10	The progress of the reverse call.
RecallRestoreType	Integer	0	The recovery type. Valid values: image restoration and reverse restoration.
RecallStatus	Integer	1	The status of the data recall task.

Parameter	Type	Example	Description
RecallType	Integer	1	The recall type. Valid values: exact match and fuzzy match.
SearchEndTime	Long	1568609597000	The end time of the query.
SearchStartTime	Long	1568609597000	The start time of the query.
SqlCounter	Long	0	The number of restored data rows.
SqlPk	String	id	The value of the SQL primary key.
SqlType	String	Insert,Update,Delete	The type of the SQL statement.
TableName	String	drds_params	The name of the table to which data belongs.
TraceId	String	trace	The trace ID of the SQL statement.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeSqlFlashbakTask
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeSqlFlashbakTaskResponse>
    <code>200</code>
    <data>
        <SqlFlashbackTasks>
            <SqlFlashbackTask>
                <GmtModified>1568611469000</GmtModified>
                <SearchEndTime>1568611397000</SearchEndTime>
                <RecallStatus>2</RecallStatus>
                <SqlPk></SqlPk>
                <RecallRestoreType>0</RecallRestoreType>
                <TableName>drds_params</TableName>
                <GmtCreate>1568611408000</GmtCreate>
                <SearchStartTime>1568609597000</SearchStartTime>
                <InstId>drdsxxxxxxxxxxx</InstId>
                <RecallType>1</RecallType>
                <TraceId></TraceId>
                <DbName>drds_flashback</DbName>
                <SqlType>Insert,Update,Delete</SqlType>
                <Id>238</Id>
                <ExpireTime>1569216270000</ExpireTime>
                <SqlCounter>0</SqlCounter>
                <RecallProgress>100</RecallProgress>
                <DownloadUrl></DownloadUrl>
            </SqlFlashbackTask>
        </SqlFlashbackTasks>
        <RequestId>11449CF1-6D32-4759-8F93-EAF36F176C41</RequestId>
        <Success>true</Success>
    </data>
    <requestId>11449CF1-6D32-4759-8F93-EAF36F176C41</requestId>
    <successResponse>true</successResponse>
</DescribeSqlFlashbakTaskResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"11449CF1-6D32-4759-8F93-EAF36F176C41",
  "data":{
    "RequestId":"11449CF1-6D32-4759-8F93-EAF36F176C41",
    "SqlFlashbackTasks":{
      "SqlFlashbackTask":[
        {
          "SearchStartTime":1568609597000,
          "DbName":"drds_flashback",
          "RecallStatus":2,
          "InstId":"drdsxxxxxxxxxxx",
          "SqlType":"Insert,Update,Delete",
          "SqlPk":"",
          "RecallProgress":100,
          "GmtCreate":1568611408000,
          "GmtModified":1568611469000,
          "RecallRestoreType":0,
          "ExpireTime":1569216270000,
          "DownloadUrl":"",
          "TraceId":"",
          "Id":238,
          "TableName":"drds_params",
          "SqlCounter":0,
          "SearchEndTime":1568611397000,
          "RecallType":1
        }
      ]
    },
    "Success":true
  },
  "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.37. DescribeTable

Queries the table schema of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeTable	The operation that you want to perform. Set the value to DescribeTable.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxx	The ID of the PolarDB-X instance.
TableName	String	Yes	employee_split	The name of the table.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
List	-	-	The details about the table structure.
ColumnName	String	Id	The name of the column.
ColumnType	String	bigint(20)	The type of the column.
Extra	String	auto_increment	The extra information.
Index	String	PRI	The primary key.
IsAllowNull	String	NO	Indicates whether the column can be empty.
IsPk	String	YES	Indicates whether it is a primary key.

Parameter	Type	Example	Description
RequestId	String	5DC75BC2-3F3F-4FE5-8DC9-2A808B9AA401	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeTable
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeTableResponse>
    <code>200</code>
    <data>
        <RequestId>5DC75BC2-3F3F-4FE5-8DC9-2A808B9AA401</RequestId>
        <Data>
            <List>
                <ColumnName>Id</ColumnName>
                <ColumnType>bigint(20)</ColumnType>
                <Extra>auto_increment</Extra>
                <IsAllowNull>NO</IsAllowNull>
                <IsPk>YES</IsPk>
                <Index>PRI</Index>
            </List>
            <List>
                <ColumnName>Name</ColumnName>
                <ColumnType>varchar(20)</ColumnType>
                <Extra></Extra>
                <IsAllowNull>NO</IsAllowNull>
                <IsPk>NO</IsPk>
                <Index></Index>
            </List>
            <List>
                <ColumnName>Address</ColumnName>
                <ColumnType>varchar(20)</ColumnType>
                <Extra></Extra>
                <IsAllowNull>NO</IsAllowNull>
                <IsPk>NO</IsPk>
                <Index></Index>
            </List>
        </Data>
        <Success>true</Success>
    </data>
    <requestId>5DC75BC2-3F3F-4FE5-8DC9-2A808B9AA401</requestId>
    <successResponse>true</successResponse>
</DescribeTableResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"5DC75BC2-3F3F-4FE5-8DC9-2A808B9AA401",
  "data":{
    "Data":{
      "List":[
        {
          "IsPk":"YES",
          "Extra":"auto_increment",
          "Index":"PRI",
          "IsAllowNull":"NO",
          "ColumnType":"bigint(20)",
          "ColumnName":"Id"
        },
        {
          "IsPk":"NO",
          "Extra":"",
          "Index":"",
          "IsAllowNull":"NO",
          "ColumnType":"varchar(20)",
          "ColumnName":"Name"
        },
        {
          "IsPk":"NO",
          "Extra":"",
          "Index":"",
          "IsAllowNull":"NO",
          "ColumnType":"varchar(20)",
          "ColumnName":"Address"
        }
      ]
    },
    "RequestId":"5DC75BC2-3F3F-4FE5-8DC9-2A808B9AA401",
    "Success":true
  },
  "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.38. DescribeTableListByType

Queries type-specific tables of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeTableListByType	The operation that you want to perform. Set the value to DescribeTableListByType.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
Query	String	Yes	drdsxxxxxxxxxxx	The search field.
TableType	String	Yes	SINGLE	The type of the table.
CurrentPage	Integer	No	1	The page number of the current page.
PageSize	Integer	No	10	The number of entries to return on each page.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
List	-	-	The list of returned tables.
Property	String	single	The property.
TableName	String	employee_split	The name of the table.

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	B360F47B-59E3-4D1C-BA03-6BFB1C993F88	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.
Total	Integer	1	The total number of tables.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeTableListByType
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeTableListByTypeResponse>
    <code>200</code>
    <data>
        <PageSize>10</PageSize>
        <RequestId>B360F47B-59E3-4D1C-BA03-6BFB1C993F88</RequestId>
        <PageNumber>1</PageNumber>
        <Total>1</Total>
        <List>
            <TableName>employee_split</TableName>
            <Property>single</Property>
        </List>
        <Success>true</Success>
    </data>
    <requestId>B360F47B-59E3-4D1C-BA03-6BFB1C993F88</requestId>
    <successResponse>true</successResponse>

</DescribeTableListByTypeResponse>
```

JSON format

```
{
   "successResponse":true,
   "requestId":"B360F47B-59E3-4D1C-BA03-6BFB1C993F88",
   "data":{
      "PageNumber":1,
      "PageSize":10,
      "RequestId":"B360F47B-59E3-4D1C-BA03-6BFB1C993F88",
      "Success":true,
      "List":[
         {
            "TableName":"employee_split",
            "Property":"single"
         }
      ],
      "Total":1
   },
   "code":"200"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.39. DescribeTables

Queries the details about tables in a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeTables	The operation that you want to perform. Set the value to DescribeTables.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
Query	String	Yes	drdsxxxxxxxxxxx	The query criteria.
CurrentPage	Integer	No	1	The page number of the current page.
PageSize	Integer	No	10	The number of entries to return on each page.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
List	-	-	The list of returned objects.
Broadcast	Boolean	false	Indicates whether the table is a broadcast table.
DbInstType	Integer	0	The type of the database instance.

Parameter	Type	Example	Description
IsLocked	Boolean	false	Indicates whether the table was locked.
IsShard	Boolean	false	Indicates whether the table was partitioned.
ShardKey	String	null	The shard key of the table.
Table	String	employee_split	The name of the table.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
RequestId	String	83AC3D7E-461C-4D87-8ACD-6CC295178C6F	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.
Total	Integer	1	The total number of tables.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeTables
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeTablesResponse>
    <code>200</code>
    <data>
        <PageSize>40</PageSize>
        <RequestId>83AC3D7E-461C-4D87-8ACD-6CC295178C6F</RequestId>
        <PageNumber>1</PageNumber>
        <Total>2</Total>
        <List>
            <DbInstType>0</DbInstType>
            <IsShard>false</IsShard>
            <Table>employee_split</Table>
            <Broadcast>false</Broadcast>
            <ShardKey>null</ShardKey>
            <IsLocked>false</IsLocked>
            <AllowFullTableScan>false</AllowFullTableScan>
        </List>
        <List>
            <DbInstType>0</DbInstType>
            <IsShard>true</IsShard>
            <Table>employee_split2</Table>
            <Broadcast>false</Broadcast>
            <ShardKey>["id"]</ShardKey>
            <IsLocked>false</IsLocked>
            <AllowFullTableScan>true</AllowFullTableScan>
        </List>
        <Success>true</Success>
    </data>
    <requestId>83AC3D7E-461C-4D87-8ACD-6CC295178C6F</requestId>
    <successResponse>true</successResponse>
</DescribeTablesResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"83AC3D7E-461C-4D87-8ACD-6CC295178C6F",
  "data":{
    "PageNumber":1,
    "PageSize":40,
    "RequestId":"83AC3D7E-461C-4D87-8ACD-6CC295178C6F",
    "Success":true,
    "List":[
      {
        "IsLocked":false,
        "IsShard":false,
        "ShardKey":"null",
        "Broadcast":false,
        "Table":"employee_split",
        "DbInstType":0,
        "AllowFullTableScan":false
      },
      {
        "IsLocked":false,
        "IsShard":true,
        "ShardKey":"[\"id\"]",
        "Broadcast":false,
        "Table":"employee_split2",
        "DbInstType":0,
        "AllowFullTableScan":true
      }
    ],
    "Total":2
  },
  "code":"200"
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.40. DisableSqlAudit

Disables SQL audit for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DisableSqlAudit	The operation that you want to perform. Set the value to DisableSqlAudit.
DbName	String	Yes	drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx x	The ID of the PolarDB-X instance.
AccessKeyId	String	No	*****	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0C B908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Result	Boolean	true	Indicates whether SQL audit was disabled for the PolarDB-X database.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DisableSqlAudit
&<Common request parameters>
```

Sample success responses

XML format

```
<DisableSqlAuditResponse>
    <code>200</code>
    <data>
        <RequestId>5F72CB0C-7A49-458E-AAC7-F53BCDAACEEF</RequestId>
        <Success>true</Success>
        <Result>true</Result>
    </data>
    <requestId>5F72CB0C-7A49-458E-AAC7-F53BCDAACEEF</requestId>
    <successResponse>true</successResponse>

</DisableSqlAuditResponse>
```

JSON format

```
{
   "successResponse":true,
   "requestId":"5F72CB0C-7A49-458E-AAC7-F53BCDAACEEF",
   "data":{
      "Result":true,
      "RequestId":"5F72CB0C-7A49-458E-AAC7-F53BCDAACEEF",
      "Success":true
   },
   "code":"200"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.41. ModifyAccountPrivilege

Modifies account permissions for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
AccountName	String	Yes	mytest@%	The name of the account.

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyAccountPrivilege	The operation that you want to perform. Set the value to ModifyAccountPrivilege.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	2zAEvsxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
DbPrivilege.N.DbName	String	No	ddltest	The name of the PolarDB-X database whose permissions will be changed.
DbPrivilege.N.Privilege	String	No	RW	The database permissions that the account has.

Response parameters

Parameter	Type	Example	Description
RequestId	String	40E313AD-3B7C-4E34-B3BF-58AD0174D638	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyAccountPrivilege
&AccountName=mytest@%
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifyAccountPrivilegeResponse>
    <RequestId>40E313AD-3B7C-4E34-B3BF-58AD0174D638</RequestId>
    <Success>true</Success>
</ModifyAccountPrivilegeResponse>
```

JSON format

c

```
{
  "RequestId":"40E313AD-3B7C-4E34-B3BF-58AD0174D638",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.42. ModifyDrdsIpWhiteList

Modifies the IP address whitelist of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyDrdsIpWhiteList	The operation that you want to perform. Set the value to ModifyDrdsIpWhiteList.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
IpWhiteList	String	Yes	127.xx.xx.1,10.xx.xx.1	The IP address whitelist after modification. Note Use commas (,) to separate multiple IP addresses.
AccessKeyId	String	No	LTAIxxxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
GroupAttribute	String	No	Group 1	The attribute of an IP address group in the whitelist.
GroupName	String	No	drds_group	The name of an IP address group in the whitelist.
Mode	Boolean	No	false	• The modification mode. Valid values: true: appending • false: overwriting

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyDrdsIpWhiteList
&DbName=test
&DrdsInstanceId=drdsxxxxxxxxxx
&IpWhiteList=127.xx.xx.1,10.xx.xx.1
&<Common request parameters>
```

Sample success responses

JSON format

```
{
   "successResponse":true,
   "requestId":"CAC59657-4B81-478E-BB68-6C8FF8FFDF22",
   "data":{
      "RequestId":"CAC59657-4B81-478E-BB68-6C8FF8FFDF22",
      "Success":true
   },
   "code":"200"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.43. ModifyRdsReadWeight

Modifies the read weights of ApsaraDB RDS for MySQL instances.

