

Alibaba Cloud

Apsara Stack Enterprise

Developer Guide - Analytics
and Artificial Intelligence

Product Version: 2109, Internal: V3.15.0

Document Version: 20211210

Legal disclaimer

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

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

Document conventions

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

Table of Contents

1. MaxCompute	22
1.1. Developer Guide	22
1.1.1. Java SDK	22
1.1.1.1. SDK for Java	22
1.1.1.2. Prerequisites	23
1.1.1.2.1. Overview	23
1.1.1.2.2. Obtain an AccessKey pair	23
1.1.1.2.2.1. Log on to the Apsara Uni-manager Managem...	23
1.1.1.2.2.2. Obtain the AccessKey pair of an organization	24
1.1.1.2.2.3. Obtain the AccessKey pair of an Alibaba Clou...	24
1.1.1.2.3. View the endpoint	25
1.1.1.3. Major APIs	27
1.1.1.3.1. AliyunAccount	27
1.1.1.3.2. Odps	27
1.1.1.3.3. Projects	28
1.1.1.3.4. Project	28
1.1.1.3.5. SQLTask	28
1.1.1.3.6. Instances	29
1.1.1.3.7. Instance	29
1.1.1.3.8. Tables	30
1.1.1.3.9. Table	30
1.1.1.3.10. Resources	31
1.1.1.3.11. Resource	31
1.1.1.3.12. Functions	32
1.1.1.3.13. Function	32
1.1.1.3.14. Spark Shell	33

1.1.1.3.15. Spark R	33
1.1.1.3.16. Spark SQL	34
1.1.1.3.17. Spark JDBC	34
1.1.1.3.18. Document APIs	34
1.1.1.3.19. Search APIs	35
1.1.1.3.20. CheckPermissionTest	35
1.1.2. SDK for Python	35
2.DataWorks	50
2.1. Developer Guide	50
2.1.1. Overview	50
2.1.2. Terms	50
2.1.3. Introduction	52
2.1.4. List of operations by function	61
2.1.5. Make API requests	62
2.1.5.1. Java SDK	63
2.1.5.2. HTTP API	66
2.1.5.2.1. Endpoints	66
2.1.5.2.2. Protocol	67
2.1.5.2.3. Request method	67
2.1.5.2.4. Encoding	67
2.1.5.2.5. Signature method	67
2.1.5.2.6. Response parameters	68
2.1.5.2.7. API example	68
2.1.6. Java SDK	69
2.1.6.1. Tenants	70
2.1.6.1.1. ListCalcEngines	70
2.1.6.1.2. CreateDataSource	80
2.1.6.1.3. ListDataSources	87

2.1.6.1.4. UpdateDataSource	94
2.1.6.1.5. DeleteDataSource	101
2.1.6.1.6. ListResourceGroups	103
2.1.6.1.7. GetProject	107
2.1.6.1.8. AddProjectMemberToRole	114
2.1.6.1.9. CreateProjectMember	115
2.1.6.1.10. DeleteProjectMember	117
2.1.6.1.11. ListProjectRoles	119
2.1.6.1.12. RemoveProjectMemberFromRole	121
2.1.6.1.13. ListProjectMembers	123
2.1.6.1.14. ListProjects	127
2.1.6.2. Metadata	129
2.1.6.2.1. GetMetaDBInfo	130
2.1.6.2.2. GetMetaDBTableList	134
2.1.6.2.3. CheckMetaTable	138
2.1.6.2.4. CheckMetaPartition	141
2.1.6.2.5. SearchMetaTables	144
2.1.6.2.6. GetMetaTableBasicInfo	149
2.1.6.2.7. GetMetaTableColumn	156
2.1.6.2.8. GetMetaTableFullInfo	161
2.1.6.2.9. GetMetaTablePartition	169
2.1.6.2.10. GetMetaTableOutput	174
2.1.6.2.11. GetMetaTableChangeLog	178
2.1.6.2.12. GetMetaTableIntroWiki	184
2.1.6.2.13. GetMetaTableLineage	188
2.1.6.2.14. GetMetaColumnLineage	192
2.1.6.2.15. CreateMetaCategory	197
2.1.6.2.16. DeleteMetaCategory	200

2.1.6.2.17. UpdateMetaCategory	204
2.1.6.2.18. GetMetaCategory	207
2.1.6.2.19. GetMetaTableListByCategory	212
2.1.6.2.20. AddToMetaCategory	216
2.1.6.2.21. DeleteFromMetaCategory	218
2.1.6.2.22. CreateTable	221
2.1.6.2.23. UpdateTable	226
2.1.6.2.24. UpdateTableAddColumn	232
2.1.6.2.25. DeleteTable	235
2.1.6.2.26. GetDDLJobStatus	238
2.1.6.2.27. UpdateMetaTable	240
2.1.6.2.28. UpdateMetaTableIntroWiki	242
2.1.6.2.29. ListMetaDB	244
2.1.6.2.30. UpdateTableModelInfo	249
2.1.6.2.31. CreateTableTheme	251
2.1.6.2.32. DeleteTableTheme	253
2.1.6.2.33. UpdateTableTheme	256
2.1.6.2.34. ListTableTheme	258
2.1.6.2.35. CreateTableLevel	262
2.1.6.2.36. DeleteTableLevel	264
2.1.6.2.37. UpdateTableLevel	266
2.1.6.2.38. ListTableLevel	269
2.1.6.3. DataStudio	272
2.1.6.3.1. CreateFile	272
2.1.6.3.2. CreateUdfFile	281
2.1.6.3.3. CreateBusiness	285
2.1.6.3.4. DeleteBusiness	288
2.1.6.3.5. CreateFolder	291

2.1.6.3.6. UpdateBusiness	294
2.1.6.3.7. ListFiles	296
2.1.6.3.8. EstablishRelationTableToBusiness	304
2.1.6.3.9. SubmitFile	306
2.1.6.3.10. ListBusiness	309
2.1.6.3.11. GetFileVersion	313
2.1.6.3.12. DeleteFolder	318
2.1.6.3.13. DeleteFile	320
2.1.6.3.14. GetDeployment	323
2.1.6.3.15. UpdateFile	327
2.1.6.3.16. GetFolder	335
2.1.6.3.17. ListFileVersions	338
2.1.6.3.18. GetFile	343
2.1.6.3.19. ListFolders	353
2.1.6.3.20. DeployFile	356
2.1.6.3.21. UpdateUdfFile	359
2.1.6.3.22. GetBusiness	363
2.1.6.3.23. UpdateFolder	367
2.1.6.4. Operation Center	370
2.1.6.4.1. ListAlertMessages	370
2.1.6.4.2. GetBaselineConfig	382
2.1.6.4.3. ListNodesByBaseline	387
2.1.6.4.4. ListBaselineConfigs	391
2.1.6.4.5. GetBaselineKeyPath	398
2.1.6.4.6. GetBaselineStatus	405
2.1.6.4.7. ListBaselineStatuses	414
2.1.6.4.8. DeleteRemind	420
2.1.6.4.9. UpdateRemind	423

2.1.6.4.10. CreateRemind	430
2.1.6.4.11. GetRemind	437
2.1.6.4.12. ListReminds	448
2.1.6.4.13. ListTopics	455
2.1.6.4.14. GetTopic	461
2.1.6.4.15. GetTopicInfluence	467
2.1.6.4.16. GetNode	472
2.1.6.4.17. ListNodeInputOrOutput	475
2.1.6.4.18. ListNodes	478
2.1.6.4.19. GetNodeCode	483
2.1.6.4.20. GetInstance	485
2.1.6.4.21. ListInstances	492
2.1.6.4.22. GetInstanceLog	501
2.1.6.4.23. StopInstance	503
2.1.6.4.24. RestartInstance	505
2.1.6.4.25. SetSuccessInstance	507
2.1.6.4.26. SuspendInstance	508
2.1.6.4.27. ResumeInstance	510
2.1.6.4.28. RunCycleDagNodes	512
2.1.6.4.29. RunSmokeTest	515
2.1.6.4.30. ListSuccessInstanceAmount	517
2.1.6.4.31. ListInstanceAmount	520
2.1.6.4.32. TopTenElapsedTimeInstance	523
2.1.6.4.33. TopTenErrorTimesInstance	527
2.1.6.4.34. GetFileTypeStatistic	531
2.1.6.4.35. GetInstanceStatusStatistic	534
2.1.6.4.36. ListFileType	536
2.1.6.4.37. RunManualDagNodes	541

2.1.6.4.38. GetManualDagInstances	543
2.1.6.4.39. ListManualDagInstances	549
2.1.6.4.40. GetDag	554
2.1.6.4.41. ListNodesByOutput	558
2.1.6.4.42. RunTriggerNode	563
2.1.6.4.43. GetNodeParents	566
2.1.6.4.44. GetNodeChildren	569
2.1.6.5. Data Quality	572
2.1.6.5.1. UpdateQualityFollower	572
2.1.6.5.2. UpdateQualityRule	574
2.1.6.5.3. GetQualityRule	577
2.1.6.5.4. ListQualityRules	582
2.1.6.5.5. CreateQualityRule	588
2.1.6.5.6. DeleteQualityFollower	592
2.1.6.5.7. DeleteQualityRule	593
2.1.6.5.8. GetQualityFollower	594
2.1.6.5.9. CreateQualityFollower	597
2.1.6.5.10. DeleteQualityEntity	599
2.1.6.5.11. GetQualityEntity	601
2.1.6.5.12. CreateQualityEntity	605
2.1.6.5.13. CreateQualityRelativeNode	607
2.1.6.5.14. DeleteQualityRelativeNode	609
2.1.6.5.15. ListQualityResultsByRule	611
2.1.6.5.16. ListQualityResultsByEntity	621
2.1.6.6. DataService Studio	631
2.1.6.6.1. CreateDataServiceApi	631
2.1.6.6.2. PublishDataServiceApi	638
2.1.6.6.3. DeleteDataServiceApi	639

2.1.6.6.4. ListDataServiceApis	641
2.1.6.6.5. GetDataServiceApi	669
2.1.6.6.6. UpdateDataServiceApi	696
2.1.6.6.7. GetDataServicePublishedApi	700
2.1.6.6.8. ListDataServicePublishedApis	732
2.1.6.6.9. ListDataServiceApiAuthorities	765
2.1.6.6.10. ListDataServiceAuthorizedApis	771
2.1.6.6.11. AbolishDataServiceApi	776
2.1.6.6.12. GetDataServiceApplication	777
2.1.6.6.13. ListDataServiceApplications	780
2.1.6.6.14. CreateDataServiceFolder	784
2.1.6.6.15. GetDataServiceFolder	786
2.1.6.6.16. ListDataServiceFolders	789
2.1.6.6.17. CreateDataServiceGroup	793
2.1.6.6.18. GetDataServiceGroup	795
2.1.6.6.19. ListDataServiceGroups	798
2.1.6.6.20. CreateDataServiceApiAuthority	802
2.1.6.6.21. DeleteDataServiceApiAuthority	804
2.1.7. Obtain an AccessKey pair	806
3.Realtime Compute	807
3.1. Developer Guide	807
3.1.1. Preparations	807
3.1.1.1. Log on to the API Tool console	807
3.1.1.2. View information about API operations	808
3.1.1.3. Obtain an AccessKey pair	808
3.1.1.4. Obtain common header parameters	810
3.1.1.5. Obtain an STS AccessKey pair	811
3.1.2. Use POP to make API calls	813

3.1.2.1. Preparations	813
3.1.2.2. SDK for Java	814
3.1.2.3. SDK for Go	818
3.1.2.4. SDK for Python	821
3.1.3. Use ASAPI to make API calls	823
3.1.3.1. ASAPI overview	823
3.1.3.2. Generate an endpoint for ASAPI	824
3.1.3.3. Obtain Apsara Stack SDKs	826
3.1.3.4. Common request parameters	826
3.1.3.5. SDK description	828
3.1.3.5.1. SDK for Java	828
3.1.3.5.2. SDK for Python	834
3.1.3.5.3. SDK for Node.js	836
3.1.3.6. HTTP requests	838
3.1.3.6.1. Overview	838
3.1.3.6.2. Request structure	838
3.1.3.6.3. Request signatures	839
3.1.3.6.4. Responses	844
3.1.4. Flink SQL	845
3.1.4.1. Overview	845
3.1.4.2. Keywords	845
3.1.4.3. Basic concepts	847
3.1.4.3.1. Time attributes	847
3.1.4.3.2. Watermark	850
3.1.4.3.3. Computed column	851
3.1.4.4. Data types	852
3.1.4.4.1. Data types	852
3.1.4.4.2. Data types and operators	854

3.1.4.5. DDL statements	855
3.1.4.5.1. Overview	855
3.1.4.5.2. Create a data source table	857
3.1.4.5.2.1. Overview	858
3.1.4.5.2.2. Create a DataHub source table	860
3.1.4.5.2.3. Create a Message Queue for Apache Kafka s... ..	865
3.1.4.5.2.4. Create an Oracle database source table	881
3.1.4.5.2.5. Create a Log Service source table	886
3.1.4.5.2.6. Create a full MaxCompute source table	890
3.1.4.5.2.7. Create a Message Queue source table	894
3.1.4.5.2.8. Create a Tablestore source table	899
3.1.4.5.2.9. Create a Hologres source table	901
3.1.4.5.2.10. Create an incremental MaxCompute source t... ..	904
3.1.4.5.3. Create a data result table	908
3.1.4.5.3.1. Overview	908
3.1.4.5.3.2. Create a DataHub result table	909
3.1.4.5.3.3. Create a Message Queue result table	913
3.1.4.5.3.4. Create an ApsaraDB RDS result table	915
3.1.4.5.3.5. Create a TSDB result table	920
3.1.4.5.3.6. Create an KVStore for Redis result table	923
3.1.4.5.3.7. Create an AnalyticDB for MySQL V2.0 result	928
3.1.4.5.3.8. Create an AnalyticDB for MySQL V3.0 result	931
3.1.4.5.3.9. Create a Log Service result table	934
3.1.4.5.3.10. Create a Tablestore result table	936
3.1.4.5.3.11. Create a MaxCompute result table	939
3.1.4.5.3.12. Create an Elasticsearch result table	946
3.1.4.5.3.13. Create a Message Queue for Apache Kafka	949
3.1.4.5.3.14. Create an ApsaraDB RDS for SQL Server res... ..	952

3.1.4.5.3.15. Create an ApsaraDB for MongoDB result tab..	957
3.1.4.5.3.16. Create a result table in an Oracle database	959
3.1.4.5.3.17. Create an InfluxDB result table	963
3.1.4.5.3.18. Create a Phoenix5 result table	965
3.1.4.5.3.19. Create a Hologres result table	966
3.1.4.5.3.20. Create an AnalyticDB for PostgreSQL result ..	971
3.1.4.5.3.21. Create an ApsaraDB for HBase result table	973
3.1.4.5.4. Create a data dimension table	979
3.1.4.5.4.1. Overview	979
3.1.4.5.4.2. Create a Tablestore dimension table	980
3.1.4.5.4.3. Create an ApsaraDB for HBase dimension tab..	982
3.1.4.5.4.4. Create a KVStore for Redis dimension table	989
3.1.4.5.4.5. Create an ApsaraDB RDS dimension table	992
3.1.4.5.4.6. Create a MaxCompute dimension table	997
3.1.4.5.4.7. Create an AnalyticDB for MySQL V3.0 dimens...	1003
3.1.4.5.4.8. Create a Hologres dimension table	1008
3.1.4.5.4.9. Create a Phoenix5 dimension table	1013
3.1.4.5.4.10. Create an Elasticsearch dimension table	1017
3.1.4.6. DML statements	1019
3.1.4.6.1. INSERT INTO statement	1019
3.1.4.7. Query statements	1020
3.1.4.7.1. SELECT statements	1020
3.1.4.7.2. WHERE	1022
3.1.4.7.3. HAVING	1023
3.1.4.7.4. GROUP BY	1024
3.1.4.7.5. JOIN statements	1025
3.1.4.7.6. JOIN statements for dimension tables	1028
3.1.4.7.7. UNION ALL	1030

3.1.4.7.8. TopN	1032
3.1.4.7.9. CEP statements	1038
3.1.4.8. Create a view	1045
3.1.4.9. Window functions	1047
3.1.4.9.1. Overview	1047
3.1.4.9.2. Tumbling window	1048
3.1.4.9.3. Sliding windows	1053
3.1.4.9.4. Session window	1056
3.1.4.9.5. OVER window	1059
3.1.4.10. Logical operations	1067
3.1.4.10.1. = ...	1067
3.1.4.10.2. >	1068
3.1.4.10.3. >=	1069
3.1.4.10.4. <	1069
3.1.4.10.5. <=	1070
3.1.4.10.6. <>	1071
3.1.4.10.7. AND	1072
3.1.4.10.8. BETWEEN AND	1073
3.1.4.10.9. IS NOT FALSE	1074
3.1.4.10.10. IS NOT NULL	1075
3.1.4.10.11. IS NOT TRUE	1075
3.1.4.10.12. IS NOT UNKNOWN	1076
3.1.4.10.13. IS NULL	1077
3.1.4.10.14. IS TRUE	1078
3.1.4.10.15. IS UNKNOWN	1078
3.1.4.10.16. LIKE	1079
3.1.4.10.17. NOT	1081
3.1.4.10.18. NOT BETWEEN AND	1081

3.1.4.10.19. OR	1083
3.1.4.10.20. IN	1083
3.1.4.10.21. NOT IN	1084
3.1.4.10.22. IS DISTINCT FROM	1085
3.1.4.10.23. IS NOT DISTINCT FROM	1086
3.1.4.11. Built-in functions	1087
3.1.4.11.1. String functions	1087
3.1.4.11.1.1. Overview	1087
3.1.4.11.1.2. CHAR_LENGTH	1092
3.1.4.11.1.3. CHR	1092
3.1.4.11.1.4. CONCAT	1093
3.1.4.11.1.5. CONCAT_WS	1094
3.1.4.11.1.6. FROM_BASE64	1095
3.1.4.11.1.7. HASH_CODE	1096
3.1.4.11.1.8. INITCAP	1096
3.1.4.11.1.9. JSON_VALUE	1097
3.1.4.11.1.10. KEYVALUE	1099
3.1.4.11.1.11. LOWER	1100
3.1.4.11.1.12. LPAD	1101
3.1.4.11.1.13. MD5	1102
3.1.4.11.1.14. OVERLAY	1103
3.1.4.11.1.15. PARSE_URL	1103
3.1.4.11.1.16. POSITION	1104
3.1.4.11.1.17. REGEXP	1105
3.1.4.11.1.18. REGEXP_EXTRACT	1106
3.1.4.11.1.19. REGEXP_REPLACE	1107
3.1.4.11.1.20. REPEAT	1109
3.1.4.11.1.21. REVERSE	1109

3.1.4.11.1.22. RPAD	1110
3.1.4.11.1.23. SPLIT_INDEX	1112
3.1.4.11.1.24. STR_TO_MAP	1113
3.1.4.11.1.25. SUBSTRING	1113
3.1.4.11.1.26. TO_BASE64	1115
3.1.4.11.1.27. TRIM	1115
3.1.4.11.1.28. UPPER	1116
3.1.4.11.2. Mathematical unctions	1117
3.1.4.11.2.1. Addition	1117
3.1.4.11.2.2. Subtraction	1117
3.1.4.11.2.3. Multiplication	1118
3.1.4.11.2.4. Division	1119
3.1.4.11.2.5. ABS	1119
3.1.4.11.2.6. ACOS	1120
3.1.4.11.2.7. BIN	1121
3.1.4.11.2.8. ASIN	1122
3.1.4.11.2.9. ATAN	1122
3.1.4.11.2.10. BITAND	1123
3.1.4.11.2.11. BITNOT	1124
3.1.4.11.2.12. BITOR	1124
3.1.4.11.2.13. BITXOR	1125
3.1.4.11.2.14. CARDINALITY	1126
3.1.4.11.2.15. COS	1126
3.1.4.11.2.16. COT	1127
3.1.4.11.2.17. E	1128
3.1.4.11.2.18. EXP	1129
3.1.4.11.2.19. FLOOR	1129
3.1.4.11.2.20. LN	1130

3.1.4.11.2.21. LOG	1131
3.1.4.11.2.22. LOG10	1132
3.1.4.11.2.23. LOG2	1132
3.1.4.11.2.24. PI	1133
3.1.4.11.2.25. POWER	1134
3.1.4.11.2.26. RAND	1134
3.1.4.11.2.27. SIN	1135
3.1.4.11.2.28. SQRT	1136
3.1.4.11.2.29. TAN	1136
3.1.4.11.2.30. CEIL	1137
3.1.4.11.2.31. CHARACTER_LENGTH	1138
3.1.4.11.2.32. DEGREES	1138
3.1.4.11.2.33. MOD	1139
3.1.4.11.2.34. ROUND	1140
3.1.4.11.3. Date functions	1141
3.1.4.11.3.1. CURRENT_TIMESTAMP	1141
3.1.4.11.3.2. DATEDIFF	1141
3.1.4.11.3.3. DATE_ADD	1142
3.1.4.11.3.4. DATE_FORMAT	1143
3.1.4.11.3.5. DATE_SUB	1144
3.1.4.11.3.6. DAYOFMONTH	1145
3.1.4.11.3.7. FROM_UNIXTIME	1146
3.1.4.11.3.8. HOUR	1146
3.1.4.11.3.9. LOCALTIME	1147
3.1.4.11.3.10. MINUTE	1148
3.1.4.11.3.11. MONTH	1148
3.1.4.11.3.12. NOW	1149
3.1.4.11.3.13. SECOND	1150

3.1.4.11.3.14. TIMESTAMPADD	1151
3.1.4.11.3.15. TO_DATE	1152
3.1.4.11.3.16. TO_TIMESTAMP	1153
3.1.4.11.3.17. UNIX_TIMESTAMP	1154
3.1.4.11.3.18. WEEK	1154
3.1.4.11.3.19. YEAR	1155
3.1.4.11.4. Conditional functions	1156
3.1.4.11.4.1. CASE WHEN	1156
3.1.4.11.4.2. COALESCE	1158
3.1.4.11.4.3. IF	1159
3.1.4.11.4.4. IS_ALPHA	1160
3.1.4.11.4.5. IS_DECIMAL	1160
3.1.4.11.4.6. IS_DIGIT	1161
3.1.4.11.4.7. NULLIF	1162
3.1.4.11.5. Table-valued functions	1162
3.1.4.11.5.1. GENERATE_SERIES	1162
3.1.4.11.5.2. JSON_TUPLE	1163
3.1.4.11.5.3. STRING_SPLIT	1164
3.1.4.11.6. Type conversion functions	1165
3.1.4.11.6.1. CAST	1165
3.1.4.11.7. Aggregate functions	1166
3.1.4.11.7.1. AVG	1166
3.1.4.11.7.2. CONCAT_AGG	1167
3.1.4.11.7.3. COUNT	1168
3.1.4.11.7.4. FIRST_VALUE	1168
3.1.4.11.7.5. LAST_VALUE	1170
3.1.4.11.7.6. MAX	1171
3.1.4.11.7.7. MIN	1172

3.1.4.11.7.8. STDDEV_POP	1173
3.1.4.11.7.9. SUM	1174
3.1.4.11.7.10. VAR_POP	1175
3.1.4.11.8. Other functions	1176
3.1.4.11.8.1. UUID	1176
3.1.4.12. UDXs	1176
3.1.4.12.1. Overview	1176
3.1.4.12.2. UDF	1179
3.1.4.12.3. UDAF	1182
3.1.4.12.4. UDTF	1187
3.1.4.12.5. Develop a UDX by using IntelliJ IDEA	1192
4.DataHub	1197
4.1. Developer Guide	1197
4.1.1. SDK overview	1197
4.1.2. Prerequisites	1197
4.1.2.1. Overview	1197
4.1.2.2. Obtain an AccessKey pair	1197
4.1.2.2.1. Log on to the Apsara Uni-manager Management...	1197
4.1.2.2.2. Obtain the AccessKey pair of an organization	1198
4.1.2.2.3. Obtain the AccessKey pair of an Alibaba Cloud ...	1199
4.1.2.3. Obtain an endpoint	1199
4.1.3. DataHub SDK for Java	1202
4.1.3.1. Overview	1202
4.1.3.2. Preparations	1202
4.1.3.3. Ingest tuple data	1203
4.1.3.3.1. Create a tuple topic	1203
4.1.3.3.2. Obtain the list of shards	1203
4.1.3.3.3. Write data into DataHub	1203

4.1.3.3.4. Consume data	1204
4.1.3.4. Ingest blob data	1205
4.1.3.4.1. Create a blob topic	1205
4.1.3.4.2. Obtain the list of shards	1205
4.1.3.4.3. Write data into DataHub	1206
4.1.3.4.4. Consume data	1206
4.1.3.5. Data ingestion methods	1207
4.1.3.5.1. Ingest data based on shard IDs	1207
4.1.3.5.2. Ingest data based on hash values	1208
4.1.3.5.3. Ingest data based on partition keys	1208
4.1.3.6. Create a DataConnector	1209
4.1.3.6.1. Create a DataConnector for MaxCompute	1209
4.1.3.6.2. Create a DataConnector for AnalyticDB or Apsara...	1210
4.1.4. DataHub SDK for Python	1211
4.1.4.1. Install DataHub SDK for Python	1211
4.1.4.2. Preparations	1212
4.1.4.3. Create topics	1212
4.1.4.4. Write records into DataHub and subscribe applicati...	1214
4.1.5. Scalability	1216
4.1.5.1. Scenarios	1216
4.1.5.2. Obtain shard details	1217
4.1.5.3. Split a shard	1217
4.1.5.4. Merge shards	1217
4.1.6. Insert a column to a topic schema	1218
4.1.6.1. Insert a column	1218

1. MaxCompute

1.1. Developer Guide

1.1.1. Java SDK

1.1.1.1. SDK for Java

This topic provides the overall information about the SDK packages provided by MaxCompute. It also provides a link for you to obtain complete API information.