Request parameters

Parameter	Type	Required	Example	Description
DbName	String	Yes	db_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
InstanceNames	String	Yes	rm-bp1xxxxxxxxxxxx,rm-bp1xxxxxxxxxxxx	The list of ApsaraDB RDS for MySQL instances. Use commas (,) to separate multiple RDS instances.

Parameter	Type	Required	Example	Description
Weights	String	Yes	14,86	The weight of each ApsaraDB RDS for MySQL instance. Use commas (,) to separate multiple weights.
AccessKeyId	String	No	LTAI8Gxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyRdsReadWeight
&DbName=db_test
&DrdsInstanceId=drdsxxxxxxxxxx
&InstanceNames=rm-bp1qxxxxxxxxxx,rm-bp1xxxxxxxxxxx
&Weights=14,86
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifyRdsReadWeightResponse>
    <RequestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</RequestId>
    <Success>true</Success>
</ModifyRdsReadWeightResponse>
```

JSON format

```
{
  "RequestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
  "Success":"true"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.44. DescribeDbInstances

Queries the storage instances, namely, ApsaraDB RDS for MySQL instances of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDbInstances	The operation that you want to perform. Set the value to DescribeDbInstances.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	xxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
DbInstType	String	No	RDS	The type of the storage instance. Set the value to RDS.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page.
Search	String	No	pc-bp1xxxxxxxxxxx	Query by storage instance ID.

Response parameters

Parameter	Type	Example	Description
Items	-	-	The list of returned storage instances.
DBInstance	-	-	The details about a storage instance.
DBInstanceDescription	String	pc-bp1xxxxxxxxxxxxxx	The description of the storage instance.
DBInstanceId	String	pc-bp1xxxxxxxxxxxxxx	The ID of the storage instance.
DBInstanceStatus	Integer	1	The status of the storage instance. Valid values: • 0: creating • 1: running • 3: deleting • 5: restarting • 6: changing specifications • 7: recovering • 8: switching between the classic network and VPC network
DBInstanceType	String	RDS	The type of the storage instance. Example: RDS.
Engine	String	mysql	The engine that the storage instance runs.
EngineVersion	String	5.6	The version of the engine that the storage instance runs.

Parameter	Type	Example	Description
InstanceNetworkType	String	VPC	The network type of the storage instance. Valid values: • CLASSIC: classic network • VPC: Virtual Private Cloud (VPC) network Note Apsara Stack does not support VPC networks.
ReadOnlyDBInstanceId	-	[]	The list of read-only storage instances.
ReadOnlyDBInstanceId	-	-	The details about a read-only storage instance.
RegionId	String	cn-hangzhou	The ID of the region where the storage instance is located.
ZoneId	String	cn-hangzhou-a	The ID of the zone where the storage instance is located.
RequestId	String	05670C01-093E-4DF4-99FA-0C63FF60250D	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDbInstances
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDbInstancesResponse>
    <RequestId>05670C01-093E-4DF4-99FA-0C63FF60250D</RequestId>
    <Items>
        <DBInstance>
            <EngineVersion>5.6</EngineVersion>
            <DBInstanceStatus>1</DBInstanceStatus>
            <DBInstanceId>pc-bp1xxxxxxxxxxxxxx</DBInstanceId>
            <ReadOnlyDBInstanceId></ReadOnlyDBInstanceId>
            <DBInstanceType>Primary</DBInstanceType>
            <RegionId>cn-hangzhou</RegionId>
            <DBInstanceDescription>pc-bp1xxxxxxxxxxxxx</DBInstanceDescription>
            <Engine>POLARDB</Engine>
            <InstanceNetworkType>VPC</InstanceNetworkType>
        </DBInstance>
    </Items>
</DescribeDbInstancesResponse>
```

JSON format

```
{
  "Items":{
    "DBInstance":[
      {
        "RegionId":"cn-hangzhou",
        "DBInstanceType":"Primary",
        "DBInstanceId":"pc-bp1xxxxxxxxxxxxx",
        "InstanceNetworkType":"VPC",
        "DBInstanceDescription":"pc-bp1xxxxxxxxxxxxx",
        "EngineVersion":"5.6",
        "DBInstanceStatus":1,
        "Engine":"POLARDB",
        "ReadOnlyDBInstanceId":{
          "ReadOnlyDBInstanceId":[]
        }
      }
    ]
  },
  "RequestId":"05670C01-093E-4DF4-99FA-0C63FF60250D"
}
```

Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.5.45. ReleaseHiStoreInstance

Releases a columnar storage instance, namely, a HiStore instance of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
HistoreInstanceId	String	Yes	hi-hbgxxxxxxxxx	The ID of the HiStore instance.
RegionId	String	No	cn-beijing	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ReleaseHiStoreInstance
&DrdsInstanceId=drdsxxxxxxxxxx
&HistoreInstanceId=hi-hbgxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<ReleaseHiStoreInstanceResponse>
    <RequestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</RequestId>
    <Success>true</Success>
</ReleaseHiStoreInstanceResponse>
```

JSON format

```
{
  "RequestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
  "Success":"true"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.46. RemoveBackupsSet

Removes a backup set from a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveBackupsSet	The operation that you want to perform. Set the value to RemoveBackupsSet.
BackupId	String	Yes	c778036f-d464-11e9-8d45-7370dca6206b	The ID of the backup set.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	LTAI8xxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Result	String	success	Indicates whether the backup set was removed.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RemoveBackupsSet
&BackupId=c778036f-d464-11e9-8d45-7370dca6206b
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<RemoveBackupsSetResponse>
    <RequestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</RequestId>
    <Success>true</Success>
</RemoveBackupsSetResponse>
```

JSON format

```
{
  "RequestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
  "Success":"true"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.47. RemoveDrdsDbFailedRecord

Removes database creation failure records from a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveDrdsDbFailedRecord	The operation that you want to perform. Set the value to RemoveDrdsDbFailedRecord.
DbName	String	Yes	db_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	LTAI8Gxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The ID of the request.
Result	Boolean	true	Indicates whether the database creation failure records were removed from the PolarDB-X instance.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RemoveDrdsDbFailedRecord
&DbName=db_test
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<RemoveDrdsDbFailedRecordResponseponse>
    <code>200</code>
    <data>
        <RequestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</RequestId>
        <Success>true</Success>
        <Result>true</Result>
    </data>
    <requestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</requestId>
    <successResponse>true</successResponse>
</RemoveDrdsDbFailedRecordResponseponse>
```

JSON format

```
{
   "successResponse":true,
   "requestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
   "data":{
      "Result":true,
      "RequestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
      "Success":true
   },
   "code":"200"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.48. RemoveRecycleBinTable

Removes tables from the recycle bin for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveRecycleBinTable	The operation that you want to perform. Set the value to RemoveRecycleBinTable.
DbName	String	Yes	my_drds_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
TableName	String	Yes	buy_order	The name of the PolarDB-X logical table.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether tables were removed from the table recycle bin.
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RemoveRecycleBinTable
&DbName=my_drds_db
&DrdsInstanceId=drdsxxxxxxxxxx
&TableName=buy_order
&<Common request parameters>
```

Sample success responses

XML format

```
<RemoveRecycleBinTableResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Data>true</Data>
</RemoveRecycleBinTableResponse>
```

JSON format

```
{
  "Data":true,
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.49. ResetDrdsToRdsConnections

Resets the number of connections from the PolarDB-X instance to ApsaraDB RDS for MySQL instances.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ResetDrdsToRdsConnections	The operation that you want to perform. Set the value to ResetDrdsToRdsConnections.
DbName	String	Yes	test_my_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	fab3xxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Result	String	true	Indicates whether the number of connections from the PolarDB-X instance to ApsaraDB RDS for MySQL instances was reset.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ResetDrdsToRdsConnections
&DbName=test_my_db
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<ResetDrdsToRdsConnectionsResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Result>true</Result>
</ResetDrdsToRdsConnectionsResponse>
```

JSON format

```
{
  "Result":true,
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.50. SetupBroadcastTables

Configures a broadcast table for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetupBroadcastTables	The operation that you want to perform. Set the value to SetupBroadcastTables.
Active	Boolean	Yes	true	Specifies whether to turn on the broadcast table switch.
DbName	String	Yes	test_db	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
TableName.N	RepeatList	Yes	tb1	The name of the PolarDB-X table.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SetupBroadcastTables
&Active=true
&DbName=test_db
&DrdsInstanceId=drdsxxxxxxxxxxxxx
&TableName.1=tb1
&<Common request parameters>
```

Sample success responses

XML format

```
<SetupBroadcastTablesResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Data>true</Data>
</SetupBroadcastTablesResponse>
```

JSON format

```
{
  "Data":true,
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.51. SetupRecycleBinStatus

Enables the table recycle bin for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetupRecycleBinStatus	The operation that you want to perform. Set the value to SetupRecycleBinStatus.
DbName	String	Yes	db1	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
StatusAction	String	Yes	enable	Specifies whether to enable the recycle bin. Valid values: • enable • disable

Parameter	Type	Required	Example	Description
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The returned results.
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SetupRecycleBinStatus
&DbName=db1
&DrdsInstanceId=drdsxxxxxxxxxxxx
&StatusAction=enable
&<Common request parameters>
```

Sample success responses

XML format

```
<SetupRecycleBinStatusResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Data>true</Data>
</SetupRecycleBinStatusResponse>
```

JSON format

```
{
  "Data":"true",
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.52. SetupTable

Enables full table scan for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetupTable	The operation that you want to perform. Set the value to SetupTable.
AllowFullTableScan	Boolean	Yes	true	Specifies whether to enable full table scan.
DbName	String	Yes	dbtest	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
TableName.N	RepeatList	Yes	tbtest	The name of the destination table.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the full table scan was enabled.
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SetupTable
&AllowFullTableScan=true
&DbName=dbtest
&DrdsInstanceId=drdsxxxxxxxxx
&TableName.1=tbtest
&<Common request parameters>
```

Sample success responses

XML format

```
<SetupTableResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Data>true</Data>
</SetupTableResponse>
```

JSON format

```
{
  "Data":"true",
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.53. StartRestore

Starts a backup-based recovery task for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	StartRestore	The operation that you want to perform. Set the value to StartRestore.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	sdj23xxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
PreferredBackupTime	String	No	2019-09-10 20:18:! 8	The point in time to which the PolarDB-X instance is restored. It is in the yyyy-MM-dd HH:mm:ss format.
BackupLevel	String	No	-	The backup level.
BackupMode	String	No	-	The backup mode.
BackupDbNames	String	No	test1,test2	The names of databases to be restored. Use commas (,) to separate multiple names. If you do not configure this parameter, all databases on the PolarDB-X instance are restored by default.
BackupId	String	No	23443	The ID of the PolarDB-X backup set. If this parameter is left blank, databases are restored based on PreferredBackupTime by default.

Response parameters

Parameter	Type	Example	Description
RequestId	String	721C71DD-D3D0-4327-BFDD-67832698F55B	The ID of the request.
Result	String	success	Indicates whether the recovery task was successful. If success was returned, the recovery task was successful. If an error message was returned, the recovery task failed.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=StartRestore
&DrdsInstanceId=drds43dsaew1
&<Common request parameters>
```

Sample success responses

XML format

```
<StartRestoreResponse>
    <Result>success</Result>
    <RequestId>FE104D26-AC19-49B5-AC67-947F6915E07E</RequestId>
    <Success>true</Success>
</StartRestoreResponse>
```

JSON format

```
{
    "Result":"success",
    "RequestId":"721C71DD-D3D0-4327-BFDD-67832698F55B",
    "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.54. SubmitCleanTask

Submits a cleanup task for smooth scale-out of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitCleanTask	The operation that you want to perform. Set the value to SubmitCleanTask.
DbName	String	Yes	test1	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
ExpandType	String	Yes	smooth_expand	The scale-out type. Valid values: • smooth_expand: smooth scale-out. • hot_expand: hot-spot scale-out Note Apsara Stack Enterprise Edition does not support hot-spot scale-out.
JobId	String	Yes	123	The job ID of the smooth scale-out task. The value of this parameter is the same as that of the ParentJobId parameter.

Parameter	Type	Required	Example	Description
ParentJobId	String	Yes	123	The ID of the smooth scale-out task. This parameter is returned when you send a request for the smooth scale-out task.
AccessKeyId	String	No	sejkxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitCleanTask
&DbName=test1
&DrdsInstanceId=drdsxxxxxxxxxxx
&ExpandType=smooth_expand
&JobId=123
&ParentJobId=123
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitCleanTaskResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
</SubmitCleanTaskResponse>
```

JSON format

```
{
  "RequestId":"DSSDF-SEWE-23ERW",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.55. SubmitSmoothExpandPreCheck

Submits a precheck task for smooth scale-out of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitSmoothExpandPreCheck	The operation that you want to perform. Set the value to SubmitSmoothExpandPreCheck.
DbInstType	String	Yes	RDS	The type of the database instance. Set the value to ApsaraDB RDS for MySQL.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	sdjwxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Msg	String	true	Indicates whether the precheck task was submitted.

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
TaskId	Long	2321	The ID of the precheck task.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitSmoothExpandPreCheck
&DbInstType=RDS
&DbName=test
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitSmoothExpandPreCheckResponse>
    <Msg>success</Msg>
    <RequestId>FE104D26-AC19-49B5-AC67-947F6915E07E</RequestId>
    <Success>true</Success>
    <TaskId>97358</TaskId>
</SubmitSmoothExpandPreCheckResponse>
```

JSON format

```
{
    "RequestId":"FE104D26-AC19-49B5-AC67-947F6915E07E",
    "Msg":"success",
    "TaskId":97358,
    "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.56. SubmitSmoothExpandPreCheckTask

Submits a precheck task for smooth scale-out of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitSmoothExpandPreCheckTask	The operation that you want to perform. Set the value to SubmitSmoothExpandPreCheckTask.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	kswxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Msg	String	scucess	Indicates whether the precheck task was submitted.
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
TaskId	Long	2321	The ID of the precheck task.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitSmoothExpandPreCheckTask
&DbName=test
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitSmoothExpandPreCheckTaskResponse>
    <Msg>success</Msg>
    <RequestId>FE104D26-AC19-49B5-AC67-947F6915E07E</RequestId>
    <Success>true</Success>
    <TaskId>97358</TaskId>
</SubmitSmoothExpandPreCheckTaskResponse>
```

JSON format

```
{
    "RequestId":"FE104D26-AC19-49B5-AC67-947F6915E07E",
    "Msg":"success",
    "TaskId":97358,
    "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.57. SubmitSmoothExpandTask

Submits a smooth scale-out task for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitSmoothExpandTask	The operation that you want to perform. Set the value to SubmitSmoothExpandTask.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	swdxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
DbInstanceIsCreating	Boolean	No	true	Specifies whether the target ApsaraDB RDS for MySQL instance is created in the PolarDB-X console.
RdsSuperInstances.N.AccountName	String	No	test_drds	The privileged account of the ApsaraDB RDS for MySQL instance.
RdsSuperInstances.N.Password	String	No	Test1234	The password of the privileged account of the ApsaraDB RDS for MySQL instance.
RdsSuperInstances.N.RdsName	String	No	rm-s34xxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance in privileged mode.
TransferTaskInfos.N.DbName	String	No	test_abds_0001	The name of the physical database shard to be migrated.
TransferTaskInfos.N.DstInstanceName	String	No	rm-s34xxxxxxxxx	The ID of the destination instance.
TransferTaskInfos.N.InstanceType	String	No	RDS	The type of the instance to be migrated. Set the value to ApsaraDB RDS for MySQL.
TransferTaskInfos.N.SrcInstanceName	String	No	rm-m23xxxxxxxxx	The ID of the instance to be migrated.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitSmoothExpandTask
&DbName=test
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitSmoothExpandTaskResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <success>true</success>
</SubmitSmoothExpandTaskResponse>
```

JSON format

```
{
   "RequestId":"DSSDF-SEWE-23ERW",
   "success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.58. SubmitSwitchTask

Submits a switchover task for smooth scale-out of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitSwitchTask	The operation that you want to perform. Set the value to SubmitSwitchTask.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
ExpandType	String	Yes	smooth_expand	The scale-out type. Valid values: • smooth_expand: smooth scale-out. • hot_expand: hot-spot scale-out Note Apsara Stack does not support hot-spot scale-out.
JobId	String	Yes	123	The job ID of the smooth scale-out task. The value of this parameter is the same as that of the ParentJobId parameter.
ParentJobId	String	Yes	123	The parent job ID of the smooth scale-out task. The value of this parameter is the same as that of the JobId parameter.
AccessKeyId	String	No	sejkwxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitSwitchTask
&DbName=test
&DrdsInstanceId=drdsxxxxxxxxx
&ExpandType=smooth_expand
&JobId=123
&ParentJobId=123
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitSwitchTaskResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
</SubmitSwitchTaskResponse>
```

JSON format

```
{
  "RequestId":"DSSDF-SEWE-23ERW",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.59. ValidateShardTask

Verifies whether a partitioning task of a PolarDB-X database is valid.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ValidateShardTask	The operation that you want to perform. Set the value to ValidateShardTask.
DbName	String	Yes	test	The name of the PolarDB-X database.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxx	The ID of the PolarDB-X instance.
SourceTableName	String	Yes	buyer	The name of the source table.
TargetTableName	String	Yes	buyer_new	The name of the destination table.
TaskType	String	Yes	SINGLE_TO_SHARD	The type of the task. Valid values: • SINGLE_TO_SHARD: converts a single table to a table shard. • SHARD_TO_SINGLE: converts a table shard to a single table. • SHARD_TO_SHARD: converts a table shard to another table shard.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
List	-	-	The list of returned check results.
Item	String	same_schema	The details about a check item.
Result	Integer	0	The check result. Valid values: • 0: passed. • 1: failed.

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ValidateShardTask
&DbName=test
&DrdsInstanceId=drds23dser21
&SourceTableName=buyer
&TargetTableName=buyer_new
&TaskType=SINGLE_TO_SHARD
&<Common request parameters>
```

Sample success responses

XML format

```
<ValidateShardTaskResponse>
   <List>
      <element>
         <Item>has_data</Item>
         <Result>0</Result>
      </element>
      <element>
         <item>same_schema</item>
         <result>0</result>
      </element>
      <element>
         <item>uk_in_source_table</item>
         <result>0</result>
      </element>
      <element>
         <item>uk_in_target_table</item>
         <result>0</result>
      </element>
   </List>
   <RequestId>DSSDF-SEWE-23ERW</RequestId>
   <success>true</success>
</ValidateShardTaskResponse>
```

JSON format

```
{
  "RequestId":"DSSDF-SEWE-23ERW",
  "List":[
    {
      "Result":0,
      "Item":"has_data"
    },
    {
      "result":0,
      "item":"same_schema"
    },
    {
      "result":0,
      "item":"uk_in_source_table"
    },
    {
      "result":0,
      "item":"uk_in_target_table"
    }
  ],
  "success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.5.60. RemoveDrdsDb

Deletes a database from a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveDrdsDb	The operation that you want to perform. Set the value to RemoveDrdsDb.
DbName	String	Yes	db_test	The name of the database to be deleted.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance from which the database will be deleted.
AccessKeyId	String	No	LTAI8xxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RemoveDrdsDb
&DbName=db_test
&DrdsInstanceId=drds****c6vxxyzd
&<Common request parameters>
```

Sample success responses

XML format

```
<RemoveDrdsDbResponse>
    <RequestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</RequestId>
    <Success>true</Success>
</RemoveDrdsDbResponse>
```

JSON format

```
{
  "RequestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.5.61. RefreshDrdsAtomUrl

Refreshes the connection between the PolarDB-X instance and the ApsaraDB RDS for MySQL instance whose network type is changed to ensure that the connection is smooth.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RefreshDrdsAtomUrl	The operation that you want to perform. Set the value to RefreshDrdsAtomUrl.
DbName	String	Yes	db_test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	LTAI8xxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	5D64DE5944A1E541E0CB908A	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Result	Boolean	true	Indicates whether the connection after refresh was smooth.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RefreshDrdsAtomUrl
&DbName=db_test
&DrdsInstanceId=drds****c6vxxyzd
&<Common request parameters>
```

Sample success responses

XML format

```
<RefreshDrdsAtomUrlResponse>
    <RequestId>B12FC174-D5CE-4A6E-83C1-0F8F864FF70A</RequestId>
    <Success>true</Success>
</RefreshDrdsAtomUrlResponse>
```

JSON format

```
{
  "RequestId":"B12FC174-D5CE-4A6E-83C1-0F8F864FF70A",
  "Success":"true"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6. Instance management

5.1.6.1. CreateInstanceAccount

Creates an account for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
AccountName	String	Yes	drds_sample_account	The name of the account.
Action	String	Yes	CreateInstanceAccount	The operation that you want to perform. Set the value to CreateInstanceAccount.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
Password	String	Yes	password	The password of the account.
DbPrivilege.N.DbName	String	No	testdb	The name of the PolarDB-X database.
DbPrivilege.N.Privilege	String	No	RW	The access permissions that the account has on the PolarDB-X database.

Response parameters

Parameter	Type	Example	Description
RequestId	String	drds_sample_request_id	The ID of the request.
Success	Boolean	true	Indicates whether the account was created.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=CreateInstanceAccount
&AccountName=drds_sample_account
&DrdsInstanceId=drdsxxxxxxxxxxxxxx
&Password=password
&DbPrivilege=[{"DbName":"testdb", "Privilege":"RW"}]
&<Common request parameters>
```

Sample success responses

XML format

```
<CreateInstanceAccount>
    <RequestId>FF13E47D-4E38-4A5A-BA68-4E610EF35A25</RequestId>
    <Success>true</Success>
</CreateInstanceAccount>
```

JSON format

```
{
  "RequestId":"FF13E47D-4E38-4A5A-BA68-4E610EF35A25",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.2. DescribeDrdsDBs

Queries the details about databases on a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsDBs	The operation that you want to perform. Set the value to DescribeDrdsDBs.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	DJKSJKF12831283	The AccessKey ID provided to you by Alibaba Cloud.
PageNumber	Integer	No	1	The number of the page to return. Pages start from page 1.
PageSize	Integer	No	20	The number of entries to return on each page.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data set.
Db	-	-	The details about a PolarDB-X database.

Parameter	Type	Example	Description
CreateTime	String	1563773824000	The timestamp when the PolarDB-X database was created.
DbInstType	String	RDS	The type of the PolarDB-X database.
DbName	String	drds_test	The name of the PolarDB-X database.
Mode	String	HORIZONTAL	The partition mode of the PolarDB-X database.
Schema	String	drds_test_1563773871118kxqd	The partitioning configuration ID of the PolarDB-X database.
Status	String	NORMAL	The status of the PolarDB-X database.
PageNumber	String	1	The page number of the returned page.
PageSize	String	10	The number of entries returned on the page.
RequestId	String	006B7D19-8CDB-4AA6-AAE7-23C1075276E1	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Total	String	1	The total number of databases.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsDBs
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsDBs>
    <PageSize>10</PageSize>
    <RequestId>006B7D19-8CDB-4AA6-AAE7-23C1075276E1</RequestId>
    <PageNumber>1</PageNumber>
    <Total>1</Total>
    <Data>
        <Db>
            <Status>NORMAL</Status>
            <DbInstType>RDS</DbInstType>
            <Schema>george_1563773871118kxqd</Schema>
            <CreateTime>1563773824000</CreateTime>
            <Mode>HORIZONTAL</Mode>
            <DbName>george</DbName>
        </Db>
    </Data>
    <Success>true</Success>
</DescribeDrdsDBs>
```

JSON format

```
{
  "Data":{
    "Db":[
      {
        "Status":"NORMAL",
        "DbName":"george",
        "CreateTime":1563773824000,
        "Schema":"george_1563773871118kxqd",
        "DbInstType":"RDS",
        "Mode":"HORIZONTAL"
      }
    ]
  },
  "PageNumber":1,
  "PageSize":10,
  "RequestId":"006B7D19-8CDB-4AA6-AAE7-23C1075276E1",
  "Success":true,
  "Total":1
}
```

Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.3. DescribeDrdsInstance

Queries the details about a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsInstance	The operation that you want to perform. Set the value to DescribeDrdsInstance.
DrdsInstanceId	String	Yes	drdssexxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	ASU678xxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The details about the PolarDB-X instance.
CommodityCode	String	drdsPost	The commodity code.
CreateTime	Long	1568620311000	The timestamp when the PolarDB-X instance was created.
Description	String	drds_test	The description of the PolarDB-X instance.

Parameter	Type	Example	Description
DrdsInstanceId	String	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
ExpireDate	Long	4724323200000	The timestamp when the PolarDB-X instance expires.
InstRole	String	MASTER	The role of the PolarDB-X instance. Valid values: • MASTER: the primary instance. • SLAVE: the analytic read-only instance. • SLAVE_FLOW: the concurrent read-only instance.
InstanceSeries	String	drds.sn2.4c16g	The instance type family.
InstanceSpec	String	drds.sn2.4c16g.8C32G	The specifications of the PolarDB-X instance.
Label	String	NORMAL	The label of the PolarDB-X instance. Valid values: • NORMAL: the normal instance. • HA: the high-availability instance. • VPC: the Virtual Private Cloud (VPC) instance. Note Apsara Stack does not support VPC instances.
MachineType	String	ecs	The type of the machine where the PolarDB-X instance is deployed. Set the value to ECS.

Parameter	Type	Example	Description
MasterInstanceId	String	drdsxxxxxxxxx	The ID of the primary instance. This parameter is returned only for read-only instances.
NetworkType	String	CLASSIC	The network type of the PolarDB-X instance. Valid values: • CLASSIC: the classic network. • VPC: the VPC network.
OrderInstanceId	String	drdsxxxxxxxxx	The ID of the purchased PolarDB-X instance.
ReadOnlyDBInstanceIds	-	drdsxxxxxxxxx	The list of read-only PolarDB-X instances.
ReadOnlyDBInstanceId	-	-	The details about of a read-only PolarDB-X instance.
RegionId	String	cn-hangzhou	The region of the PolarDB-X instance.
Status	String	RUN	The status of the PolarDB-X instance.
Type	String	PRIVATE	The type of the PolarDB-X instance. Valid values: • PRIVATE: the dedicated instance. • PUBLIC: the shared instance.
Version	Long	0	The version of the PolarDB-X instance. This parameter is empty.
VersionAction	String	Upgradeable	The available action for the PolarDB-X instance version.
Vips	-	-	The list of returned virtual IP addresses (VIPs).