For complete API information, see [ODPS SDK Core 0.32.5-public API](#).

You can use Maven to configure the version of the SDK that you want to use. For example, to specify the version of odps-sdk-core that you want to use, add the following Maven dependency:

```
<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>odps-sdk-core</artifactId>
  <version>0.36.2</version>
</dependency>
```

The following table describes the SDK packages provided by MaxCompute.

Package	Description
odps-sdk-core	Basic features of MaxCompute. This package encapsulates MaxCompute features and APIs, such as Odps, Project, and Table. It also includes Tunnel-related features.
odps-sdk-core-internal	Extended features of MaxCompute. This package encapsulates the less commonly used MaxCompute features and operations, such as Event and XFlow.
odps-sdk-commons	MaxCompute infrastructure. This package includes TableSchema, Column, Record, OdpsType, and some util packages.
odps-sdk-udf	The UDF API for MaxCompute.
odps-sdk-mapred	The MapReduce API for MaxCompute.
odps-sdk-graph	The Graph API for MaxCompute.

You can visit the [Maven repository](#) to download these SDK packages.

 **Note** If your MaxCompute version is earlier than V3.8.0, download the SDK packages of the 0.27.2-public version. If your MaxCompute version is a version from V3.8.0 to V3.11.0, download the SDK packages of the 0.30.8-public or 0.30.9-public version. If your MaxCompute version is V3.12.0 or later, download the SDK packages of the 0.36.2-public version.

1.1.1.2. Prerequisites

1.1.1.2.1. Overview

Before you use MaxCompute SDKs, you must prepare the required environment.

- Create an Alibaba Cloud account that is authorized to access MaxCompute and obtain an AccessKey pair that consists of an AccessKey ID and an AccessKey secret.
- Obtain the endpoint of MaxCompute.

1.1.1.2.2. Obtain an AccessKey pair

1.1.1.2.2.1. Log on to the Apsara Uni-manager

Management Console

This topic describes how to log on to the Apsara Uni-manager Management Console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations

in the following scenarios:

- It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the account and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
- You have enabled MFA and bound an MFA device.
Enter a six-digit MFA authentication code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Operations Console User Guide*.

1.1.1.2.2.2. Obtain the AccessKey pair of an organization

This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only the operation administrators and level-1 organization administrators can obtain the AccessKey pair of an organization.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, find the organization that you want to add and click the  icon on the right of an organization.
5. Select **AccessKey** from the drop-down list.
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 of the level-1 organization use the AccessKey pair of the level-1 organization.

1.1.1.2.2.3. Obtain the AccessKey pair of an Alibaba

Cloud account

To secure cloud resources, the system must verify the identity of visitors and ensure that they have the relevant permissions. You must obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account to access cloud resources. This topic describes how to obtain the AccessKey pair of an Alibaba Cloud account.

Procedure

1. Log on to the Apsara Uni-manager Management 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 your AccessKey pair.

Apsara Stack AccessKey Pair You must use the AccessKey pair when you access Apsara Stack resources.		
The AccessKey pair including the AccessKey ID and AccessKey secret is the credential to for you to use Apsara Stack resources with full permissions. You must keep the AccessKey pair confidential.		
Region	AccessKey ID	AccessKey Secret
cn-qingdao-env4b-d01		Show

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

1.1.1.2.3. View the endpoint

This topic describes how to obtain the endpoint of a DataHub machine in the Apsara Uni-manager Operations Console.

Context

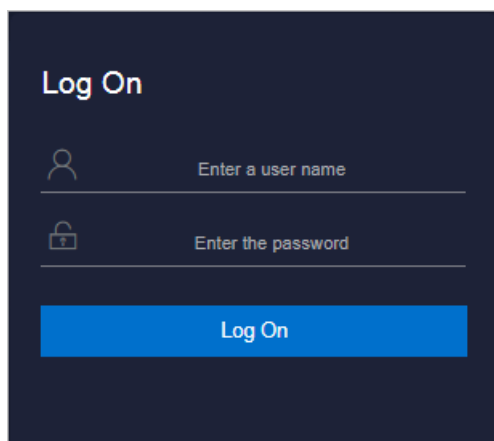
- The URL, username, and password that are used to log on to the Apsara Uni-manager Operations Console are obtained from the deployment engineers or administrators.

The URL of the Apsara Uni-manager Operations Console is in the following format: *ops.asconsole.intranet-domain-id.com*.

- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. Open your browser. In the address bar, enter the URL (*ops.asconsole.intranet-domain-id.com*). Then, 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 To obtain the username and password that are used to log on to the Apsara Uni-manager Operations Console, contact the deployment engineers or administrators.

If you log on to the Apsara Uni-manager Operations Console for the first time, you must change the password of your username.

To ensure the security of your account, make sure that the password meet the following requirements:

- The password contains uppercase or lowercase letters.
- The password contains digits.
- The password contains special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).
- The password is 10 to 20 characters in length.

3. Click **Log On** to go to the Apsara Uni-manager Operations Console.
4. In the top navigation bar of the Apsara Uni-manager Operations Console, click **O&M**. In the left-side navigation pane, choose **Product Management > Products**. In the **Apsara Stack O&M** section, click **Apsara Infrastructure Management Framework**.
5. In the left-side navigation pane, choose **Operations > Machine Operations**. The **Machine Operations** page appears.
6. Find the machine to which you want to log on, and click **Terminal** in the Actions column.

Machines

Project: Clusters: Machines:

☐	Hostname	Clusters	Project	Region	Status	Machine Metrics	Actions
☐	62 10.106.6.8	vpcAzoneCluster-A-20210812-00bc	vpc	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.15	regionCluster-A-20210812-00be	dnsProduct	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.5	slbCluster-A-20210812-00c3	slb	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.2	slbCluster-A-20210812-00c3	slb	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.9	ads-A-20210812-00c8	ads	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.10	ads-A-20210812-00c8	ads	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.4	slbCluster-A-20210812-00c3	slb	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management
☐	62 10.106.6.13	ads-A-20210812-00c8	ads	cn-qingdao-env211-d02	Normal Details	View	Operations Terminal Machine Management

7. Run the following command on the machine to obtain JSON data:

```
curl -l "http://127.0.0.1:7070/api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration"
```

```
[admin@ ~]$ curl -s http://10.10.10.10:8080/api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration | grep frontend
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current	
			Dload	Upload	Total	Spent	Left	Speed
100	3973	0	3973	0	122k	0	--:--:-- --:--:-- --:--:--	122k

```
"c.sr.id": "datahub-frontend",
  "c.sr.service_registration": "{\"datahub_frontend.endpoint\":\"datahub.cn-qingdao-env12-d01.dh.env12.shuguang.com\", \"datahub_frontend.port\":\"80,443\"}",
  "c.sr.service_registration": "{\"biggraph_dbhost\":\"biggraph.mysql.minirds.env12.ops.shuguang.com\", \"biggraph_dbname\":\"biggraph\", \"biggraph_dbpasswd\":\"xod2Cfsc8Rueboqd\", \"biggraph_dbport\":\"3122\", \"biggraph_dbuser\":\"biggraph\", \"biggraph_frontend_server.endpoint\":\"biggraph.cn-qingdao-env12-d01.odps.env12.ops.shuguang.com\", \"biggraph_frontend_server.httpsport\":\"443\", \"biggraph_frontend_server.port\":\"80\", \"biggraph_frontend_server_public.endpoint\":\"biggraph.cn-qingdao-env12-d01.odps.env12.shuguang.com\", \"biggraph_frontend_server_public.httpsport\":\"443\", \"biggraph_frontend_server_public.port\":\"80\", \"biggraph_project\":\"biggraph_internal_project\", \"cluster.name\":\"HybridOdpsCluster-A-20181110-d3cc\", \"quota.id\":\"9249\", \"quota.name\":\"biggraph_quota\"}"",
  "c.sr.id": "odps-service-frontend",
  "c.sr.service_registration": "{\"odps_frontend_server.endpoint\":\"service.cn-qingdao-env12-d01.odps.env12.ops.shuguang.com\", \"odps_frontend_server.httpsport\":\"443\", \"odps_frontend_server.port\":\"80\", \"odps_frontend_server.vip\":\"\", \"odps_frontend_server_internet.endpoint\":\"service.cn-qingdao-env12-d01.odps.public.env12.shuguang.com\", \"odps_frontend_server_internet.httpsport\":\"443\", \"odps_frontend_server_internet.ip\":\"\", \"odps_frontend_server_internet.port\":\"80\", \"odps_frontend_server_public.endpoint\":\"service.cn-qingdao-env12-d01.odps.env12.shuguang.com\", \"odps_frontend_server_public.httpsport\":\"443\", \"odps_frontend_server_public.ip\":\"\", \"odps_frontend_server_public.port\":\"80\", \"tunnel_frontend_server.endpoint\":\"\"}"
```

- Find the endpoint of *DataHub* based on the service ID (*datahub-frontend*) and the key value (*datahub_frontend.endpoint*) of *DataHub*.

```
[admin:~]
$curl -l /api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration | grep
databuh
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left     Speed

100 3973    0 3973    0  0 75893    0  --:--:-- --:--:-- --:--:-- 75893
d",
  "c.sr.service_registration": "{\\\"databuh_frontend.endpoint\\\": \"databuh.cn-qingdao-env12-d01.dh.env12.shu
guang.com\\\", \\\"databuh_frontend.port\\\": \\\"80,443\\\"}"
  "c.sr.id": "databuh-service",
  "c.sr.service_registration": "{\\\"admin_account.accesskey-id\\\": \\\"FuZrpsHFw4pKFaCm\\\", \\\"admin_account.access
key-secret\\\": \\\"Hgppfxke5b0XTx2JDCnD9CrldlTBwf\\\", \\\"admin_account.id\\\": \\\"1154035346989039\\\", \\\"admin_account.passwor
d\\\": \\\"Sqgm0moaHnwyjao6agfw\\\", \\\"admin_account.user\\\": \\\"databuh_admin@aliyun.com\\\", \\\"databuh_service_suffix\\\": \\\"x\\\",
\\\"rds_db_name\\\": \\\"databuh_db\\\", \\\"rds_host\\\": \\\"databuh-db.mysql.minirds.env12.ops.shuguang.com\\\", \\\"rds_passwd\\\": \\\"
ruevhgWzdy8Uc5vu\\\", \\\"rds_port\\\": \\\"3117\\\", \\\"rds_user\\\": \\\"databuh_db\\\"}"
  "c.sr.id": "databuh-webconsole",
  "c.sr.service_registration": "{\\\"databuh_webconsole.endpoint\\\": \\\"databuh.cn-qingdao-env12-d01.webconsole.
env12.shuguang.com\\\", \\\"databuh_webconsole.port\\\": \\\"80,443\\\"}"
100 141k  0 141k  0  0 2652k    0  --:--:-- --:--:-- --:--:-- 133M
```

1.1.1.3. Major APIs

1.1.1.3.1. AliyunAccount

This topic describes the AliyunAccount API provided by SDK for Java.

You can call this API to obtain your Alibaba Cloud account. The input parameters are `accessId` and `accessKey`. This API is also considered a class that is used to initialize MaxCompute.

 Note accessId indicates the AccessKey ID that uniquely identifies an Alibaba Cloud account. accessKey indicates the AccessKey secret that is used to authenticate the identity of the account.

1.1.1.3.2. Odps

This topic describes the Odps API provided by SDK for Java.

This API is also considered a class that is the entry point of MaxCompute SDK for Java. This class allows you to access all types of objects in MaxCompute, including projects, tables, resources, functions, and instances.

You can pass this class into the AliyunAccount instance to construct a MaxCompute object. Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
odps.setDefaultProject("my_project");
for (Table t : odps.tables()) {
    ....
}
```

1.1.1.3.3. Projects

This topic describes the Projects API provided by SDK for Java.

This API is a collection of all projects in MaxCompute.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
Project p = odps.projects().get("my_exists");
p.reload();
Map<String, String> properties = prj.getProperties();
...
```

1.1.1.3.4. Project

This topic describes the Project API provided by SDK for Java.

This API provides descriptions of projects. You can call the Projects API to obtain the projects.

1.1.1.3.5. SQLTask

This topic describes the SQL task API provided by SDK for Java.

You can call this API to run and process SQL tasks. You can use SQLTask.run to run SQL tasks. SQLTask.run returns an instance, and obtains the running status and results of SQL tasks from the instance.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key"); Odps odps = new Odps(account);
;
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
Instance instance = SQLTask.run(odps, "my_project", "select ...");
String id = instance.getId();
instance.waitForSuccess();
Set<String> taskNames = instance.getTaskNames();
for (String name
taskNames) {
TaskSummary summary = instance.getTaskSummary(name);
String s = summary.getSummaryText();
}
Map<String, String> results = instance.getTaskResults();
Map<String, TaskStatus> taskStatus = instance.getTaskStatus();
for (Entry<String, TaskStatus> status : taskStatus.entrySet()) {
String result = results.get(status.getKey());
}
}
```

 **Note** To create a table, you must call the SQL task API instead of the Table API. In this case, you must pass the CREATE TABLE statement into SQL task.