Parameter	Type	Example	Description
Vip	-	-	The details about a VIP.
Dns	String	drdsxxxxxxxxx.drds.aliyuncs.com	The domain name bound to the VIP.
ExpireDays	Long	0	The number of days before the VIP expires.
Port	String	3306	The open port of the VIP.
Type	String	intranet	The type of the VIP. Valid values: • intranet • internet
VpcId	String	vpc-bpxxxxxxxxxxxy	The ID of the VPC network connected to the instance.
VswitchId	String	vsw-bpxxxxxxxxxxxxx96	The ID of the virtual switch (VSwitch).
VpcCloudInstanceId	String	drdsxxxxxxxxx	The ID of the PolarDB-X instance in the VPC cluster.
ZoneId	String	cn-hangzhou-e	The ID of the zone.
RequestId	String	B4F76641-BA45-4320-BE7C-9C62CFDAC9B2	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsInstance
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsInstance>
    <RequestId>B4F76641-BA45-4320-BE7C-9C62CFDAC9B2</RequestId>
    <Data>
        <Status>RUN</Status>
        <Description>drds_test</Description>
        <InstanceSpec>drds.sn2.4c16g.8C32G</InstanceSpec>
        <ZoneId>cn-hangzhou-e</ZoneId>
        <ExpireDate>4724323200000</ExpireDate>
        <CreateTime>1568620311000</CreateTime>
        <MachineType>ecs</MachineType>
        <Label>NORMAL</Label>
        <InstanceSeries>drds.sn2.4c16g</InstanceSeries>
        <VersionAction>Can_Be_Rolled_Back</VersionAction>
        <Type>PRIVATE</Type>
        <DrdsInstanceId>drdsxxxxxxxxx</DrdsInstanceId>
        <OrderInstanceId>drdsxxxxxxxxx</OrderInstanceId>
        <InstRole>MASTER</InstRole>
        <NetworkType>CLASSIC</NetworkType>
        <CommodityCode>drdsPost</CommodityCode>
        <RegionId>cn-hangzhou</RegionId>
        <Vips>
            <Vip>
                <Type>intranet</Type>
                <VpcId></VpcId>
                <Port>3306</Port>
                <Dns>drdsxxxxxxxxx.drds.aliyuncs.com</Dns>
            </Vip>
        </Vips>
    </Data>
    <Success>true</Success>
</DescribeDrdsInstance>
```

JSON format

```
{
  "Data":{
    "MachineType":"ecs",
    "Description":"drds_test",
    "OrderInstanceId":"drdsxxxxxxxxx",
    "Type":"PRIVATE",
    "ZoneId":"cn-hangzhou-e",
    "InstRole":"MASTER",
    "NetworkType":"CLASSIC",
    "Status":"RUN",
    "ExpireDate":4724323200000,
    "Vips":{
      "Vip":[
        {
          "Dns":"drdsxxxxxxxxx.drds.aliyuncs.com",
          "Port":3306,
          "Type":"intranet",
          "VpcId":""
        }
      ]
    },
    "VersionAction":"Can_Be_Rolled_Back",
    "RegionId":"cn-hangzhou",
    "CommodityCode":"drdsPost",
    "InstanceSpec":"drds.sn2.4c16g.8C32G",
    "CreateTime":1568620311000,
    "InstanceSeries":"drds.sn2.4c16g",
    "Label":"NORMAL",
    "DrdsInstanceId":"drdsxxxxxxxxx"
  },
  "RequestId":"B4F76641-BA45-4320-BE7C-9C62CFDAC9B2",
  "Success":true
}
```


Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.4. DescribeDrdsInstanceLevelTasks

Queries incomplete tasks for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsInstanceLevelTasks	The operation that you want to perform. Set the value to DescribeDrdsInstanceLevelTasks.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	ASU67xxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Tasks	-	-	The list of returned incomplete tasks.
Task	-	-	The details about an incomplete task.
AllowCancel	Boolean	false	Indicates whether the task can be canceled.

Parameter	Type	Example	Description
ErrMsg	String	The status of the instance is abnormal.	The error message of the task.
GmtCreate	Long	1568705520000	The timestamp when the task was created.
Progress	Integer	99	The task progress. Valid values: 0 to 100.
ProgressDescription	String	Executing	The description of the task execution progress.
ShowProgress	Boolean	true	Indicates whether the progress was displayed.
TaskName	String	upgrade_instance	The name of the task.
TaskPhase	String	1	The phase of the task.
TaskStatus	Integer	0	The status of the task. Valid values: • 0: executing • 1: successful • 2: failed • 3: canceled
TaskType	Integer	11	The type of the task.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsInstanceLevelTasks
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsInstanceLevelTasks>
    <Tasks>
        <Task>
            <AllowCancel>false</AllowCancel>
            <GmtCreate>1568696422000</GmtCreate>
            <Progress>90</Progress>
            <ShowProgress>true</ShowProgress>
            <TargetId>154</TargetId>
            <TaskDetail></TaskDetail>
            <ProgressDescription>Executing</ProgressDescription>
            <TaskName>Upgrade task</TaskName>
            <TaskPhase>1</TaskPhase>
            <TaskStatus>0</TaskStatus>
            <TaskType>11</TaskType>
            <ErrMsg></ErrMsg>
        </Task>
    </Tasks>
    <RequestId>1BEEBC05-437E-456D-94A4-CEE1F5676BCF</RequestId>
    <Success>true</Success>
</DescribeDrdsInstanceLevelTasks>
```

JSON format

```
{
  "RequestId":"1BEEBC05-437E-456D-94A4-CEE1F5676BCF",
  "Tasks":{
    "Task":[
      {
        "ShowProgress":true,
        "TaskPhase":1,
        "TaskDetail":"",
        "ErrMsg":"",
        "TaskStatus":0,
        "TaskType":11,
        "TaskName":"Upgrade task",
        "GmtCreate":1568696422000,
        "AllowCancel":false,
        "ProgressDescription":"Executing",
        "TargetId":154,
        "Progress":90
      }
    ]
  },
  "Success":true
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.6.5. DescribeDrdsInstanceMonitor

Queries the monitoring data of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsInstanceMonitor	The operation that you want to perform. Set the value to DescribeDrdsInstanceMonitor.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
EndTime	Long	Yes	1568705520000	The timestamp that specifies the end of the time range to query.
Key	String	Yes	cpu	The monitoring metric.
RegionId	String	Yes	cn-hangzhou	The region ID of the PolarDB-X instance.
StartTime	Long	Yes	1568701920000	The timestamp that specifies the beginning of the time range to query.
AccessKeyId	String	No	ASU6xxxxxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
PeriodMultiple	Integer	No	1	The multiple of the monitoring data period. By default, the monitoring data period is 5 minutes.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The list of returned monitoring data.
Key	String	cpu	The monitoring metric.
NodeNum	Integer	1	The number of the PolarDB-X nodes.
Unit	String	%	The unit of the monitoring metric.
Values	-	-	The obtained values of the monitoring metric.

Parameter	Type	Example	Description
Date	Long	1568703720000	The timestamp when the value was obtained.
Value	String	1.00	The metric value.
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsInstanceMonitor
&DrdsInstanceId=drdsxxxxxxxxxx
&EndTime=1568705520000
&Key=cpu
&RegionId=cn-hangzhou
&StartTime=1568701920000
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsInstanceMonitor>
    <RequestId>0EB04DD8-A3C8-4C40-A7DC-865BE490216B</RequestId>
    <Data>
        <NodeNum>1</NodeNum>
        <Values>
            <Value>1.00</Value>
            <Date>1568703720000</Date>
        </Values>
        <Values>
            <Value>0.93</Value>
            <Date>1568703780000</Date>
        </Values>
        <Values>
            <Value>0.67</Value>
            <Date>1568703840000</Date>
        </Values>
        <Unit>%</Unit>
        <Key>cpu</Key>
    </Data>
</DescribeDrdsInstanceMonitor>
```

JSON format

```
{
  "Data":[
    {
      "Values":[
        {
          "Value":"1.00",
          "Date":1568703720000
        },
        {
          "Value":"0.93",
          "Date":1568703780000
        },
        {
          "Value":"0.67",
          "Date":1568703840000
        }
      ],
      "Key":"cpu",
      "NodeNum":1,
      "Unit":"%"
    }
  ],
  "RequestId":"0EB04DD8-A3C8-4C40-A7DC-865BE490216B"
}
```


Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.6. DescribeDrdsInstances

Queries PolarDB-X instances that meet specific criteria in a region.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsInstances	The operation that you want to perform. Set the value to DescribeDrdsInstances.
RegionId	String	Yes	cn-hangzhou	The region ID of the PolarDB-X instance.
AccessKeyId	String	No	ASU678xxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
Description	String	No	drds_test	The description of the PolarDB-X instance.
Expired	Boolean	No	false	Specifies whether the instance expires.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	20	The number of entries to return on each page.
Tag.N.Key	String	No	acs:newretail:domain	The key of the Nth tag.
Tag.N.Value	String	No	NEW_RETAIL	The value of the Nth tag.
Type	String	No	1	The type of the PolarDB-X instance. Valid values: • 0: the shared instance. • 1: the dedicated instance.

Response parameters

Parameter	Type	Example	Description
Instances	-	-	The list of returned PolarDB-X instances.
Instance	-	-	The details about a PolarDB-X instance.
CommodityCode	String	drdsPost	The commodity code.
CreateTime	Long	1568620311000	The timestamp when the PolarDB-X instance was created.
Description	String	drds_test	The description of the PolarDB-X instance.
DrdsInstanceId	String	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
ExpireDate	Long	4724323200000	The timestamp when the PolarDB-X instance expires.
InstRole	String	MASTER	The role of the PolarDB-X instance. • MASTER: the primary instance. • SLAVE: the analytic read-only instance. • SLAVE_FLOW: the concurrent read-only instance.
InstanceSeries	String	drds.sn2.4c16g	The instance type family.
InstanceSpec	String	drds.sn2.4c16g.8C32G	The specifications of the PolarDB-X instance.
Label	String	NORMAL	The label of the PolarDB-X instance. Valid values: • NORMAL: the normal instance. • HA: the high-availability instance. • VPC: the Virtual Private Cloud (VPC) instance.

Parameter	Type	Example	Description
MachineType	String	ecs	The type of machine where the PolarDB-X instance is deployed. Set the value to ECS.
MasterInstanceId	String	drdsxxxxxxxxxx	The ID of the primary instance. This parameter is only available for read-only instances.
NetworkType	String	CLASSIC	The network type of the PolarDB-X instance. Valid values: • CLASSIC: the classic network. • VPC: the VPC network. Note Apsara Stack does not support VPC networks.
OrderInstanceId	String	drdsxxxxxxxxxx	The ID of the created PolarDB-X instance.
ReadOnlyDBInstanceIds	-	drdsxxxxxxxxxx	The list of read-only PolarDB-X instances.
ReadOnlyDBInstanceId	-	-	The details about of a read-only PolarDB-X instance.
RegionId	String	cn-hangzhou	The region ID of the PolarDB-X instance.
Status	String	RUN	The status of the PolarDB-X instance.
Type	String	PRIVATE	The type of the PolarDB-X instance. Valid values: • PUBLIC: the shared instance. • PRIVATE: the dedicated instance.
Version	Long	0	The version of the PolarDB-X instance.

Parameter	Type	Example	Description
VersionAction	String	Upgradable	The available actions for the PolarDB-X instance version.
Vips	-	-	The list of returned virtual IP addresses (VIPs).
Vip	-	-	The details about a VIP.
IP	String	10.xx.xx.51	The IP address.
Port	String	3306	The open port of the VIP.
Type	String	intranet	The type of the VIP. Valid values: • intranet • internet
VpcId	String	vpc-bpxxxxxxxxxxxy	The ID of the VPC network connected to the instance.
VswitchId	String	vsw-bpxxxxxxxxxxxxx96	The ID of the virtual switch (VSwitch).
VpcCloudInstanceId	String	drdsxxxxxxxxxx	The ID of the instance in the VPC cluster.
VpcId	String	vpc-bpxxxxxxxxxxxy	The ID of the VPC network connected to the instance.
ZoneId	String	vsw-bpxxxxxxxxxxxxx96	The ID of the VSwitch.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	20	The number of entries returned on the page.
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Total	Integer	1	The total number of PolarDB-X instances.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsInstances
&RegionId=cn-hangzhou
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsInstances>
    <Instances>
        <Instance>
            <Status>RUN</Status>
            <Description>drds_test</Description>
            <InstanceSpec>drds.sn2.4c16g.8C32G</InstanceSpec>
            <ZoneId>cn-hangzhou-e</ZoneId>
            <ExpireDate>4724323200000</ExpireDate>
            <CreateTime>1568620311000</CreateTime>
            <MachineType>ecs</MachineType>
            <Label>NORMAL</Label>
            <InstanceSeries>drds.sn2.4c16g</InstanceSeries>
            <VersionAction>Upgradable</VersionAction>
            <Type>PRIVATE</Type>
            <VpcId></VpcId>
            <DrdsInstanceId>drdsxxxxxxxxxx</DrdsInstanceId>
            <OrderInstanceId>drdsxxxxxxxxxx</OrderInstanceId>
            <InstRole>MASTER</InstRole>
            <NetworkType>CLASSIC</NetworkType>
            <CommodityCode>drdsPost</CommodityCode>
            <RegionId>cn-hangzhou</RegionId>
            <Vips></Vips>
        </Instance>
    </Instances>
    <PageSize>20</PageSize>
    <RequestId>8734773E-7B21-4A22-9106-CBD245F8AF7D</RequestId>
    <PageNumber>1</PageNumber>
    <Total>1</Total>
</DescribeDrdsInstances>
```

JSON format

```
{
  "PageNumber":1,
  "PageSize":20,
  "RequestId":"8734773E-7B21-4A22-9106-CBD245F8AF7D",
  "Instances":{
    "Instance":[
      {
        "MachineType":"ecs",
        "Description":"drds_test",
        "OrderInstanceId":"drdsxxxxxxxxxx",
        "Type":"PRIVATE",
        "ZoneId":"cn-hangzhou-e",
        "VpcId":"",
        "InstRole":"MASTER",
        "NetworkType":"CLASSIC",
        "Status":"RUN",
        "ExpireDate":4724323200000,
        "Vips":{
          "Vip":[]
        },
        "VersionAction":"Upgradable",
        "RegionId":"cn-hangzhou",
        "CommodityCode":"drdsPost",
        "InstanceSpec":"drds.sn2.4c16g.8C32G",
        "CreateTime":1568620311000,
        "InstanceSeries":"drds.sn2.4c16g",
        "Label":"NORMAL",
        "DrdsInstanceId":"drdsxxxxxxxxx"
      }
    ]
  },
  "Total":1
}
```


Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.7. DescribeDrdsInstanceVersion

Queries the versions of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsInstanceVersion	The operation that you want to perform. Set the value to DescribeDrdsInstanceVersion.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	ASU6xxxxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The version data of the PolarDB-X instance.
InstanceVersion	String	5.3.12-15682777	The current version of the PolarDB-X instance.
NewestVersion	String	5.3.12-15682777	The latest version of the PolarDB-X instance.
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsInstanceVersion
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsInstanceVersion>
    <RequestId>88AFEF28-DF04-4E21-9CE9-277D3C19E3CD</RequestId>
    <Data>
        <InstanceVersion>5.3.12-15682777</InstanceVersion>
        <NewestVersion>5.3.12-15682777</NewestVersion>
    </Data>
    <Success>true</Success>
</DescribeDrdsInstanceVersion>
```

JSON format

```
{
  "Data":{
    "InstanceVersion":"5.3.12-15682777",
    "NewestVersion":"5.3.12-15682777"
  },
  "RequestId":"88AFEF28-DF04-4E21-9CE9-277D3C19E3CD",
  "Success":true
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.8. DescribeDrdsParams

Queries the parameters of a PolarDB-X instance or database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsParams	The operation that you want to perform. Set the value to DescribeDrdsParams.
DrdsInstanceId	String	Yes	DescribeDrdsParams	The ID of the PolarDB-X instance.
ParamLevel	String	Yes	INSTANCE	The parameter level. Valid values: • INSTANCE: the instance level. • DB: the database level.
DbName	String	No	drds_test	The name of the PolarDB-X database.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
List	-	-	The details about the parameters that were queried.
DbName	String	drds_test	The name of the PolarDB-X database.
NeedRestart	Boolean	true	Indicates whether to restart the database.
ParamDefaultValue	String	1000	The default value of the parameter.
ParamDesc	String	The slow SQL query threshold (unit: ms)	The description of the parameter.
ParamEnglishName	String	SLOW_SQL_TIME	The English name of the parameter.
ParamLevel	String	INSTANCE	The parameter level.
ParamName	String	SLOW_SQL_TIME	The name of the parameter.
ParamRanges	String	[1000-900000]	The value range of the parameter.
ParamType	String	CONFIG	The type of the parameter.
ParamValue	String	1000	The value of the parameter.
ParamVariableName	String	slowSqlTime	The name of the parameter variable.
RequestId	String	2F7F8080-9132-4279-85D0-B7E5C4305162	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsParams
&DrdsInstanceId=DescribeDrdsParams
&ParamLevel=INSTANCE
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeDrdsParams>
    <RequestId>29D7422F-24DD-42D8-A04E-EBCE4B4A416A</RequestId>
    <List>
        <ParamLevel>INSTANCE</ParamLevel>
        <ParamRanges>[1000-900000]</ParamRanges>
        <ParamVariableName>slowSqlTime</ParamVariableName>
        <ParamEnglishName>SLOW_SQL_TIME</ParamEnglishName>
        <NeedRestart>false</NeedRestart>
        <ParamValue>1000</ParamValue>
        <ParamDefaultValue>1000</ParamDefaultValue>
        <DbName></DbName>
        <UserVisible>true</UserVisible>
        <ParamName>SLOW_SQL_TIME</ParamName>
        <ParamType>CONFIG</ParamType>
        <ParamDesc>The slow SQL query threshold (unit: ms)</ParamDesc>
    </List>
    <List>
        <ParamLevel>INSTANCE</ParamLevel>
        <ParamRanges>[3600000-86400000]</ParamRanges>
        <ParamVariableName>idleTimeout</ParamVariableName>
        <ParamEnglishName>LOGIC_IDLE_TIMEOUT</ParamEnglishName>
        <NeedRestart>false</NeedRestart>
        <ParamValue>28800000</ParamValue>
        <ParamDefaultValue>28800000</ParamDefaultValue>
        <DbName></DbName>
        <UserVisible>true</UserVisible>
        <ParamName>LOGIC_IDLE_TIMEOUT</ParamName>
        <ParamType>CONFIG</ParamType>
        <ParamDesc>The idle timeout for the connection between applications and PolarDB-X (unit: ms
)</ParamDesc>
    </List>
    <List>
        <ParamLevel>INSTANCE</ParamLevel>
        <ParamRanges>[4194304-33554432]</ParamRanges>
        <ParamVariableName>maxAllowedPacket</ParamVariableName>
        <ParamEnglishName>MAX_ALLOWED_PACKET</ParamEnglishName>
        <NeedRestart>false</NeedRestart>
        <ParamValue>16777216</ParamValue>
```

```
        <ParamDefaultValue>16777216</ParamDefaultValue>
        <DbName></DbName>
        <UserVisible>true</UserVisible>
        <ParamName>MAX_ALLOWED_PACKET</ParamName>
        <ParamType>CONFIG</ParamType>
        <ParamDesc>The maximum network packet for the interaction between applications and PolarD
B-X (unit: byte).</ParamDesc>
    </List>
    <List>
        <ParamLevel>INSTANCE</ParamLevel>
        <ParamRanges>[1073741824-6442450944]</ParamRanges>
        <ParamVariableName>GLOBAL_MEMORY_LIMIT</ParamVariableName>
        <ParamEnglishName>GLOBAL_MEMORY_LIMIT</ParamEnglishName>
        <NeedRestart>false</NeedRestart>
        <ParamValue>2147483648</ParamValue>
        <ParamDefaultValue>2147483648</ParamDefaultValue>
        <DbName></DbName>
        <UserVisible>true</UserVisible>
        <ParamName>GLOBAL_MEMORY_LIMIT</ParamName>
        <ParamType>CONFIG</ParamType>
        <ParamDesc>The maximum size of the memory pool for a read-only instance. If the memory usa
ge on an instance exceeds the value, an error occurs and the query fails.</ParamDesc>
    </List>
    <Success>true</Success>
</DescribeDrdsParams>
```

JSON format

```
{
  "RequestId":"29D7422F-24DD-42D8-A04E-EBCE4B4A416A",
  "Success":true,
  "List":[
    {
      "NeedRestart":false,
      "ParamEnglishName":"SLOW_SQL_TIME",
      "ParamDesc":"The slow SQL query threshold (unit: ms).",
      "ParamName":"SLOW_SQL_TIME",
      "DbName":"",
      "ParamValue":"1000",
      "ParamType":"CONFIG",
      "ParamLevel":"INSTANCE",
```

```
        "UserVisible":true,
        "ParamDefaultValue":"1000",
        "ParamVariableName":"slowSqlTime",
        "ParamRanges":"[1000-900000]"
      },
      {
        "NeedRestart":false,
        "ParamEnglishName":"LOGIC_IDLE_TIMEOUT",
        "ParamDesc":"The idle timeout for the connection between users and PolarDB-X (unit: ms).",
        "ParamName":"LOGIC_IDLE_TIMEOUT",
        "DbName":"",
        "ParamValue":"28800000",
        "ParamType":"CONFIG",
        "ParamLevel":"INSTANCE",
        "UserVisible":true,
        "ParamDefaultValue":"28800000",
        "ParamVariableName":"idleTimeout",
        "ParamRanges":"[3600000-86400000]"
      },
      {
        "NeedRestart":false,
        "ParamEnglishName":"MAX_ALLOWED_PACKET",
        "ParamDesc":"The maximum network packet for the interaction between users and PolarDB-X (
unit: byte).",
        "ParamName":"MAX_ALLOWED_PACKET",
        "DbName":"",
        "ParamValue":"16777216",
        "ParamType":"CONFIG",
        "ParamLevel":"INSTANCE",
        "UserVisible":true,
        "ParamDefaultValue":"16777216",
        "ParamVariableName":"maxAllowedPacket",
        "ParamRanges":"[4194304-33554432]"
      },
      {
        "NeedRestart":false,
        "ParamEnglishName":"GLOBAL_MEMORY_LIMIT",
        "ParamDesc":"The maximum size of the memory pool for a read-only instance. If the memory us
age on an instance exceeds the value, an error occurs and the query fails.",
        "ParamName":"GLOBAL_MEMORY_LIMIT",
        "DbName":"",
```

```
        "ParamValue":"2147483648",
        "ParamType":"CONFIG",
        "ParamLevel":"INSTANCE",
        "UserVisible":true,
        "ParamDefaultValue":"2147483648",
        "ParamVariableName":"GLOBAL_MEMORY_LIMIT",
        "ParamRanges":"[1073741824-6442450944]"
      }
    ]
}
```


Error codes

For a list of error codes, visit the API Error Center.

5.1.6.9. DescribeDrdsTasks

Queries the change tasks of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDrdsTasks	The operation that you want to perform. Set the value to DescribeDrdsTasks.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.
DbName	String	No	test	The name of the PolarDB-X database.
TaskType	String	No	test	The type of the task.

Response parameters

Parameter	Type	Example	Description
RequestId	String	CD412DF7-F21D-44CE-88FF-ED24917174A8	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
Tasks	-	-	The list of returned tasks.
Task	-	-	The details about a task.
Content	String	upgrade_instance	The content of the task.
Id	Long	64148	The ID of the task.
State	String	FAILED	The status of the task.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeDrdsTasks
&<Common request parameters>
```

Sample success responses

```
<DescribeDrdsTasksResponse>
    <RequestId>CD412DF7-F21D-44CE-88FF-ED24917174A8</RequestId>
    <Tasks>
        <Task>
            <State>FAILED</State>
            <Id>64148</Id>
            <Content>upgrade_instance</Content>
        </Task>

    </Tasks>
    <Success>true</Success>
</DescribeDrdsTasksResponse>
```

Error codes

HttpCode	Error code	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.10. DescribeHiStoreInstanceInfo

Queries the details about a HiStore instance of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeHiStoreInstanceInfo	The operation that you want to perform. Set the value to DescribeHiStoreInstanceInfo.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
HistoreInstanceId	String	Yes	hi-hbxxxxxxxxxxx	The ID of the HiStore instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
HiStoreInstanceInfo	-	-	The details about the HiStore instance.
RequestId	String	509BDE17-505A-4B3B-854D-30D3F092502F	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeHiStoreInstanceInfo
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeHiStoreInstanceInfoResponse>
    <RequestId>509BDE17-505A-4B3B-854D-30D3F092502F</RequestId>
      <HiStoreInstanceInfo>
          <DiskSize>100</DiskSize>
          <GmtCreate>1568267711</GmtCreate>
          <HistoreInstanceId>test</HistoreInstanceId>
          <MachineSpec>4C8G</MachineSpec>
          <RpmVersion>1.0.15</RpmVersion>
      </HiStoreInstanceInfo>

    <Success>true</Success>
</DescribeHiStoreInstanceInfoResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.11. DescribeInstanceAccounts

Queries the account information of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeInstanceAccounts	The operation that you want to perform. Set the value to DescribeInstanceAccounts.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	ABCxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
InstanceAccounts	-	-	The returned list of instance accounts.
InstanceAccount	-	-	The details about an instance account.
AccountName	String	drdsuser	The name of the account.
AccountType	Integer	1	The type of the account.
DbPrivileges	-	-	The database permissions that the account has.
DbPrivilege	-	-	The details about the database permission.
DbName	String	drdsdb	The name of the database.
Privilege	String	root	The permission.
Description	String	Permission	The description of the permission.
Host	String	%	The IP address of the instance.
RequestId	String	2D77B55F-50C4-4FC5-A89A-7746B7D5781D	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeInstanceAccounts
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeInstanceAccountsResponse>
    <InstanceAccounts>
        <InstanceAccount>
            <Description>Created by DRDS</Description>
            <DbPrivileges>
                <DbPrivilege>
                    <DbName>drdsdb</DbName>
                    <Privilege>ROOT</Privilege>
                </DbPrivilege>
            </DbPrivileges>
            <Host>%</Host>
            <AccountName>drdsuser</AccountName>
            <AccountType>0</AccountType>
        </InstanceAccount>

    </InstanceAccounts>
    <RequestId>2D77B55F-50C4-4FC5-A89A-7746B7D5781D</RequestId>
    <Success>true</Success>
</DescribeInstanceAccountsResponse>
```

Error codes

HttpCode	Error codes	Error message	Description
400	ActionUnauthorized	The specified action is not available for you	The error message returned because you are not authorized to perform this action. Perform RAM authorization first.

For a list of error codes, visit the API Error Center.

5.1.6.12. DescribeInstanceMenuSwitch

Queries the switch configuration of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeInstanceMenuSwitch	The operation that you want to perform. Set the value to DescribeInstanceMenuSwitch.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeInstanceMenuSwitch
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeInstanceMenuSwitchResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
    <Config>Config</Config>
</DescribeInstanceMenuSwitchResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.13. DescribeInstDbSlsInfo

Queries the names of the Log Service project and Logstore used by SQL audit.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeInstDbSlsInfo	The operation that you want to perform. Set the value to DescribeInstDbSlsInfo.
DbName	String	Yes	drdsdb	The name of the PolarDB-X database.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	test	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
AuditInfo	-	-	The SQL audit details.
LogStore	String	test1	The Logstore used by SQL audit.
Project	String	test2	The Log Service project used by SQL audit.
RequestId	String	DC3ABA3E-0F8A-4596-9104-F5155C34315B	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeInstDbSlsInfo
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeInstDbSlsInfoResponse>
    <AuditInfo>
        <LogStore>test</LogStore>
        <Project>test2</Project>
    </AuditInfo>
    <RequestId>DC3ABA3E-0F8A-4596-9104-F5155C34315B</RequestId>
    <Success>true</Success>
</DescribeInstDbSlsInfoResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.14. DescribePreCheckResult

Queries precheck results of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribePreCheckResult	The operation that you want to perform. Set the value to DescribePreCheckResult.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-Xinstance.
TaskId	String	Yes	4561	The ID of the precheck task.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
PreCheckResult	-	-	The precheck result.
PreCheckName	String	Does a unique key exist	The name of the precheck task.
State	String	FAIL	The status of the precheck task.
SubCheckItems	-	-	The details about a precheck item.
ErrorMsgCode	String	1004	The error code.
ErrorMsgParams	-	table1	The error parameter.
PreCheckItemName	String	Unique key	The name of the precheck item.
State	String	SUCCESS	The status of the precheck item.