1.1.1.3.6. Instances

This topic describes the Instances API provided by SDK for Java.

This API is a collection of all instances in MaxCompute.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
odps.setDefaultProject("my_project");
for (Instance i : odps.instances ()) {
....
}
}
```

1.1.1.3.7. Instance

This topic describes the Instance API provided by SDK for Java.

This API provides descriptions of instances. You can call the Instances API to obtain the instances.

Sample code:

```

Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
Instance ins = odps.instances().get("instance id");
Date startTime = instance.getStartTime();
Date endTime = instance.getEndTime();
...
Status instanceStatus = instance.getStatus();
String instanceStatusStr = null;
if (instanceStatus == Status.TERMINATED) {
    instanceStatusStr = TaskStatus.Status.SUCCESS.toString();
    Map<String, TaskStatus> taskStatus = instance.getTaskStatus();
    for (Entry<String, TaskStatus> status : taskStatus.entrySet()) {
        if (status.getValue().getStatus() != TaskStatus.Status.SUCCESS) {
            instanceStatusStr = status.getValue().getStatus().toString();
            break;
        }
    }
} else {
    instanceStatusStr = instanceStatus.toString();
}
...
TaskSummary summary = instance.getTaskSummary("instance name");
String s = summary.getSummaryText();

```

1.1.1.3.8. Tables

This topic describes the Tables API provided by SDK for Java.

This API is a collection of tables in MaxCompute.

Sample code:

```

Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
odps.setDefaultProject("my_project");
for (Table t : odps.tables()) {
    ....
}

```

1.1.1.3.9. Table

This topic describes the Table API provided by SDK for Java.

This API provides descriptions of tables in MaxCompute. You can call the Tables API to obtain the tables.

Sample code:


```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
Table t = odps.tables().get("table name");
t.reload();
Partition part = t.getPartition(new PartitionSpec(tableSpec[1]));
part.reload();
...
```

1.1.1.3.10. Resources

This topic describes the Resources API provided by SDK for Java.

This API is a collection of resources in MaxCompute.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
odps.setDefaultProject("my_project");
for (Resource r : odps.resources()) {
    ....
}
```

1.1.1.3.11. Resource

This topic describes the Resource API provided by SDK for Java.

This API provides descriptions of resources. You can call the Resources API to obtain the resources.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
Resource r = odps.resources().get("resource name");
r.reload();
if (r.getType() == Resource.Type.TABLE) { TableResource tr = new TableResource(r);
String tableSource = tr.getSourceTable().getProject() + "." + tr.getSourceTable().getName();
if (tr.getSourceTablePartition() != null) {
    tableSource += " partition(" + tr.getSourceTablePartition().toString() + ")";
}
....
}
```

The following sample code shows how to create a file resource:

```
String projectName = "my_project";
String source = "my_local_file.txt";
File file = new File(source);
InputStream is = new FileInputStream(file);
FileResource resource = new FileResource();
String name = file.getName();
resource.setName(name);
odps.resources().create(projectName, resource, is);
```

The following sample code shows how to create a table resource:

```
TableResource resource = new TableResource(tableName, tablePrj, partitionSpec);
//resource.setName(INVALID_USER_TABLE);
resource.setName("table_resource_name");
odps.resources().update(projectName, resource);
```

1.1.1.3.12. Functions

This topic describes the Functions API provided by SDK for Java.

This API is a collection of functions in MaxCompute.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
odps.setDefaultProject("my_project");
for (Function f : odps.functions()) {
    ....
}
```

1.1.1.3.13. Function

This topic describes the Function API provided by SDK for Java.

This API provides descriptions of functions. You can call the Functions API to obtain the functions.

Sample code:

```
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
String odpsUrl = "<your odps endpoint>";
odps.setEndpoint(odpsUrl);
Function f = odps.functions().get("function name");
List<Resource> resources = f.getResources();
```

The following sample code shows how to create a function:

```
String resources = "xxx:xxx";
String classType = "com.aliyun.odps.mapred.open.example.WordCount";
ArrayList<String> resourceList = new ArrayList<String>();
for (String r : resources.split(":")) {
    resourceList.add(r);
}
Function func = new Function();
func.setName(name);
func.setClassType(classType);
func.setResources(resourceList);
odps.functions().create(projectName, func);
```

1.1.1.3.14. Spark Shell

This topic describes the Spark Shell API provided by SDK for Java.

This API is commonly used by Spark to submit tasks to a cluster.

Run the following commands to enable the API:

```
$cd $SPARK_HOME
-- Access the Spark directory.
$bin/spark-shell --master yarn
-- Select the running mode and enable the API.
```

Sample code:

```
sc.parallelize(0 to 100, 2).collect
sql("show tables").show
sql("select * from spark_user_data").show(200,100)
```

1.1.1.3.15. Spark R

This topic describes the Spark R API provided by SDK for Java.

This API is commonly used by Spark to submit tasks to a cluster.

Run the following commands to enable the API:

```
$mkdir -p /home/admin/R && unzip ./R/R.zip -d /home/admin/R/
-- Create a directory named R and decompress R.zip in the directory.
$export PATH=/home/admin/R/bin/:$PATH
-- Configure environment variables.
$bin/sparkR --master yarn --archives ./R/R.zip
-- Select the running mode and enable the API.
```

Sample code:

```
df<- as.DataFrame(faithful)
df
head(select(df, df$eruptions))
head(select(df, "eruptions"))
head(filter(df, df$waiting < 50))
results <- sql("FROM spark_user_data SELECT *")
head(results)
```

1.1.1.3.16. Spark SQL

This topic describes the Spark SQL API provided by SDK for Java.

This API is commonly used by Spark to submit tasks to a cluster.

Run the following commands to enable the API:

```
$cd $SPARK_HOME
-- Access the Spark directory.
$bin/spark-sql --master yarn
-- Select the running mode and enable the API.
```

Sample code:

```
show tables;
select * from spark_user_data limit 3;
quit;
```

1.1.1.3.17. Spark JDBC

This topic describes the Spark JDBC API provided by SDK for Java.

This API is commonly used by Spark to submit tasks to a cluster.

Run the following commands to enable the API:

```
$sbin/stop-thriftserver.sh
-- Stop a thread.
/sbin/start-thriftserver.sh
-- Restart a thread.
$bin/beeline
-- Enable the API.
```

Sample code:

```
! connect jdbc:hive2://localhost:10000/odps_smoke_test
show tables;
select * from mr_input limit 3;
! quit
```

1.1.1.3.18. Document APIs

This topic describes the Document APIs provided by SDK for Java.

It is an open source API repository that is compatible with Elasticsearch on MaxCompute. For more information, see [Document APIs](#).

1.1.1.3.19. Search APIs

This topic describes the Search APIs provided by SDK for Java.

It is an open source API repository that is compatible with Elasticsearch on MaxCompute. For more information, see [Search APIs](#).

1.1.1.3.20. CheckPermissionTest

This topic describes the CheckPermissionTest API provided by SDK for Java.

You can call this API to check whether a specified user has the specified permission on a resource.

Sample code:

```
Account account = new AliyunAccount("accessId", "accessKey");
Odps odps = new Odps(account);
SecurityManager sm = odps.projects().get().getSecurityManager();
SecurityManager.PermissionDesc desc = new SecurityManager.PermissionDesc(
    odps.getDefaultProject(),
    CheckPermissionConstants.ObjectType.Table,
    "table_name",
    CheckPermissionConstants.ActionType.Describe);
desc.setColumns(Arrays.asList("col1", "col2"));
desc.setPrincipal("ALIYUN$test@aliyun.com");
SecurityManager.CheckPermissionResultInfo res = sm.checkPermission(desc);
System.out.println("Result: " + res.getResult());
System.out.println("Message: " + res.getMessage());
```

1.1.2. SDK for Python

This topic describes how to use SDK for Python to create, view, and delete tables.

PyODPS is MaxCompute SDK for Python. It provides easy-to-use Python programming interfaces. Similar to pandas, PyODPS provides fast, flexible, and expressive data structures. You can use the data processing feature of PyODPS, which is similar to that of pandas, by calling the DataFrame API provided by PyODPS. For more information, see [aliyun-odps-python-sdk](#) and [PyODPS: ODPS Python SDK and data analysis framework](#).

For more information about PyODPS, see the relevant description in *MaxCompute User Guide*.

The following sections describe how to use SDK for Python to create, view, and delete tables.

Install PyODPS

PyODPS supports Python 2.6 and later (including Python 3.0). After you install pip in the system, you need only to run the `pip install pyodps` command. After you run this command, related dependencies of PyODPS are automatically installed. For more information, see the **Installation guide** section in *MaxCompute User Guide*.

Perform initialization

Use your Apsara Stack tenant account to initialize a MaxCompute entry object. Example:

```
from odps import ODPS
odps = ODPS(**your-access-id**, **your-secret-access-key**, **your-default-project**,
            endpoint=**your-end-point**)
```

After initialization, you can perform operations on tables, resources, and functions.

 **Note** Unless otherwise specified, objects in the following sections all refer to MaxCompute entry objects.

Obtain a PyODPS project

A project is the basic organizational unit in MaxCompute, similar to a database.

You can call the `get_project` method to obtain information about a project. Example:

```
project = odps.get_project('my_project') # Obtain information about a project.
project = odps.get_project()             # Obtain information about the default project.
```

 **Note**

- If no parameters are specified, information about the default project is obtained.
- You can call the `exist_project` method to check whether a project exists.
- A table is the data storage unit in MaxCompute.

Perform operations on PyODPS tables

Call the `list_tables` method to query all tables in a project. Example:

```
for table in odps.list_tables(): # Process each table.
```

Call the `exist_table` method to check whether a table exists. Call the `get_table` method to obtain information about a table.

```
t = odps.get_table('dual')
t.schema
odps.Schema {
  c_int_a      bigint
  c_int_b      bigint
  c_double_a   double
  c_double_b   double
  c_string_a   string
  c_string_b   string
  c_bool_a     boolean
  c_bool_b     boolean
  c_datetime_a datetime
  c_datetime_b datetime
}
t.lifecycle
-1
print(t.creation_time)
2014-05-15 14:58:43
t.is_virtual_view
False
t.size
1408
t.schema.columns
[<column c_int_a, type bigint>,
 <column c_int_b, type bigint>,
 <column c_double_a, type double>,
 <column c_double_b, type double>,
 <column c_string_a, type string>,
 <column c_string_b, type string>,
 <column c_bool_a, type boolean>,
 <column c_bool_b, type boolean>,
 <column c_datetime_a, type datetime>,
 <column c_datetime_b, type datetime>]
```

Create the schema of a PyODPS table

Use one of the following methods to initialize the schema of a PyODPS table:

- Initialize the schema by using table columns and optional partitions.

```
from odps.models import Schema, Column, Partition
columns = [Column(name='num', type='bigint', comment='the column')]
partitions = [Partition(name='pt', type='string', comment='the partition')]
schema = Schema(columns=columns, partitions=partitions)
schema.columns
[<column num, type bigint>, <partition pt, type string>]
```

- Call the `Schema.from_lists` method. Although this method is more convenient, you cannot directly set comments for columns and partitions when you use this method.

```
schema = Schema.from_lists(['num'], ['bigint'], ['pt'], ['string'])
schema.columns
[<column num, type bigint>, <partition pt, type string>]
```


Create a PyODPS table

You can use a table schema to create a table. Example:

```
table = odps.create_table('my_new_table', schema)
table = odps.create_table('my_new_table', schema, if_not_exists=True) # Create a table only if no table with the same name exists.
table = o.create_table('my_new_table', schema, lifecycle=7) # Set the lifecycle of the table.
```

You can also use a comma-delimited string of field names and field data types to create a table. Example:

```
# Create a non-partitioned table.
table = o.create_table('my_new_table', 'num bigint, num2 double', if_not_exists=True)
# Create a partitioned table with table columns and partitions specified.
table = o.create_table('my_new_table', ('num bigint, num2 double', 'pt string'), if_not_exists=True)
```

When you create a table, you can only use the BIGINT, DOUBLE, DECIMAL, STRING, DATETIME, BOOLEAN, MAP, and ARRAY data types, unless otherwise specified.