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribePreCheckResult
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribePreCheckResultResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
    <PreCheckResult>
     <PreCheckName>Whether a unique key exists</PreCheckName>
     <State>FAIL</State>
     <SubCheckItems>
         <ErrorMsgCode>1004</ErrorMsgCode>
         <ErrorMsgParams>table1</ErrorMsgParams>
         <PreCheckItemName>Shard key</PreCheckItemName>
         <State>SUCCESS</State>
     </SubCheckItems>
    </PreCheckResult>
    <PreCheckResult></PreCheckResult>
</DescribePreCheckResultResponse>
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.15. DescribeStorageInstanceSubDbInfo

Queries the details about a storage instance of a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeStorageInstanceSubDbInfo	The operation that you want to perform. Set the value to DescribeStorageInstanceSubDbInfo.
DbName	String	Yes	drds_flashback	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxx	The ID of the PolarDB-X instance.
InstanceId	String	Yes	rm-bp1xxxxxxxxxxxxxx	The ID of the storage instance, namely, ApsaraDB RDS for MySQL instance.
AccessKeyId	String	No	******	The AccessKey ID provided to you by Alibaba Cloud.
InstType	String	No	null	The type of the instance. This parameter is currently unavailable.

Response parameters

Parameter	Type	Example	Description
Data	-	-	The returned data object.
Capacity	Long	194560	The capacity of the ApsaraDB RDS for MySQL instance.
RdsInstanceId	String	rm-bp1xxxxxxxxxxxxxx	The ID of the ApsaraDB RDS for MySQL instance.
SubDbList	-	-	The returned list of database shards.

Parameter	Type	Example	Description
SubDb	-	-	The details about a database shard.
SubDbName	String	drds_flashback_b8cd_0005	The name of the database shard.
Use	Long	0	Indicates whether the database shard was in use.
WarningLevel	Integer	0	The alert level.
TipsType	Integer	0	The type of the tip.
Use	Long	2293	The usage of the database shard.
WarningLevel	Integer	0	The alert level.
RequestId	String	78BA6289-5E03-4A31-8CB8-FC298DB63184	The unique ID of the request. If any problem occurs, provide this request ID for troubleshooting.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=DescribeStorageInstanceSubDbInfo
&<Common request parameters>
```

Sample success responses

XML format

```
<DescribeStorageInstanceSubDbInfoResponse>
    <code>200</code>
    <data>
        <RequestId>78BA6289-5E03-4A31-8CB8-FC298DB63184</RequestId>
        <Data>
            <SubDbList>
                <SubDb>
                    <WarningLevel>0</WarningLevel>
                    <Use>0</Use>
```

```
        <SubDbName>drds_flashback_b8cd_0005</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0004</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0003</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0002</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0007</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0006</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0001</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>0</WarningLevel>
        <Use>0</Use>
        <SubDbName>drds_flashback_b8cd_0000</SubDbName>
    </SubDb>
    <SubDb>
        <WarningLevel>3</WarningLevel>
        <Use>2293</Use>
        <SubDbName>other data</SubDbName>
```

```
                <SubDbName> other data </SubDbName>
            </SubDb>
        </SubDbList>
        <WarningLevel>0</WarningLevel>
        <Capacity>194560</Capacity>
        <Use>2293</Use>
        <TipsType>0</TipsType>
        <RdsInstanceId>rm-bp1xxxxxxxxxxxxxxx</RdsInstanceId>
    </Data>
    <Success>true</Success>
  </data>
  <requestId>78BA6289-5E03-4A31-8CB8-FC298DB63184</requestId>
  <successResponse>true</successResponse>

</DescribeStorageInstanceSubDbInfoResponse>
```

JSON format

```
{
  "successResponse":true,
  "requestId":"78BA6289-5E03-4A31-8CB8-FC298DB63184",
  "data":{
    "Data":{
      "WarningLevel":0,
      "TipsType":0,
      "RdsInstanceId":"rm-bp1xxxxxxxxxxxxxx",
      "SubDbList":{
        "SubDb":[
          {
            "WarningLevel":0,
            "SubDbName":"drds_flashback_b8cd_0005",
            "Use":0
          },
          {
            "WarningLevel":0,
            "SubDbName":"drds_flashback_b8cd_0004",
            "Use":0
          },
          {
            "WarningLevel":0,
            "SubDbName":"drds_flashback_b8cd_0003",
            "Use":0
```

```
            },
            {
              "WarningLevel":0,
              "SubDbName":"drds_flashback_b8cd_0002",
              "Use":0
            },
            {
              "WarningLevel":0,
              "SubDbName":"drds_flashback_b8cd_0007",
              "Use":0
            },
            {
              "WarningLevel":0,
              "SubDbName":"drds_flashback_b8cd_0006",
              "Use":0
            },
            {
              "WarningLevel":0,
              "SubDbName":"drds_flashback_b8cd_0001",
              "Use":0
            },
            {
              "WarningLevel":0,
              "SubDbName":"drds_flashback_b8cd_0000",
              "Use":0
            },
            {
              "WarningLevel":3,
              "SubDbName":"other data",
              "Use":2293
            }
          ]
        },
        "Capacity":194560,
        "Use":2293
      },
      "RequestId":"78BA6289-5E03-4A31-8CB8-FC298DB63184",
      "Success":true
    },
    "code":"200"
```

```
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.16. EnableSqlAudit

Enables SQL audit for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	EnableSqlAudit	The operation that you want to perform. Set the value to EnableSqlAudit.
DbName	String	Yes	tpch_50g_4c32gx4	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	2zAEvxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
IsRecall	Boolean	No	true	Specifies whether to backtrack historical SQL statements for auditing.
RecallEndTimestamp	String	No	1568875132000	The end time of the backtracking.
RecallStartTimestamp	String	No	1568875132000	The start time of the backtracking.

Response parameters

Parameter	Type	Example	Description
RequestId	String	463A5F0F-12AD-4544-A902-B2B983CD2C11	The ID of the request.

Parameter	Type	Example	Description
Result	Boolean	true	Indicates whether SQL audit was enabled for the database. Valid values: • true: enabled. • false: failed.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=EnableSqlAudit
&DbName=tpch_50g_4c32gx4
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<EnableSqlAuditResponse>
    <Result>true</Result>
    <RequestId>463A5F0F-12AD-4544-A902-B2B983CD2C11</RequestId>
    <Success>true</Success>
</EnableSqlAuditResponse>
```

JSON format

```
{
  "Result":true,
  "RequestId":"463A5F0F-12AD-4544-A902-B2B983CD2C11",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.17. EnableSqlFlashbackMatchSwitch

Enables SQL flashback matching. After it is enabled, the query information is added to the ApsaraDB RDS for MySQL binary logs that are used by the PolarDB-X database to precisely restore data.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	EnableSqlFlashbackMatchSwitch	The operation that you want to perform. Set the value to EnableSqlFlashbackMatchSwitch.
DbName	String	Yes	testdb	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	2zAEvsxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	463A5F0F-12AD-4544-A902-B2B983CD2C11	The ID of the request.
Result	Boolean	true	Indicates whether SQL flashback matching was enabled for the database. Valid values: • true: enabled. • false: failed.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=EnableSqlFlashbackMatchSwitch
&DbName=testdb
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<EnableSqlFlashbackMatchSwitchResponse>
    <Result>true</Result>
    <RequestId>463A5F0F-12AD-4544-A902-B2B983CD2C11</RequestId>
    <Success>true</Success>
</EnableSqlFlashbackMatchSwitchResponse>
```

JSON format

```
{
  "Result":true,
  "RequestId":"463A5F0F-12AD-4544-A902-B2B983CD2C11",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.18. FlashbackRecycleBinTable

Restores a logical table in the recycle bin.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	FlashbackRecycleBinTable	The operation that you want to perform. Set the value to FlashbackRecycleBinTable.
DbName	String	Yes	testdb	The name of the database where the target table is located.

Parameter	Type	Required	Example	Description
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxxxx	The ID of the PolarDB-X instance where the target table is located.
TableName	String	Yes	test	The name of the logical table to be restored.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the restoration was enabled. Valid values: • true: enabled. • false: failed.
RequestId	String	463A5F0F-12AD-4544-A902-B2B983CD2C11	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=FlashbackRecycleBinTable
&DbName=testdb
&DrdsInstanceId=drdsxxxxxxxxxxxxxx
&TableName=test
&<Common request parameters>
```

Sample success responses

XML format

```
<FlashbackRecycleBinTableResponse>
    <Result>true</Result>
    <RequestId>463A5F0F-12AD-4544-A902-B2B983CD2C11</RequestId>
    <Success>true</Success>
</FlashbackRecycleBinTableResponse>
```

JSON format

```
{
  "Result":true,
  "RequestId":"463A5F0F-12AD-4544-A902-B2B983CD2C11",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.19. ModifyAccountDescription

Modifies the account description of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
AccountName	String	Yes	testuser	The name of the account.
Action	String	Yes	ModifyAccountDescription	The operation that you want to perform. Set the value to ModifyAccountDescription.
Description	String	Yes	Created by PolarDB-X	The description of the account.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	2zAEvxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	2F93CCD5-806F-4470-BBC7-20476AC48DC6	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyAccountDescription
&AccountName=testuser
&Description=Created by PolarDB-X
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifyAccountDescriptionResponse>
    <RequestId>2F93CCD5-806F-4470-BBC7-20476AC48DC6</RequestId>
    <Success>true</Success>
</ModifyAccountDescriptionResponse>
```

JSON format

```
{
  "RequestId":"2F93CCD5-806F-4470-BBC7-20476AC48DC6",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.20. ModifyDrdsInstanceDescription

Modifies the description of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ModifyDrdsInstanceDescription	The operation that you want to perform. Set the value to ModifyDrdsInstanceDescription.
Description	String	Yes	Long occupied	The description of the PolarDB-X instance.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	2zAEvsxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	C44CA24C-C7C4-4C0F-8AC9-1343F279B09C	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=ModifyDrdsInstanceDescription
&Description=Long occupied
&DrdsInstanceId=drdsxxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<ModifyDrdsInstanceDescriptionResponse>
    <RequestId>C44CA24C-C7C4-4C0F-8AC9-1343F279B09C</RequestId>
    <Success>true</Success>
</ModifyDrdsInstanceDescriptionResponse>
```

JSON format

```
{
  "RequestId":"C44CA24C-C7C4-4C0F-8AC9-1343F279B09C",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.21. RemoveDrdsInstance

Releases a PolarDB-X instance.

> Note You cannot release a PolarDB-X instance that is changing configuration or has associated databases.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveDrdsInstance	The operation that you want to perform. Set the value to RemoveDrdsInstance.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	f32safdxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RemoveDrdsInstance
&DrdsInstanceId=drdsxxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<RemoveDrdsInstanceResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
</RemoveDrdsInstanceResponse>
```

JSON format

```
{
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.22. RemoveInstanceAccount

Deletes the instance management account of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
AccountName	String	Yes	testaccount	The name of the account.
Action	String	Yes	RemoveInstanceAccount	The operation that you want to perform. Set the value to RemoveInstanceAccount.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
AccessKeyId	String	No	f3d13xxxxxxxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Success	Boolean	true	Indicates whether the instance management account was deleted.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RemoveInstanceAccount
&AccountName=testaccount
&DrdsInstanceId=drdsxxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<RemoveInstanceAccountResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
</RemoveInstanceAccountResponse>
```

JSON format

```
{
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.23. RestartDrdsInstance

Restarts a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RestartDrdsInstance	The operation that you want to perform. Set the value to RestartDrdsInstance.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	fds14xxxxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	6F70CE62-5077-4B7B-95BC-4DAC45614DBE	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.
TaskId	Long	1	The ID of the task.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RestartDrdsInstance
&DrdsInstanceId=drdsxxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<RestartDrdsInstanceResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <TaskId>1</TaskId>
</RestartDrdsInstanceResponse>
```

JSON format

```
{
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "TaskId":1,
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.24. RollbackInstanceVersion

Rolls back the version of a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RollbackInstanceVersion	The operation that you want to perform. Set the value to RollbackInstanceVersion.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	String	true	Indicates whether the request was successful.
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=RollbackInstanceVersion
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<RollbackInstanceVersionResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Data>True</Data>
</RollbackInstanceVersionResponse>
```

JSON format

```
{
  "Data":"true",
  "RequestId":"DSSDF-SEWE-23ERW"
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.25. SetBackupPolicy

Configures a backup policy for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetBackupPolicy	The operation that you want to perform. Set the value to SetBackupPolicy.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
AccessKeyId	String	No	fafdsxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.

Parameter	Type	Required	Example	Description
BackupDbNames	String	No	db_test1,db_test2	The name of the database involved in the backup.
BackupLevel	String	No	db	The level of the backup. Valid values: • db: the database level. • instance: the instance level.
BackupLog	String	No	1	Specifies whether the backup log function is enabled. Valid values: • 1: enabled. • 0: disabled.
BackupMode	String	No	phy	The mode of the backup. Valid values: • logic: the logical backup. • phy: the physical backup.
DataBackupRetentionPeriod	String	No	1	The retention period of the data backup. Unit: days.
LogBackupRetentionPeriod	String	No	1	The retention period of the data backup. Unit: days.
PreferredBackupEndTime	String	No	12:30:30	The end time of the backup.