To use other data types, such as TINYINT and STRUCT, you can set `options.sql.use_odps2_extension` to True. Example:

```
from odps import options
options.sql.use_odps2_extension = True
table = o.create_table('my_new_table', 'cat smallint, content struct<title:varchar(100), body string>')
```

Obtain PyODPS table data

Use one of the following methods to obtain table data:

- Call the `head` method to retrieve up to the first 10,000 data records in a table. Example:

```
t = odps.get_table('dual')
for record in t.head(3):
    print(record[0]) # Obtain the value at position 0.
    print(record['c_double_a']) # Obtain a value based on a field.
    print(record[0:3]) # Perform slicing operations.
    print(record[0,2,3]) # Obtain values at multiple positions.
    print(record['c_int_a', 'c_double_a']) # Obtain values based on multiple fields.
```

- Call the `open_reader` method for a table to open a reader and use the reader to read the table data. You can open the reader with or without a WITH clause.

```
# Open the reader with a WITH clause:
with t.open_reader(partition='pt=test') as reader:
    count = reader.count
    for record in reader[5:10] # You can execute the statement multiple times until all records are read. The
                                number of records is specified by count. You can change it to parallel-operation code.
        # Process a record.
# Open the reader without a WITH clause:
reader = t.open_reader(partition='pt=test')
count = reader.count
for record in reader[5:10]
    # Process a record.
```

- Call the Tunnel API to read table data. The `open_reader` method encapsulates the Tunnel API.

Use PyODPS to write data

Similar to `open_reader`, you can use `open_writer` for a table to open a writer and use the writer to write data to a table. Example:

```
# Open the writer with a WITH clause:
with t.open_writer(partition='pt=test') as writer:
    writer.write(records) # records can be iterable records and are written to block 0 by default.
with t.open_writer(partition='pt=test', blocks=[0, 1]) as writer: # Enable two blocks.
    writer.write(0, gen_records(block=0))
    writer.write(1, gen_records(block=1)) # The two write operations can be performed at the same time by u
    sing the multithreading technology. Each block is independent.
# Open the writer without a WITH clause:
writer = t.open_writer(partition='pt=test', blocks=[0, 1])
writer.write(0, gen_records(block=0))
writer.write(1, gen_records(block=1))
writer.close() # You must close the writer. Otherwise, the written data may be incomplete.
```

Writing data to a table is a process that encapsulates the Tunnel API.

Delete a PyODPS table

Delete a table. Example:

```
odps.delete_table('my_table_name', if_exists=True) # Delete a table only if the table exists.
t.drop() # Call the drop() method to delete a table if the table exists.
```

Perform operations on PyODPS table partitions

- Basic operations

Traverse all partitions of a table. Example:

```
for partition in table.partitions:
    print(partition.name)
for partition in table.iterate_partitions(spec='pt=test'):
    # Traverse subpartitions.
```

Check whether a partition exists. Example:

```
table.exist_partition('pt=test,sub=2019')
```

Obtain information about a partition. Example:

```
partition = table.get_partition('pt=test')
print(partition.creation_time)
2019-09-18 22:22:27
partition.size
0
```

- Create a partition

```
t.create_partition('pt=test', if_not_exists=True) # Create a partition only if no partition with the same name exists.
```

- Delete a partition

```
t.delete_partition('pt=test', if_exists=True) # Delete a partition only if the partition exists.
Use the drop() method to delete a partition only if the partition exists.
```

PyODPS SQL

PyODPS supports MaxCompute SQL queries and provides methods to obtain query results.

- Execute SQL statements

```
odps.execute_sql('select * from dual') # Execute SQL statements in synchronous mode. Threads are blocked until the SQL statements are executed.
instance = odps.run_sql('select * from dual') # Execute SQL statements in asynchronous mode.
instance.wait_for_success() # Threads are blocked until the SQL statements are executed.
```

- Obtain SQL query results

You can call the `open_reader` method to obtain SQL query results. In the following example, structured data is returned.

```
with odps.execute_sql('select * from dual').open_reader() as reader:
    for record in reader:
        # Process each record.
```

When you run the `DESC` command, you can use `reader.raw` to obtain the raw SQL query results.

```
with odps.execute_sql('desc dual').open_reader() as reader:
    print(reader.raw)
```

PyODPS Resource

Resources commonly apply to user-defined functions (UDFs) and MapReduce on MaxCompute.

You can call the `list_resources` method to query all resources, the `exist_resource` method to check whether a resource exists, and the `delete_resource` method to delete a resource. You can also call the `drop()` method to delete a resource.

PyODPS supports two resource types: files and tables.

- File resources

Files of basic file types and the .py, .jar, and .archive types are supported.

- Create a file resource

You can specify a resource name, a file type, and a file-like or string object to create a file resource. Example:

```
resource = odps.create_resource('test_file_resource', 'file', file_obj=open('/to/path/file')) # Create a file-like object.
resource = odps.create_resource('test_py_resource', 'py', file_obj='import this') # Create a string object.
```

- Read and modify a file resource

You can call the `open` method for a file resource or call the `open_resource` method at the MaxCompute entry to open a file resource. The opened object is a file-like object. Similar to the `open` method built in Python, file resources also support various opening modes. Example:

```
with resource.open('r') as fp: # Open the file in read mode.
    content = fp.read() # Read all content.
    fp.seek(0) # Return to the beginning of the file.
    lines = fp.readlines() # Read multiple lines.
    fp.write('Hello World') # Error. Data cannot be written to a file in read mode.
with odps.open_resource('test_file_resource', mode='r+') as fp: # Open the file in read/write mode.
    fp.read()
    fp.tell() # Current position.
    fp.seek(10)
    fp.truncate() # Truncate the file to the specified length.
    fp.writelines(['Hello\n', 'World\n']) # Write multiple lines into the file.
    fp.write('Hello World')
    fp.flush() # Manually call the method to submit the update to MaxCompute.
```

PyODPS supports the following opening modes:

- `r`: read mode. The file can be opened but data cannot be written to it.
- `w`: write mode. Data can be written to the file, but data in the file cannot be read. If a file is opened in write mode, the file content is deleted first.
- `a`: append mode. Data can be appended to the file.
- `r+`: read/write mode. You can read data from and write data to the file.
- `w+`: This mode is similar to `r+`, but the file content is deleted first.
- `a+`: This mode is similar to `r+`, but data can only be appended to the file.

In PyODPS, file resources can be opened in binary mode. For example, some compressed files must be opened in binary mode. `rb` indicates that a file is opened in binary read mode, and `r+b` indicates that a file is opened in binary read/write mode.

- Table resources

Create a table resource.

```
odps.create_resource('test_table_resource', 'table', table_name='my_table', partition='pt=test')
```

Update a table resource.

```
table_resource = odps.get_resource('test_table_resource')
table_resource.update(partition='pt=test2', project_name='my_project2')
```

DataFrame

PyODPS provides a pandas-like API called PyODPS DataFrame. This API makes full use of the computing power of MaxCompute. For more information, see [DataFrame](#).

The following DataFrame examples are provided for reference only.

 **Note** You must create a MaxCompute object before you perform the following steps.

```
odps = ODPS(**your-access-id**, **your-secret-access-key**,
            project=**your-project**, endpoint=**your-end-point**))
```

Assume that the following tables exist: pyodps_ml_100k_movies (movie-related data), pyodps_ml_100k_users (user-related data), and pyodps_ml_100k_ratings (rating-related data).

You need only to specify a table object to create a DataFrame object. Example:

```
from odps.df import DataFrame
```

```
users = DataFrame(o.get_table('pyodps_ml_100k_users'))
```

You can use the dtypes attribute to view the fields of DataFrame and the types of these fields. Example:

```
users.dtypes
```

You can call the `head` method to obtain the first N data records for data preview. Example:

```
users.head(10)
```

number	user_id	age	sex	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213
5	6	42	M	executive	98101
6	7	57	M	administrator	91344
7	8	36	M	administrator	05201

number	user_id	age	sex	occupation	zip_code
8	9	29	M	student	01002
9	10	53	M	lawyer	90703

You can add a filter on the fields to view specified fields. Example:

```
users[['user_id', 'age']].head(5)
```

number	user_id	age
0	1	24
1	2	53
2	3	23
3	4	24
4	5	33

You can also exclude several fields from results. Example:

```
users.exclude('zip_code', 'age').head(5)
```

number	user_id	sex	occupation
0	1	M	technician
1	2	F	other
2	3	M	writer
3	4	M	technician
4	5	F	other

If you exclude some fields, you may want to obtain new columns by using computation. For example, add the `sex_bool` attribute and set this attribute to `True` if `sex` is `M`. Otherwise, set this attribute to `False`. Example:

```
users.select(users.exclude('zip_code', 'sex'), sex_bool=users.sex == 'M').head(5)
```

number	user_id	age	occupation	sex_bool
0	1	24	technician	True
1	2	53	other	False

number	user_id	age	occupation	sex_bool
2	3	23	writer	True
3	4	24	technician	True
4	5	33	other	False

Obtain the number of users aged between 20 and 25. Example:

```
users.age.between(20, 25).count().rename('count')
943
```

Obtain the numbers of male and female users. Example:

```
users.groupby(users.sex).count()
```

number	sex	count
0	F	273
1	M	670

To divide users by occupation, obtain the top 10 occupations, and sort the occupations in descending order based on population. Example:

```
df = users.groupby('occupation').agg(count=users['occupation'].count())
df.sort(df['count'], ascending=False)[:10]
```

number	occupation	count
0	student	196
1	other	105
2	educator	95
3	administrator	79
4	engineer	67
5	programmer	66
6	librarian	51
7	writer	45
8	executive	32
9	scientist	31

The DataFrame API provides the `value_counts` method for the same purpose. Example:

```
users.occupation.value_counts()[:10]
```

number	occupation	count
0	student	196
1	other	105
2	educator	95
3	administrator	79
4	engineer	67
5	programmer	66
6	librarian	51
7	writer	45
8	executive	32
9	scientist	31

Show data in an intuitive graph. Example:

```
%matplotlib inline
```

Use a horizontal column chart to visualize data. Example:

```
users['occupation'].value_counts().plot(kind='barh', x='occupation',  
label='profession')
```

Divide users into 30 groups by age and view the histogram of age distribution. Example:

```
users.age.hist(bins=30, title="Distribution of users' ages", xlabel='age', ylabel='count of users')
```

Use JOIN to join three tables and save the joined tables as a new table. Example:

```
movies = DataFrame(o.get_table('pyodps_ml_100k_movies'))  
ratings = DataFrame(o.get_table('pyodps_ml_100k_ratings'))  
o.delete_table('pyodps_ml_100k_lens', if_exists=True)  
lens = movies.join(ratings).join(users).persist('pyodps_ml_100k_lens')  
lens.dtypes
```

```
odps.Schema {
  movie_id      int64
  title         string
  release_date  string
  video_release_date string
  imdb_url     string
  user_id      int64
  rating       int64
  unix_timestamp int64
  age          int64
  sex          string
  occupation   string
  zip_code     string
}
```

Divide users who are aged from 0 to 80 into eight age groups. Example:

```
labels = ['0-9', '10-19', '20-29', '30-39', '40-49', '50-59', '60-69', '70-79']
cut_lens = lens[lens, lens.age.cut(range(0, 81, 10), right=False, labels=labels).rename('age_group')]
```

View the first 10 data records of a single age in a group. Example:

```
cut_lens['age_group', 'age'].distinct()[:10]
```

number	age_group	age
0	0-9	7
1	10-19	10
2	10-19	11
3	10-19	13
4	10-19	14
5	10-19	15
6	10-19	16
7	10-19	17
8	10-19	18
9	10-19	19

View the total rating and average rating of users in each age group. Example:

```
cut_lens.groupby('age_group').agg(cut_lens.rating.count().rename('total_rating'), cut_lens.rating.mean().rename('avg_rating'))
```

number	age_group	average_rating	total_rating
0	0-9	3.767442	43
1	10-19	3.486126	8181
2	20-29	3.467333	39535
3	30-39	3.554444	25696
4	40-49	3.591772	15021
5	50-59	3.635800	8704
6	60-69	3.648875	2623
7	70-79	3.649746	197

Configurations

PyODPS provides a series of configuration options, which can be obtained by using `odps.options`. The following table describes configurable MaxCompute options.

- General configurations

Option	Description	Default value
<code>end_point</code>	MaxCompute Endpoint	None
<code>default_project</code>	The default project.	None
<code>log_view_host</code>	The hostname of Logview.	None
<code>log_view_hours</code>	The retention time of Logview. Unit: hours.	24
<code>local_timezone</code>	The time zone to be used. True indicates local time, and False indicates UTC. You can also use the pytz time zone.	1
<code>lifecycle</code>	The lifecycle of all tables.	None
<code>temp_lifecycle</code>	The lifecycle of temporary tables.	1
<code>biz_id</code>	The ID of a user.	None
<code>verbose</code>	Specifies whether to display logs.	False
<code>verbose_log</code>	The log receiver.	None
<code>chunk_size</code>	The size of the write buffer.	1496

Option	Description	Default value
retry_times	The number of request retries.	4
pool_connections	The number of cached connections in the connection pool.	10
pool_maxsize	The maximum capacity of the connection pool.	10
connect_timeout	The time to wait before the connection times out.	5
read_timeout	The time to wait before the read operation times out.	120
completion_size	The limit on the number of object completion listing items.	10
notebook_repr_widget	Specifies whether to use interactive graphs.	True
sql.settings	MaxCompute SQL runs global hints.	None
sql.use_odps2_extension	Specifies whether to enable MaxCompute V2.0 language extension.	False

- Data upload and download configurations

Option	Description	Default value
tunnel.endpoint	Tunnel Endpoint	None
tunnel.use_instance_tunnel	Specifies whether to use Instance Tunnel to obtain the upload or download results.	True
tunnel.limited_instance_tunnel	Specifies whether to limit the number of data records obtained by using Instance Tunnel.	True
tunnel.string_as_binary	Specifies whether to use bytes instead of unicode for data of the STRING type.	False

- DataFrame configurations

Option	Description	Default value

Option	Description	Default value
interactive	Specifies whether DataFrame is used in an interactive environment.	Based on the measured value
df.analyze	Specifies whether to enable non-MaxCompute built-in functions.	True
df.optimize	Specifies whether to enable full DataFrame optimization.	True
df.optimizes.pp	Specifies whether to enable DataFrame predicate push optimization.	True
df.optimizes.cp	Specifies whether to enable DataFrame column pruning optimization.	True
df.optimizes.tunnel	Specifies whether to enable DataFrame tunnel optimization.	True
df.quote	Specifies whether to use a double grave accent (``) to mark fields and table names in the backend of MaxCompute SQL.	True
df.libraries	The resource name of the third-party library that is used for the DataFrame operation.	None

- PyODPS ML configurations

Option	Description	Default value
ml.xflow_project	The default name of the XFlow project.	algo_public
ml.use_model_transfer	Specifies whether to use ModelTransfer to obtain the PMML model.	True
ml.model_volume	The volume name used when ModelTransfer is used.	pyodps_volume

2.DataWorks

2.1. Developer Guide

2.1.1. Overview

DataWorks is an important platform as a service (PaaS) of Alibaba Cloud. It offers all-around services, such as Data Integration, DataStudio, Data Map, Data Quality, and DataService Studio. In addition, it provides an end-to-end data development and management console to help enterprises mine and explore data value. DataWorks provides fully-hosted workflow services and uses a comprehensive development and management interface to help enterprises efficiently build data centers.

2.1.2. Terms

This topic describes the terms that are related to DataWorks.

Workspaces

Workspaces are basic units for managing nodes, members, roles, and permissions in DataWorks. The administrator of a workspace can add members to the workspace and assign the workspace administrator, developer, administration expert, deployment expert, security expert, or visitor role to each member. This way, workspace members with different roles can collaborate with each other.

Business flows

A business flow consists of a set of nodes with dependency between each other.

Nodes

Each type of node is used to perform a specific data operation. Examples:

- A synchronization node can be used to synchronize data from ApsaraDB RDS to MaxCompute.
- A MaxCompute SQL node can be used to convert data by executing SQL statements that are supported by MaxCompute.

Each node has input tables or datasets and generates one or more output tables or datasets.

Nodes are classified into node tasks, flow tasks, and inner nodes.

Node type	Description
Node task	A node task is used to perform a data operation. You can configure dependencies between a node task and flow tasks or other node tasks to form a directed acyclic graph (DAG).

Node type	Description
Flow task	A flow task contains a group of inner nodes that process a workflow. You can configure dependencies between a flow task and node tasks or other flow tasks to form a DAG. We recommend that you create less than 10 flow tasks.  Note • Inner nodes in a flow task cannot be the dependencies of node tasks or other flow tasks. • In DataWorks V2.0 and later, you can find the flow tasks that are created in DataWorks V1.0 • but cannot create flow tasks. Instead, you can create workflows to perform similar operations.
Inner node	An inner node is a node within a flow task. The features of an inner node are basically the same as those of a node task. You can create dependencies between nodes by dragging them. The inner node recurrence inherits the flow task recurrence and cannot be configured separately.

Create CDH Cluster

An instance is a snapshot of a node at a specific point in time. An instance is generated each time a node is run as scheduled by the scheduling system or is manually triggered. An instance contains information such as the time at which the node is run, the running status of the node, and operational logs.

 **Note** You can query the instance information on the [Cycle Instance](#) page in [Operation Center](#).

Assume that Node 1 is scheduled to run at 02:00 every day. The scheduling system automatically generates an instance of Node 1 at 23:30 every day based on the time that is defined for the auto triggered node. At 02:00 the next day, if the scheduling system verifies that all the ancestor instances are run, the system automatically runs the instance of Node 1.

Resources and functions

Resources and functions are concepts in MaxCompute. For more information.

The DataWorks console allows you to manage resources and functions. If resources or functions are uploaded by using other services such as MaxCompute, you cannot query them in DataWorks.

Output name

Within an Apsara Stack tenant account, each node has an output name that is used to connect to its descendant nodes.

When you configure dependencies for a node, you must use its output name instead of its node name or ID. After you configure the dependencies, the output name of the node serves as the input name of its descendant nodes.



Note The output name of the node must be unique within the tenant.

Metadata

Metadata is data that provides descriptions for other data. It can describe the attributes such as the name, size, and data type, or the structure including the field, type, and length, or relevant information such as the location, owner, output node, and access permission.

the DAG for a retroactive instance

After an auto triggered node is developed, committed, and deployed to the scheduling system, the system runs the node as scheduled. If you want to perform computing on historical data in a time period, you can generate retroactive data for the node. The generated retroactive instance is run based on the specified data timestamp.

2.1.3. Introduction

This topic describes the limits and details of DataWorks OpenAPI operations.

Call limit

In DataWorks, the API operations are classified into three levels to standardize API operation calls and ensure system stability. For more information, see OpenAPI list. The following table describes the call limits of API operations at different levels.

Level	Call limit
Level 1	QPS①≤50
Level 2	QPS①≤20
Level 3	QPS ①≤5, and the total number of operations per day ②≤10000

- 1: the total QPS for the API operations at the current level in each region for an Alibaba Cloud account and its RAM users
- 2: the total number of daily calls for the API operations at level 3 in each region for an Alibaba Cloud account and its RAM users

List of operations by function

The following tables list the API operations available for use in DataWorks. The API operations are related to tenants, metadata, DataStudio, Operation Center, Data Quality, and DataService Studio.

- Tenants

Operation	Descriptions	Level
ListCalcEngines	Queries compute engine instances.	Level 3
DeleteDataSource	Deletes a data source.	Level 2
ListDataSource	Queries data sources.	Level 3
UpdateDataSource	Updates a data store.	Level 2
ListResourceGroups	Queries resource groups.	Level 1
GetProject	Queries the information about a DataWorks workspace.	Level 1
AddProjectMemberToRole	Assigns a role to a user in a DataWorks workspace. Before you call this operation, you must add the user to the DataWorks workspace.	Level 2
CreateProjectMember	Adds a user to a DataWorks workspace.	Level 2
DeleteProjectMember	Removes a user from a DataWorks workspace.	Level 2
ListProjectRoles	Queries roles in a DataWorks workspace.	Level 1
RemoveProjectMemberFromRole	Removes a role from a user in a DataWorks workspace.	Level 2
ListProjectMembers	Queries members in a DataWorks workspace.	Level 1
ListProjects	Queries DataWorks workspaces of the tenant to which a user belongs by page.	Level 1

- **Metadata**

Operation	Descriptions	Level
GetMetaDBInfo	Queries the basic metadata information about a compute engine instance.	Level 1
GetMetaDBTableList	Queries the tables of a compute engine instance.	Level 1
CheckMetaTable	Checks whether a table exists.	Level 1

Operation	Descriptions	Level
CheckMetaPartition	Checks whether a partition exists.	Level 1
SearchMetaTables	Queries tables based on specific conditions.	Level 1
GetMetaTableBasicInfo	Queries the basic information about a metatable.	Level 1
GetMetaTableColumn	Queries the field information about a table.	Level 1
GetMetaTableFullInfo	Queries the complete information about a table, including the field information.	Level 1
GetMetaTablePartition	Queries the partitions of a table.	Level 1
GetMetaTableOutput	Queries the output information about a table.	Level 1
GetMetaTableChangeLog	Queries the change logs of a table.	Level 1
GetMetaTableIntroWiki	Queries the instructions on how to use a table.	Level 1
GetMetaTableLineage	Queries the lineage of a table.	Level 1
GetMetaColumnLineage	Queries the lineage of a field.	Level 3
CreateMetaCategory	Creates a category.	Level 2
DeleteMetaCategory	Deletes a category.	Level 2
UpdateMetaCategory	Updates a category.	Level 2
GetMetaCategory	Queries the category tree.	Level 1
GetMetaTableListByCategory	Queries tables in a specified category.	Level 1
AddToMetaCategory	Adds a table to a specified category.	Level 2
DeleteFromMetaCategory	Removes a table from a specified category.	Level 2
CreateTable	Creates a MaxCompute table.	Level 2
UpdateTable	Updates a MaxCompute table.	Level 2

Operation	Descriptions	Level
DeleteTable	Deletes a MaxCompute table.	Level 2
GetDDLJobStatus	Queries the status of a table creation, update, or deletion task.	Level 1
UpdateMetaTable	Updates the metadata information about a table.	Level 2
UpdateMetaTableIntroWiki	Updates the instructions on how to use a table. If no instruction on how to use the table is available, the instructions that are specified by calling this operation are added.	Level 2
ListMetaDB	Queries metadatabases.	Level 1
UpdateTableModelInfo	Updates the topic, level, and category of a table.	Level 2
CreateTableTheme	Creates a topic.	Level 2
DeleteTableTheme	Deletes a topic.	Level 2
UpdateTableTheme	Updates a topic.	Level 2
ListTableTheme	Queries topics.	Level 1
CreateTableLevel	Creates a table level.	Level 2
DeleteTableLevel	Deletes a table level.	Level 2
UpdateTableLevel	Updates a table level.	Level 2
ListTableLevel	Queries table levels.	Level 1

- **DataStudio**

Operation	Descriptions	Level
CreateFile	Creates a file in DataStudio.	Level 3
CreateUdfFile	Creates a file for a function in DataStudio.	Level 2
CreateBusiness	Creates a workflow.	Level 2
DeleteBusiness	Deletes a workflow.	Level 3
CreateFolder	Creates a folder.	Level 2

Operation	Descriptions	Level
UpdateBusiness	Updates a workflow.	Level 2
ListFiles	Queries files.	Level 1
EstablishRelationTableToBusiness	Imports a table to a workflow. This operation is equivalent to right-clicking a workflow on the DataStudio page and selecting Import Table.	Level 2
SubmitFile	Commits a file to the development environment of the scheduling system to generate a node.	Level 3
ListBusiness	Queries workflows.	Level 1
GetFileVersion	Queries the details of a file version.	Level 1
DeleteFolder	Deletes a folder from DataStudio.	Level 3
DeleteFile	Deletes a file from DataStudio.	Level 3
GetDeployment	Queries the details of a deployment task.	Level 1
UpdateFile	Updates a file.	Level 2
GetFolder	Queries the details of a folder.	Level 1
ListFileVersions	Queries file versions.	Level 1
GetFile	Queries the details of a file.	Level 1
ListFolders	Queries folders.	Level 1
DeployFile	Deploys a file to the production environment.	Level 3
UpdateUdfFile	Updates the file information about a function.	Level 2
GetBusiness	Queries the details of a workflow.	Level 1
UpdateFolder	Updates the information about a folder.	Level 2

- Operation Center

Operation	Descriptions	Level
ListAlertMessages	Queries alerts.	Level 1
GetBaselineConfig	Queries the configurations of a baseline.	Level 1
ListNodesByBaseline	Queries nodes in a baseline.	Level 1
ListBaselineConfigs	Queries baselines.	Level 1
GetBaselineKeyPath	Queries the key path of a baseline instance.	Level 1
GetBaselineStatus	Queries the details of a baseline instance.	Level 1
ListBaselineStatuses	Queries baseline instances.	Level 1
DeleteRemind	Deletes a custom alert rule.	Level 2
UpdateRemind	Updates a custom alert rule.	Level 3
CreateRemind	Creates a custom alert rule.	Level 3
GetRemind	Queries the details of a custom alert rule.	Level 1
ListReminds	Queries custom alert rules.	Level 1
ListTopics	Queries events.	Level 1
GetTopic	Queries the details of an event.	Level 1
GetTopicInfluence	Queries baseline instances affected by an event.	Level 1
GetNode	Queries the details of a node.	Level 1
ListNodeInputOrOutput	Queries the upstream or downstream information of a node.	Level 1
ListNodes	Queries nodes.	Level 1
GetNodeCode	Queries the code of a node.	Level 1
GetInstance	Queries the details of an instance.	Level 1
ListInstances	Queries instances.	Level 1
GetInstanceLog	Queries the logs of an instance.	Level 1

Operation	Descriptions	Level
StopInstance	Terminates an instance.	Level 2
RestartInstance	Restarts an instance.	Level 2
SetSuccessInstance	Resets the state of a failed instance to successful.	Level 2
SuspendInstance	Suspends an instance.	Level 2
ResumeInstance	Resumes a suspended instance.	Level 2
RunCycleDagNodes	Creates a workflow to generate retroactive data.	Level 2
RunSmokeTest	Creates a workflow to perform smoke testing.	Level 2
ListSuccessInstanceAmount	Queries auto triggered node instances that are generated on the current day and collects statistics about the trends in the number of instances that are run every hour on the hour.	Level 3
ListInstanceAmount	Queries the trends in the number of auto triggered node instances within a specified period of time.	Level 1
TopTenElapsedTimeInstance	Queries the rankings of the durations of instances.	Level 1
TopTenErrorTimesInstance	Queries the rankings of nodes on which errors occurred within the last month.	Level 1
GetFileTypeStatistic	Queries the distribution of node types.	Level 1
GetInstanceStatusCount	Queries the statistics of instances in different states.	Level 1
ListFileType	Queries the information about one or more node types, such as the code and name of a node type.	Level 1