Parameter	Type	Required	Example	Description
PreferredBackupPeriod	String	No	0	The backup period. Valid values: • 0: Monday. • 1: Tuesday. • 2: Wednesday. • 3: Thursday. • 4: Friday. • 5: Saturday. • 6: Sunday.
PreferredBackupStartTime	String	No	11:30:30	The start time of the backup.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SetBackupPolicy
&DrdsInstanceId=drdsxxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<SetBackupPolicyResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Result>success</Result>
</SetBackupPolicyResponse>
```

JSON format

```
{
  "Result":"success",
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.26. SetupDrdsParams

Configures parameters for a PolarDB-X instance or database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetupDrdsParams	The operation that you want to perform. Set the value to SetupDrdsParams.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxxx	The ID of the PolarDB-X instance.
ParamLevel	String	Yes	DB	The parameter level. Valid values: • INSTANCE: the instance level. • DB: the database level.
Data.N.DbName	String	No	test_db	The name of the database corresponding to the parameter. The database name is valid only for database-level parameters.
Data.N.ParamRanges	String	No	[true|false]	The valid values of the parameter.

Parameter	Type	Required	Example	Description
Data.N.ParamType	String	No	ATOM	The type of the parameter. Valid values: • ATOM: the configuration item in the layer-3 data source. • CONFIG: the configuration item in ConfigServer. • DIAMOND: the configuration item in Diamond.
Data.N.ParamValue	String	No	true	The value of the parameter.
Data.N.ParamVariableName	String	No	FORBID_EXECUTE_DML_ALL	The name of the variable, which is specific to the program.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SetupDrdsParams
&DrdsInstanceId=drdsxxxxxxxxxxx
&ParamLevel=DB
&<Common request parameters>
```

Sample success responses

XML format

```
<SetupDrdsParamsResponse>
    <RequestId>A3140FC7-B78B-4D8E-B0C8-926D28669934</RequestId>
    <Success>true</Success>
    <Data>true</Data>
</SetupDrdsParamsResponse>
```

JSON format

```
{
  "Data":true,
  "RequestId":"A3140FC7-B78B-4D8E-B0C8-926D28669934",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.27. SubmitRollbackTask

Submits a smooth scale-out rollback task for a PolarDB-X database.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitRollbackTask	The operation that you want to perform. Set the value to SubmitRollbackTask.
DbName	String	Yes	test	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.

Parameter	Type	Required	Example	Description
ExpandType	String	Yes	smooth_expand	The scale-out type. Valid values: • smooth_expand: smooth scale-out. • hot_expand: hot-spot scale-out Note The hot-spot scale-out is not supported.
JobId	String	Yes	123	The job ID of the smooth scale-out task. The value of this parameter is the same as that of the ParentJobId parameter.
ParentJobId	String	Yes	123	The parent job ID of the smooth scale-out task. The value of this parameter is the same as that of the JobId parameter.
AccessKeyId	String	No	sejkwe21234sd	The AccessKey ID provided to you by Alibaba Cloud.

Response parameters

Parameter	Type	Example	Description
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitRollbackTask
&DbName=test
&DrdsInstanceId=drdsxxxxxxxxxx
&ExpandType=smooth_expand
&JobId=123
&ParentJobId=123
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitRollbackTaskResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
</SubmitRollbackTaskResponse>
```

JSON format

```
{
  "RequestId":"DSSDF-SEWE-23ERW",
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.28. SubmitSqlFlashbackTask

Submits an SQL flashback task for a PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitSqlFlashbackTask	The operation that you want to perform. Set the value to SubmitSqlFlashbackTask.

Parameter	Type	Required	Example	Description
DbName	String	Yes	testdb	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-X instance.
EndTime	String	Yes	2019-09-10 23:23:23	The time when the SQL flashback task ends.
RecallRestoreType	Integer	Yes	0	Specifies whether to implement exact match or fuzzy match. Valid values: • 0: the exact match. • 1: the fuzzy match.
StartTime	String	Yes	2019-09-10 20:23:23	The time when the SQL flashback task starts.
AccessKeyId	String	No	Accexxxxxxxxxxx	The AccessKey ID provided to you by Alibaba Cloud.
RecallType	Integer	No	0	Specifies whether to implement exact match or fuzzy match. Valid values: • 0: the exact match. • 1: the fuzzy match.
SqlPk	String	No	23423	The primary key of the flashback SQL statement.

Parameter	Type	Required	Example	Description
SqlType	String	No	INSERT,UPDATE	The type of the SQL statement. Valid values: INSERT, UPDATE, and DELETE. Separate multiple values with comma (,).
TableName	String	No	tb1	The name of the table on which you want to perform SQL flashback.
TraceId	String	No	asenw3sa	The trace ID of the flashback SQL statement. You can query it in the PolarDB-X SQL audit results.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SubmitSqlFlashbackTask
&DbName=testdb
&DrdsInstanceId=drdsxxxxxxxxxx
&EndTime=2019-09-10 23:23:23
&RecallRestoreType=0
&StartTime=2019-09-10 20:23:23
&<Common request parameters>
```

Sample success responses

XML format

```
<SubmitSqlFlashbackTaskResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>True</Success>
    <TaskId>2323</TaskId>
</SubmitSqlFlashbackTaskResponse>
```

JSON format

```
{
  "RequestId":"DSSDF-SEWE-23ERW",
  "TaskId":2323,
  "Success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.29. SwitchGlobalBroadcastType

Switches the broadcast table in a PolarDB-X database from the multi-write mode to the asynchronous link mode.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SwitchGlobalBroadcastType	The operation that you want to perform. Set the value to SwitchGlobalBroadcastType.
DbName	String	Yes	testdb	The name of the PolarDB-X database.
DrdsInstanceId	String	Yes	drdsxxxxxxxxx	The ID of the PolarDB-X instance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the broadcast table was switched from multi-write mode to asynchronous link mode.
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request was successful.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=SwitchGlobalBroadcastType
&DbName=testdb
&DrdsInstanceId=drdsxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<ValidateShardTaskResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Success>true</Success>
    <Data>true</Data>
</ValidateShardTaskResponse>
```

JSON format

```
{
   "Data":true,
   "RequestId":"DSSDF-SEWE-23ERW",
   "success":true
}
```

Error codes

For a list of error codes, visit the API Error Center.

5.1.6.30. UpgradeInstanceVersion

Upgrades the version of the PolarDB-X instance.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	UpgradeInstanceVersion	The operation that you want to perform. Set the value to UpgradeInstanceVersion.
DrdsInstanceId	String	Yes	drdsxxxxxxxxxx	The ID of the PolarDB-Xinstance.
RegionId	String	No	cn-hangzhou	The region ID of the PolarDB-X instance.

Response parameters

Parameter	Type	Example	Description
Data	String	true	Indicates whether the request was successful.
RequestId	String	DSSDF-SEWE-23ERW	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/? Action=UpgradeInstanceVersion
&DrdsInstanceId=drdsxxxxxxxxxx
&<Common request parameters>
```

Sample success responses

XML format

```
<UpgradeInstanceVersionResponse>
    <RequestId>DSSDF-SEWE-23ERW</RequestId>
    <Data>True</Data>
</UpgradeInstanceVersionResponse>
```

JSON format

```
{
  "Data":"true",
  "RequestId":"DSSDF-SEWE-23ERW"
}
```

Error codes

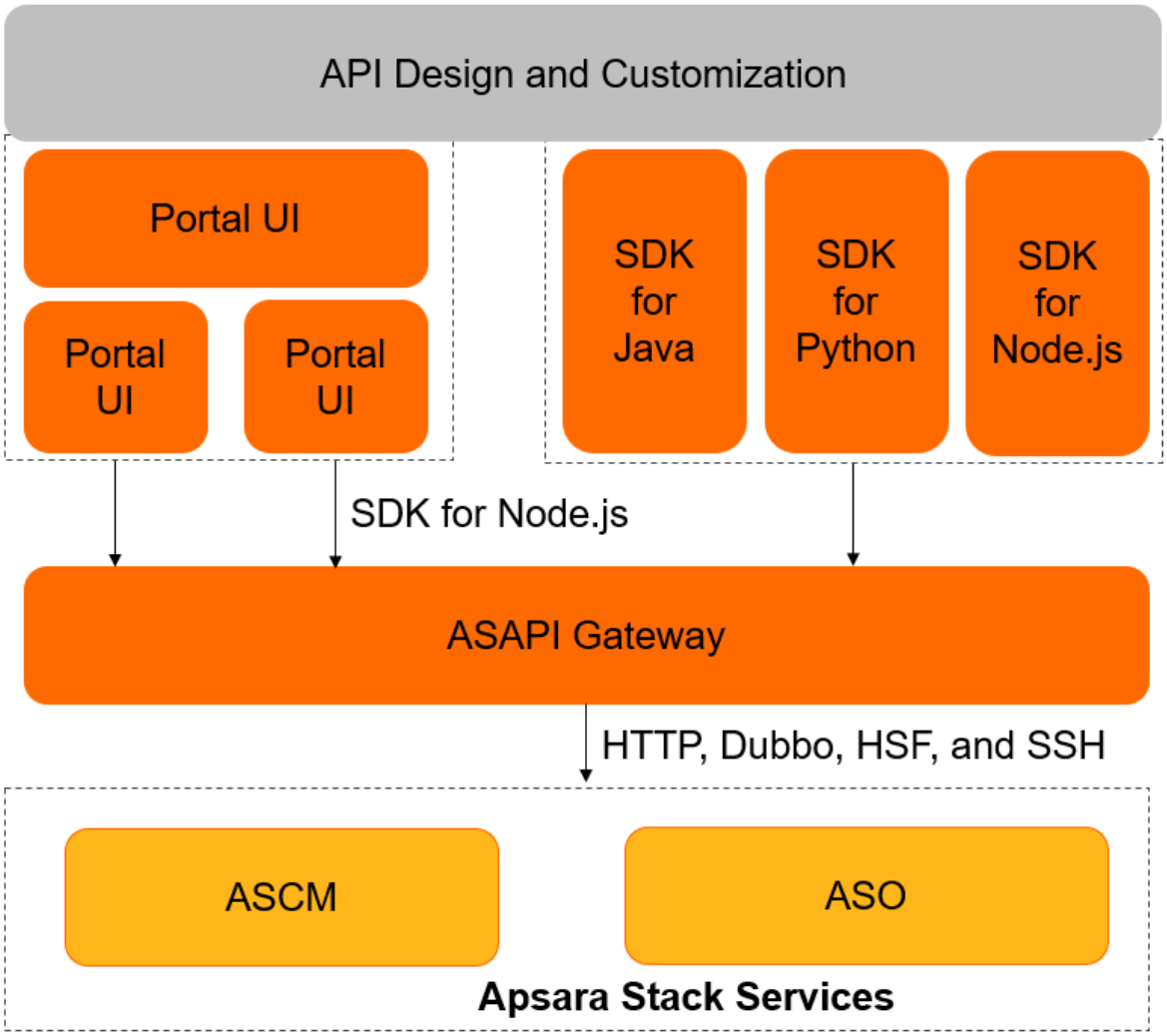
For a list of error codes, visit the API Error Center.

5.2. Call SDKs for cloud services by using ASAPI

5.2.1. Overview

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

ASAPI is a platform that integrates all features and is the only portal to connect to Apsara Stack. ASAPI can categorize operations and audit and manage their calls. It can also assemble and customize existing operations and adapt to your operations. Each cloud service can call ASAPI for required data and features, avoiding inconsistent operations caused by different call methods and separate encapsulation. This can ensure the operation program integrity and improve development efficiency and code portability.



ASAPI allows you to manage and call operations in a unified manner and customize development. ASAPI supports access of most cloud services deployed in Apsara Stack and passthrough or encapsulation of different backend APIs. This way, operations can be called with the same method. ASAPI allows you to use SDKs and API Portal to call APIs registered in ASAPI, customize SDKs, and manage versions of Apsara Stack, services, and APIs.

5.2.2. Log on to the API Tool console

You can log on to the API Tool console through the Apsara Stack Cloud Management (ASCM) console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.

2. Enter your username and password.

   Obtain the username and password used to log on to the console from the operations administrator.

   > **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum 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. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%).

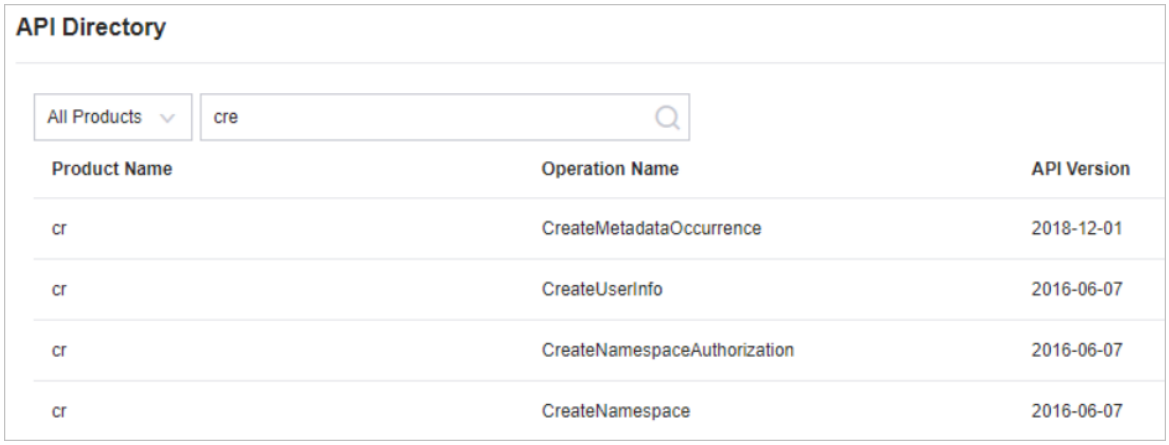
3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products** > **Others** > **API Tool**.

5.2.3. View the information about API operations

You must obtain a service name, API operation name, and API version. All the information is required when you use SDKs.

Procedure

1. Log on to the API Tool console.
2. In the left-side navigation pane, click **API fold**.
3. On the **API Directory** page, select the target service from the drop-down list, enter the API operation name in the search box, and then click the search icon.



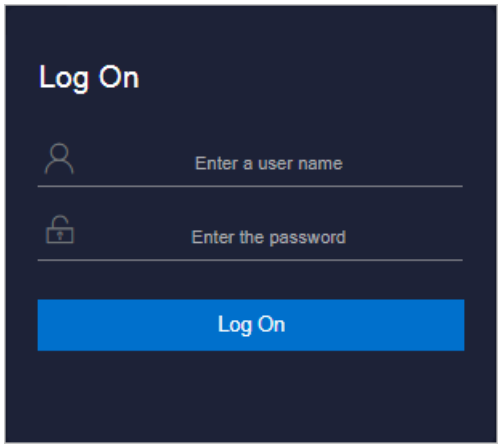
4. View and remember the service name, API operation name, and API version.

5.2.4. Obtain the endpoint of ASAPI

You can obtain the endpoint of Apsara Stack API (ASAPI) in the Apsara Stack Operations (ASO) 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.

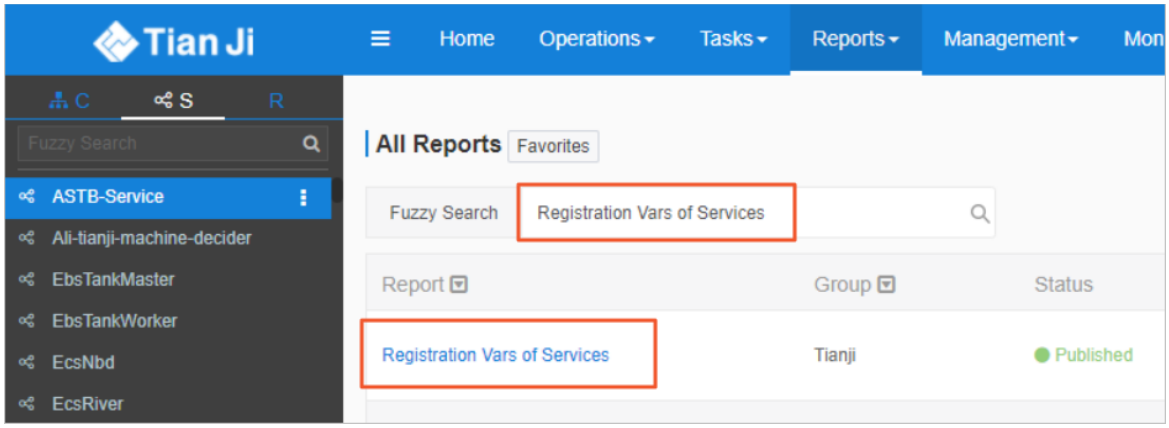
> **Note** Obtain the username and password for logging on to the ASO console from the deployment personnel or an administrator.

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

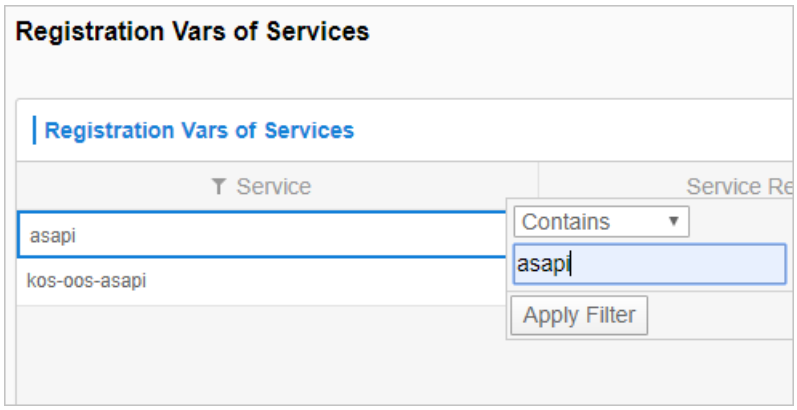
To enhance security, a password must meet the following requirements:

- 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.

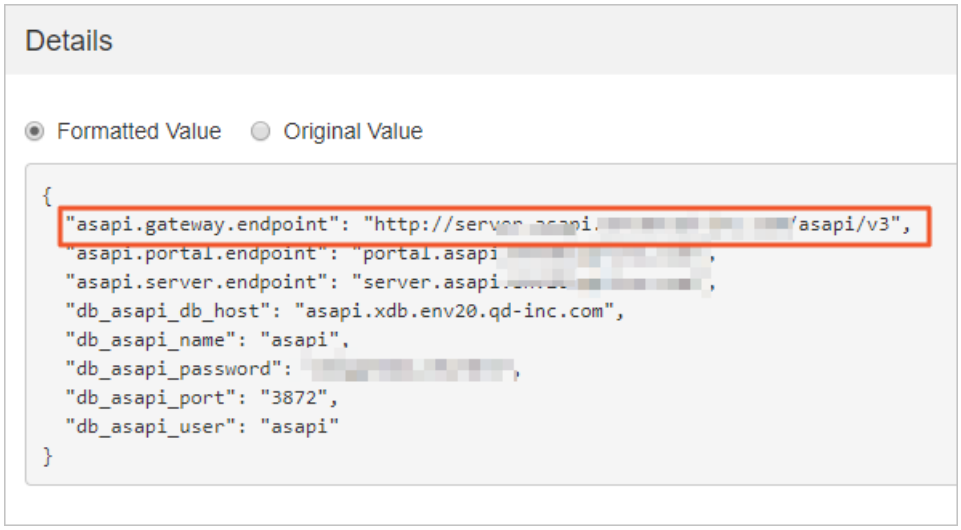
3. Click **Log On** to go to the **ASO** console.
4. In the left-side navigation pane, choose **Products** > **Product List**. On the Product List page, click **Apsara Infrastructure Management Framework**.
5. In the left-side navigation pane of the Apsara Infrastructure Management Framework console, click **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 corresponding to asapi and select **Show More**.
9. In the **Details** dialog box, view the value of **asapi.gateway.endpoint**. The value is the logon URL of the Apsara ASAPI Management platform.



5.2.5. Obtain the SDK package for Asapi

Apsara Stack uses Asapi as the unified portal for all Apsara Stack products that are connected to Asapi. Asapi also provides SDKs to call product APIs.

To obtain the SDK package for Asapi, see Apsara Stack documentation.

5.2.6. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only operations administrators and 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:

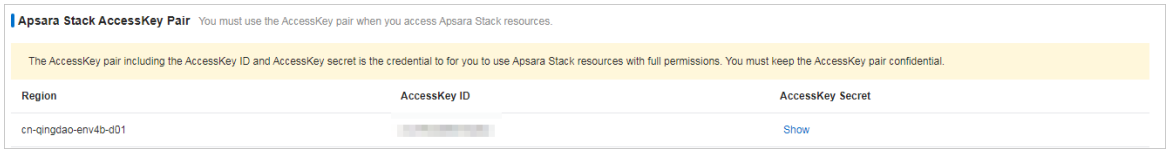
1. Log on to the Apsara Stack Cloud Management (ASCM) console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, click the icon on the right of the organization to be added.
5. Select **AccessKey**.
6. In the message that appears, view the AccessKey pair of the organization.

> **Note** An AccessKey pair is automatically allocated to a level-1 organization. Subordinate organizations use the same AccessKey pair as their level-1 organization.

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 ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section of the User Information page, view your AccessKey pair.



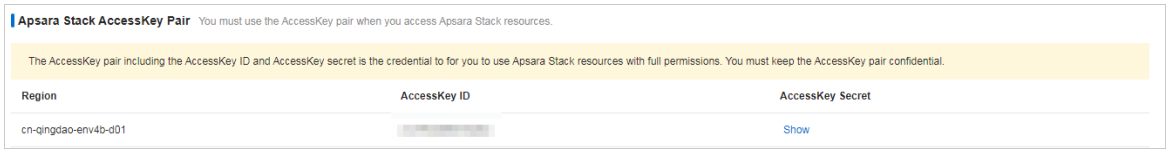
> **Note** The AccessKey pair consists of an AccessKey ID and an AccessKey secret. They allow you to access Apsara Stack resources with full permissions for your account. You must keep them confidential.

5.2.7. Query the region ID

You must specify the region ID when you call Apsara Stack SDK for Node.js. You must query the region ID in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section, you can view the region ID.



5.2.8. Call PolarDB-X SDK for Java

This topic describes how to obtain and use Cloud Native Distributed Database PolarDB-X (PolarDB-X) SDK for Java.

Prerequisites

- You must have an Apsara Stack tenant account and an AccessKey pair (AccessKey ID and AccessKey secret) to use PolarDB-X SDK for Java. For more information about how to obtain and view your AccessKey pair in the Apsara Stack Cloud Management (ASCM) console, see Obtain an AccessKey pair.
- You have obtained the endpoint of Apsara Stack API (ASAPI). For more information, see Obtain the endpoint of Asapi.
- The version of PolarDB-X SDK for Java is 1.6 or later.

Add the dependency

Add the following dependency to pom.xml:

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>asapi</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/aliyun-java-sdk-asapi.jar</systemPath>
</dependency>
```

Use SDK for Java

You can call ASAPI API operations by sending GET and POST requests.

- GET request

```
ASClient client = new ASClient();

// Form 1
String result = client.doGet(endpoint, parameters);

// Form 2
String result = client.doRequest(endpoint, parameters);
```

- POST request

```
// Set the endpoint of ASAPI.
String endpoint = "http://xxx/asapi/v3";

// Set request parameters.
Map<String, String> parameters = new HashMap<String, String>();

// Set product parameters.
parameters.put("Product", "ascm");          // The product name, which must be the same as that disp
layed on the APIs page.
parameters.put("Action", "AddUser");        // The API operation, which is case-sensitive and must b
e the same as that on the APIs page.
parameters.put("Version", "2019-05-10");    // The API version.

// Set authorization parameters.
parameters.put("AccessKeyId", "<your-access-key-id>");        // The AccessKey ID of the authorized
account.
parameters.put("AccessKeySecret", "<your-access-key-secret>");  // The AccessKey secret of the a
uthorized account.

// Set service parameters.
parameters.put("username", "<your-username>");              // The username used to log on to the
ASCM console
parameters.put("password", "<your-password>");              // The password used to log on to the A
SCM console
JSONObject jsonObject = new JSONObject();
jsonObject.put("param1", "json parameter 1");
jsonObject.put("param2", "json parameter 2");
parameters.put("json", jsonObject.toJSONString());

// Create an ASClient instance for initiating requests.
ASClient client = new ASClient();
// Add the request header.
client.addHeader("x-acs-resourceGroupId", "112233");
// Initiate the request and obtain the response.
String result = client.doPost(endpoint, parameters);
```

5.2.9. Call PolarDB-X SDK for Node.js

This topic describes how to obtain and use Cloud Native Distributed Database PolarDB-X (PolarDB-X) SDK for Node.js.

Prerequisites

- You must have an Apsara Stack tenant account and an AccessKey pair (AccessKey ID and AccessKey secret) to use PolarDB-X SDK for Node.js. For more information, see Obtain an AccessKey pair.
- You have obtained the endpoint of Apsara Stack API (ASAPI). For more information, see Obtain the endpoint of Asapi.
- You have obtained the region ID. For more information, see Query the region ID.

Add the dependency

1. Install SDK for Node.js. `tnpm install --save @ali/asapi-nodejs-sdk`
2. If *mocha.opts* depends on Egg.js, configure *mocha.opts* as follows:

```
--recursive
--colors
--timeout 60000
--reporter mochawesome
--require co-mocha
--require egg-ts-helper/register
--require espower-typescript/guess
--require intelli-espower-loader
```

Use SDK for Node.js

The following example shows the three main steps of calling SDK for Node.js:

1. Create an ASClient instance. When you create an ASClient instance, set the region ID, endpoint, AccessKey ID, and AccessKey secret in the *config.ts* file. See the following example:

```
local: {
  regionId: the region ID.
  API_GATEWAY: the endpoint of ASAPI.
  accessKeyId: the AccessKey ID.
  accessKeySecret: the AccessKey secret.
},
```

2. Create an API request and configure parameters.
3. Initiate the API request and handle the response or exception.

```
// Set the endpoint of ASAPI.
const API_GATEWAY = "http://xxxx/asapi/v3"

// Create an RPC client.
const asapiClient = ctx.service.sdk.rpcClient;

// Set parameters.
let params = {
   Action: 'DescribeDrdsDbInstances',
   Product: 'PolarDB-X',
   regionId: 'cn-qingdao-env12-d01',
   accessKeyId: 'xxxxxxx',
   accessKeySecret: 'xxxxxxx',
   Version: '2019-01-23',
};

// Set parameters in the request header.
asapiClient.addHeader("x-acs-regionId", "cn-qingdao-env12-d01");

// Initiate the request and obtain the response.
const result = await asapiClient.doRequest(API_GATEWAY, params, {});
```

5.2.10. (Optional) Use POP SDK to call ASAPI API operations

If an application used POP SDK to call Cloud Native Distributed Database PolarDB-X (PolarDB-X), you must switch the gateway to Apsara Stack API (ASAPI).

Procedure

1. Switch the gateway to ASAPI and add common parameters required by the API operation to the request header, so the API operation can be called in a transparent manner.
2. Add the **Product** field to the request header.

Example

The following code is an example of calling a PolarDB-X API operation.

```
try {
    DescribeDrdsDbInstancesResponse 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());
}
```