Operation	Descriptions	Level
RunManualDagNodes	Triggers workflow executions. The manually triggered workflow must be committed and deployed. Then, you can find the manually triggered workflow on the Operation Center page and call this operation.	Level 2
ListManualDagInstances	Queries instances in a manually triggered workflow.	Level 1
GetDag	Queries the details of a directed acyclic graph (DAG) based on the DAG ID.	Level 1
ListNodesByOutput	Queries nodes based on the output of the nodes.	Level 1
RunTriggerNode	Runs a manually triggered node.	Level 2
GetNodeParents	Queries the ancestor nodes of a node.	Level 2
GetNodeChildren	Queries the descendant nodes of a node.	Level 2

- **Data Quality**

Operation	Descriptions	Level
UpdateQualityFollower	Updates a subscriber.	Level 2
UpdateQualityRule	Updates a monitoring rule.	Level 2
GetQualityRule	Queries the information about a monitoring rule.	Level 1
ListQualityRules	Queries monitoring rules based on a partition filter expression.	Level 3
CreateQualityRule	Creates a monitoring rule.	Level 3
DeleteQualityFollower	Deletes a subscriber for a partition filter expression.	Level 2
DeleteQualityRule	Deletes a monitoring rule.	Level 2
GetQualityFollower	Queries the subscribers for a partition filter expression.	Level 1

Operation	Descriptions	Level
CreateQualityFollower	Creates a subscriber for a partition filter expression.	Level 3
DeleteQualityEntity	Deletes a partition filter expression.	Level 2
GetQualityEntity	Queries a partition filter expression.	Level 1
CreateQualityEntity	Creates a partition filter expression in Data Quality.	Level 2
CreateQualityRelativeNode	Creates an associated node for a partition filter expression.	Level 3
DeleteQualityRelativeNode	Disassociates a node from a partition filter expression.	Level 2
ListQualityResultsByEntity	Queries the historical monitoring results based on a partition filter expression.	Level 1
ListQualityResultsByRule	Queries the monitoring results based on a monitoring rule.	Level 1

- **DataService Studio**

Operation	Descriptions	Level
CreateDataServiceApi	Creates an API in DataService Studio.	Level 3
PublishDataServiceApi	Publishes an API in DataService Studio.	Level 3
DeleteDataServiceApi	Deletes an API from DataService Studio.	Level 3
ListDataServiceApis	Queries APIs in the development state in DataService Studio.	Level 1
GetDataServiceApi	Queries the details of an API in the development state in DataService Studio.	Level 1
UpdateDataServiceApi	Updates an API in the development state in DataService Studio.	Level 3
GetDataServicePublishedApi	Queries the details of an API in the published state in DataService Studio.	Level 1

Operation	Descriptions	Level
ListDataServicePublishedApis	Queries the information about APIs in the published state in DataService Studio.	Level 1
ListDataServiceApiAuthorities	Queries the APIs whose access permissions are granted to users.	Level 1
ListDataServiceAuthorizedApis	Queries authorized APIs.	Level 1
AbolishDataServiceApi	Unpublishes an API in DataService Studio.	Level 3
GetDataServiceApplication	Queries the details of an application in DataService Studio.	Level 1
ListDataServiceApplications	Queries the basic information of applications in DataService Studio.	Level 1
CreateDataServiceFolder	Creates a folder in DataService Studio.	Level 2
GetDataServiceFolder	Queries the details of a folder in DataService Studio.	Level 1
ListDataServiceFolders	Queries folders in DataService Studio.	Level 1
CreateDataServiceGroup	Creates a business process in DataService Studio.	Level 2
GetDataServiceGroup	Queries a business process in DataService Studio.	Level 1
ListDataServiceGroups	Queries business processes in DataService Studio.	Level 1
CreateDataServiceApiAuthority	Grants the access permissions on an API in DataService Studio.	Level 3
DeleteDataServiceApiAuthority	Revokes the access permissions on an API in DataService Studio.	Level 3

2.1.4. List of operations by function

Operations of the scheduling system

- Create a node
- Update a node

- Update multiple nodes at a time
- Delete a node
- Query parent nodes
- Query child nodes
- Query the node code
- Query nodes
- Query a node by ID
- Query the total number of nodes
- Query node instances
- Query the operational logs of a node instance
- Query a node instance by ID
- Query the parent node at a specified level
- Query the child node instances at a specified level
- Query the number of node instances in each status of an application
- Query the number of task instances in each status on a specified date
- Query the number of node instances in each status for applications
- Rerun an instance
- Restore an instance
- Set the instance status to Success and trigger descendant instances
- Terminate a node
- Rerun a descendant instance
- Update the scheduling configurations of an instance
- Pause an instance
- Terminate multiple instances at a time
- Rerun multiple instances at a time
- Restore multiple nodes at a time
- Pause multiple instances at a time
- Set the status of multiple instances to Success
- Query the total number of instances on the current day
- Query workflow instances
- Create a retroactive data generation task
- Run a smoke test
- Run an ad-hoc workflow
- Terminate a workflow
- Submit a resource package
- View the submission status of the resource package

2.1.5. Make API requests

DataWorks allows you to make API requests by using the Java SDK or the HTTP method. The Java SDK is newly released. You can use only the Java SDK to call API operations in the later versions of DataWorks. The HTTP method remains unchanged. No new API operations are added. In later versions, these API operations will be discontinued. We recommend that you use the Java SDK.

2.1.5.1. Java SDK

If you want to initialize and use a client to make an API request by using DataWorks SDK for Java, you must specify the following parameters: popEndpoint, accessId, accessKey, popProduct, regionId, and popEndpointName.

popEndpoint

The popEndpoint parameter specifies the endpoint that is used to access the service. The endpoint consists of the prefix dataworks. followed by the domain name of DataWorks. For example, if the DataStudio URL for Apsara Stack is `http://ide.envxxx.yyyyy.com`, the endpoint is `dataworks.envxxx.yyyyy.com`. If the actual endpoint does not match the preceding rule, contact Alibaba Cloud technical support personnel.

accessId and accessKey

The accessId and accessKey parameters specify the AccessKey ID and AccessKey secret of the Apsara Stack tenant account.

popProduct

The name of the service. Set the value to `dataworks-private-cloud`.

regionId and popEndpointName

- The regionId parameter specifies the ID of the region where the service resides. Set the value to default.
- The popEndpointName parameter specifies the name of the endpoint. Set the value to default.

Authorization

You can use only an authorized Apsara Stack tenant account to make API requests by using DataWorks SDK for Java.

Each DataWorks API operation is configured with a list of authorized Apsara Stack tenant accounts. To authorize an Apsara Stack tenant account, add the account to the list. If the account that you use to make API requests is not in the list, the following error message appears:

```
com.aliyuncs.exceptions.ClientException: InvalidApi.NotFound : Specified api is not found, please check your url and method.
```

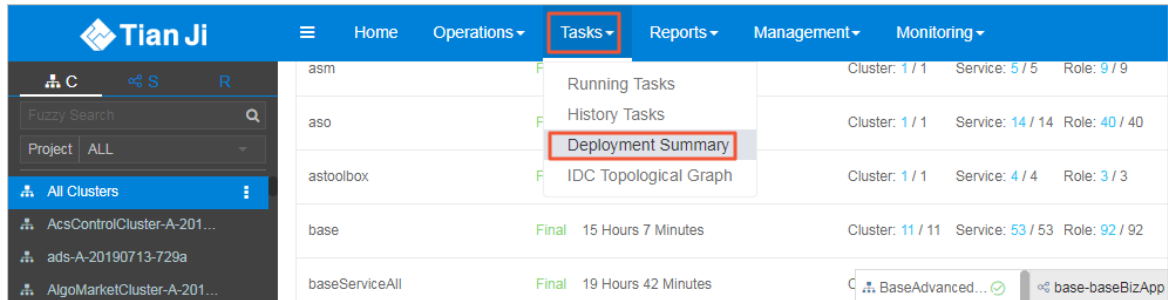
This error message indicates that the account is not authorized to call the operation. To resolve this issue, perform the following operations:

Contact the DataWorks technical support personnel and provide the Apsara Stack tenant account for which you want to obtain the required permissions and the API operation that you want to call. After you receive an encrypted license provided by the DataWorks technical support personnel based on the configuration in the runtime environment, use the license to authorize the account to make API requests.

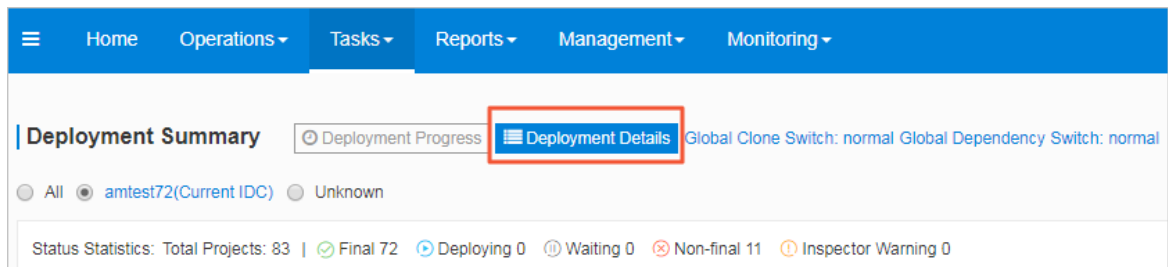
1. Log on to Apsara Infrastructure Management Framework.

For more information, see *Log on to Apsara Infrastructure Management Framework* in **Apsara Infrastructure Management Framework** under **Operations tools** in *Alibaba Cloud Apsara Stack Enterprise Operations and Maintenance Guide*.

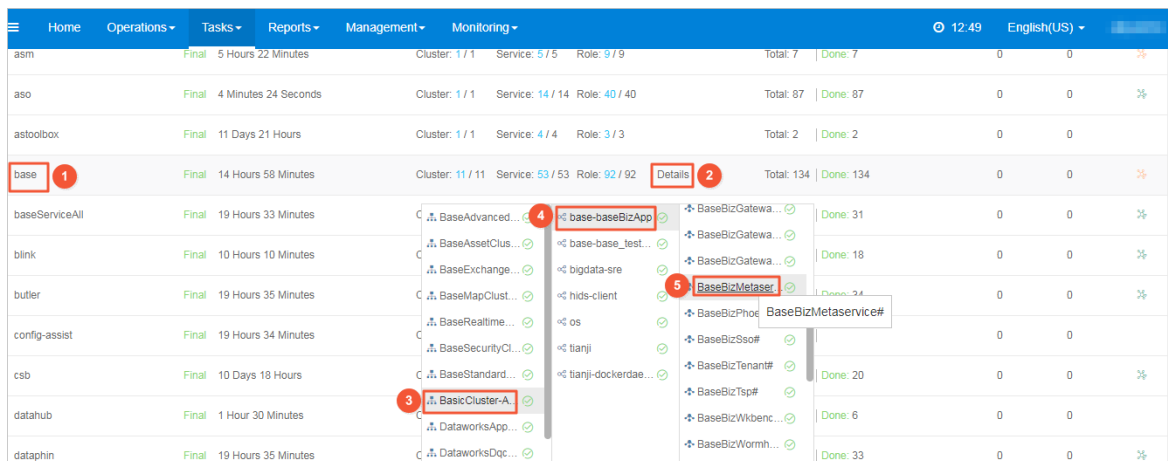
2. In the left-side navigation pane, click **Monitoring**.
3. In the top navigation bar, move the pointer over **Tasks** and select **Deployment Summary**.



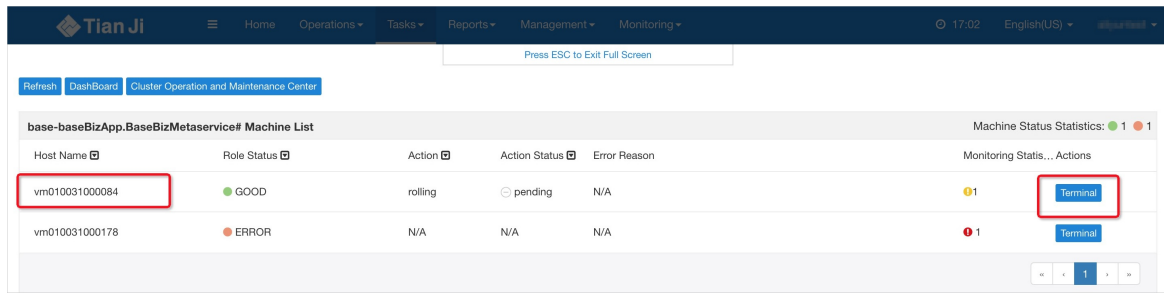
4. On the **Deployment Summary** page, click the **Deployment Details** tab.



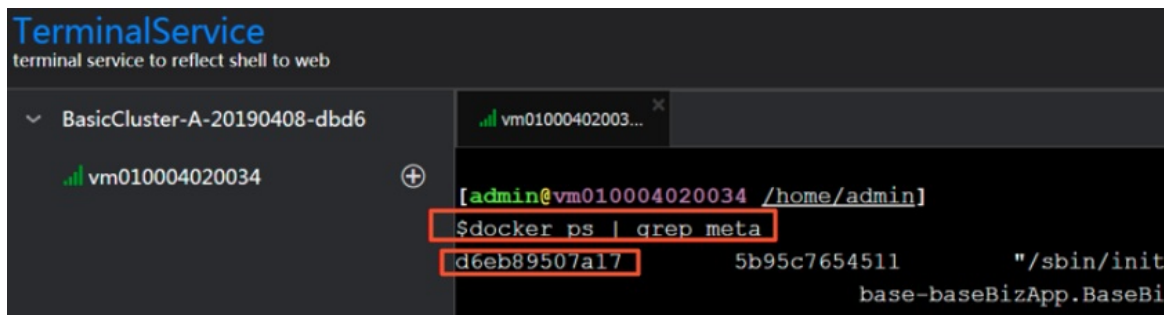
5. Move the pointer over the **base** service, click **Details**, and then choose **BasicCluster-A-20201119-daef > base-baseBizApp > BaseBizMetaservice**.



6. On the page that appears, find your desired host and click **Terminal** in the **Actions** column to go to the **TerminalService** page.



- On the TerminalService page, click the Add icon, and run the `docker ps | grep meta` command on the tab that appears to obtain the ID of the base-biz-metasevice container. For example, the following figure shows that the container ID is d6eb89507a17.



- Log on to the container and run the `docker exec -it d6eb89507a17 bash` command. Then, you can use the license for authorization.



Run the following command to authorize the account to make API requests:

```
curl -X POST -d "license=4AA481CE03EAF6529ECA131ED6E29AAD511B66A760C33786AD241BE1D90D7007C4056FA26FD13318B92BCFB4B254E42" "http://127.0.0.1/dataos/v1/conf_dataos_api";
```

In this command, the license is provided by the DataWorks technical support personnel. If `{"data":"ok","errCode":0,"errMsg":"success","requestId":"0a04142615555747936747847e031b"}` appears after you run the command, you can use the authorized account to make API requests.

Sample code

```
public void setup () throws IOException, ClientException {
    popEndpoint = "dataworks.envxxx.yyyyy.com";
    popEndpointName = "default";
    accessId = "accessId";
    accessKey = "accessKey";
    regionId = "default";
    popProduct = "dataworks-private-cloud";
    X509TrustAll.ignoreSSLCertificate();
    DefaultProfile.addEndpoint(popEndpointName, regionId, popProduct, popEndpoint);
    IClientProfile profile = DefaultProfile.getProfile(regionId, accessId, accessKey);
    IAcsClient client = new DefaultAcsClient(profile);
}
```

```
/** * Example: DemoApiRequest *
 * @throws ServerException *
 * @throws ClientException */
@Test
public void test_DemoApi() throws ServerException, ClientException {
    DemoApiRequest request = new DemoApiRequest();
    request.setKeyword("abc");
    DemoApiResponse response = client.getAcsResponse(request);
    System.out.println(gson.toJson(response));
}
```

Specify the account for calling API operations that involve the scheduling system

If you call an API operation that involves the scheduling system, you must specify the following three common request parameters: `accountId`, `accountPwd`, and `clientName`. The following procedure describes how to specify these three parameters:

1. You can ask a field engineer to log on to the database that hosts the base-biz-tsp service. The name of the database is `dwtsp`.
2. The field engineer inserts a record into the `cloudapi_account` table to specify the `accountId`, `accountPwd`, and `clientName` parameters. These parameters uniquely identify the account that is used for calling the API operation.

2.1.5.2. HTTP API

 **Warning** Alibaba Cloud has stopped maintaining the DataWorks HTTP API. To be compatible with earlier versions, the HTTP method can still be used, but will be discontinued in later versions. Change to use the Java SDK as soon as possible.

2.1.5.2.1. Endpoints

The DataWorks API operations are categorized based on functions. Each type of API operations is accessed by using a specific domain name. For more information about the domain names, see the description about each type of API operations.

2.1.5.2.2. Protocol

All APIs use the HTTP protocol in Apsara Stack.

2.1.5.2.3. Request method

You can use the HTTP GET or POST request method to fulfill your intentions based on the description in the related API reference.

- If you use the GET method, an API operation returns the result based on the parameters in the query string.
- If you use the POST method, an API operation returns the result based on the parameters in the query string. The parameters in the request body are ignored.

2.1.5.2.4. Encoding

UTF-8 encoding is used in all the DataWorks APIs.

2.1.5.2.5. Signature method

Header

(Required) base_id

(Required) tenant_id

Signature

You must sign all API requests so that your identity can be verified. To sign an API request, follow these steps:

1. Add the following two parameters to the request URL: baseKey and timestamp.
2. Calculate a signature and add it to the HTTP header.

- Sample GET request

In this example, User B provides an HTTP GET API: /getapi. The name and age parameters are query parameters in the request URL. User A needs to use a URL that includes the following part to make the API request: /getapi?name=fengyeng&age=20. After User A obtains a token from the token center with the authentication passed such as AppCode, the baseKey and timestamp parameters are added to the request URL as query parameters. You can specify the authentication method to allow others to access the API operation. Therefore, the new URL includes the following part: /getapi?name=fengyeng&age=20?baseKey=dsa134&timestamp=123456789. A signature is then generated based on the token and query parameters in the new URL, and added to the request header by the requester as a field, for example, signature:dsafadfsa141fdsaf1.

- Sample POST request

In this example, User B provides an HTTP POST API: /postapi. User A needs to use a URL that includes the following part to make the API request: /postapi. After User A obtains a token from the token center, the baseKey and timestamp parameters are added to the request URL as query parameters. Therefore, the new URL includes the following part: /postapi?baseKey=dsa134&timestamp=123456789. A signature is then generated based on the token and query parameters in the new URL, and added to the request header by the requester as a field, for example, signature:dsafadfsa141fdsaf1.

2.1.5.2.6. Response parameters

```
{
  "requestId": <STRING>, # The request ID, which is used to locate the log entries for troubleshooting.
  "returnCode": "0", # A value of 0 indicates that the API operation is successful.
  "returnMessage": <STRING>, # The operation details.
  "successCode": "0",
  "count": <INTEGER>,
  "returnValue": <The returned result>
}
```

2.1.5.2.7. API example

Sample code

```
import org.apache.commons.lang3.StringUtils; import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient; import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpUriRequest; import org.apache.http.impl.client.HttpClientBuilder;
import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader; import java.net.URLDecoder;
import java.security.MessageDigest; import java.util.Arrays;
public class SearchNodes {
  public static void main(String[] args) throws Exception {
    HttpClient httpClient = HttpClientBuilder.create().build();
    String uri = "http://baseapi.***.com/v1.0/node/prod?executeMethod=SEARCH&searchText=nodeTest";
    HttpUriRequest request = createHttpRequest(
      "base key",
      "base token", uri
    );
    request.setHeader("tenant_id", "tenant ID"); request.setHeader("base_id", "Apsara Stack tenant account ID");
    HttpResponse response = httpClient.execute(request); BufferedReader bufferedReader = new BufferedReader(
      new InputStreamReader(response.getEntity().getContent())); StringBuilder stringBuilder = new StringBuilder();
    while(bufferedReader.ready()) {
      String line = bufferedReader.readLine(); if (line == null) {
        break;
      }
      stringBuilder.append(line);
    }
    System.out.println(stringBuilder.toString());
  }
  private static String byteArrayToHexString(byte[] byteArray) {
```



```
private static String byteArrayToHexString(byte[] byteArray) {
    final char[] HEX_DIGITS = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
    String result = null;
    int l = byteArray.length; char[] str = new char[l << 1]; int k = 0;
    for (int i = 0; i < l; i++) {
        byte byte0 = byteArray[i];
        str[k++] = HEX_DIGITS[byte0 >>> 4 & 0xf];
        str[k++] = HEX_DIGITS[byte0 & 0xf];
    }
    result = new String(str); return result;
}

public static String tokenSign(String param, String salt) throws Exception {
    if (StringUtils.isEmpty(param) || StringUtils.isEmpty(salt)) {
        return null;
    }
    String query = URLDecoder.decode(param, "UTF-8");
    String[] pStr = query.trim().split("&");
    Arrays.sort(pStr);
    StringBuilder buf = new StringBuilder(); for (int i = 0; i < pStr.length; ++i) {
        buf.append(pStr[i]);
        buf.append("&");
    }
    String paramStr = buf.substring(0, buf.length() - 1);
    String sourceStr = salt + ":" + paramStr + ":" + salt;
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] byteArray = md.digest(sourceStr.getBytes("UTF-8"));
    return byteArrayToHexString(byteArray);
}

protected static HttpRequest createHttpRequest(String baseKey, String baseToken, String uri) {
    StringBuilder buf;
    int index = uri.indexOf('?');
    if (index == -1) {
        index = uri.length();
        buf = new StringBuilder(uri.length() + baseKey.length() + 13);
        buf.append(uri).append('?');
    } else {
        buf = new StringBuilder(uri.length() + baseKey.length() + 13);
        buf.append(uri).append('&');
    }
    buf.append("baseKey=").append(baseKey);
    buf.append("&timestamp=").append(System.currentTimeMillis());
    HttpRequest request = new HttpGet(buf.toString()); String signature;
    try {
        signature = tokenSign(buf.substring(index + 1), baseToken);
    } catch (Exception ex) {
        throw new RuntimeException("error on sign for uri: " + uri, ex);
    }
    request.addHeader("signature", signature);
    return request;
}
```

2.1.6. Java SDK

2.1.6.1. Tenants

2.1.6.1.1. ListCalcEngines

Queries compute engine instances.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListCalcEngines	The operation that you want to perform.
CalcEngineType	String	Yes	ODPS	The type of the compute engine instance. Valid values: ODPS , EMR , BLINK , HOLO , MaxGraph , HYBRIDDB FOR_POSTGRESOL , ADB_MYSQL , and HADOOP_CDH . The values are not case-sensitive.
Name	String	Yes	abc	The name of the compute engine instance, which must be exactly matched.
ProjectId	Long	Yes	27	The DataWorks workspace to which the compute engine instance is bound.
RegionId	String	Yes	cn-shanghai	The region ID of the compute engine instance. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
EnvType	String	No	PRD	The environment in which the compute engine instance runs. Valid values: DEV and PRD. The value DEV indicates the development environment. The value PRD indicates the production environment.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 20.

Parameter	Type	Required	Example	Description
PageNumber	Integer	No	1	The number of the page to return. Valid values: a positive integer greater than or equal to 1. Default value: 1.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The query results for compute engine instances that are returned on multiple pages.
CalcEngines	Array of CalcEngines		The list of compute engine instances.
BindingProjectId	Integer	27	The ID of the workspace to which the compute engine instance is bound.
BindingProjectName	String	abc	The name of the workspace to which the compute engine instance is bound.
CalcEngineType	String	ODPS	The type of the compute engine instance. Valid values: ODPS , EMR , BLINK , HOLO , MaxGraph , HYBRIDDB FOR_POSTGRESOL , ADB_MYSQL , and HADOOP_CDH
DwRegion	String	cn-zhangjiakou	The region of the DataWorks workspace to which the compute engine instance is bound.
EngineId	Integer	35	The ID of the compute engine instance.
			The details of the compute engine instance. ODPS

Parameter	Type	Example	Description
			<pre>"pubEndpoint": "service.cn.maxcompute.aliyun.com/api", "endpoint": "service.cn.maxcompute.aliyun-inc.com/api", "resourceGroupType": "ODPS", "resourceGroupId": "361826516****", "vpcEndpoint": "service.cn.maxcompute.aliyun-inc.com/api", "projectName": "onefall_test_zjk", "taskSameAsOwner": "true" }</pre> - EMR <pre>{ "emrClusterId": "C-xxx", "specs": "{\"emrClusterId\":\"C-xxx\", \"emrAccessMode\":\"simple\", \"emrResourceQueueName\":\"default\", \"emrProjectId\":\"FP-xxx\"}", "endpoint": "emr.aliyuncs.com", "emrResourceQueueName": "default", "emrAccessMode": "simple", "resourceGroupType": "DW", "projectName": "xx-xxxx", "emrProjectId": "FP-xxxx", "taskSameAsOwner": "false" }</pre> - BLINK

Parameter	Type	Example	Description
			<pre>"bayesProjectId": "xxxx", "bayesProjectName": "xc_blxxixxnk_1", "cluster": "xxssxsx", "endpoint": "https://stream.console.aliyun.com", "engineType": "BLINK", "name": "xsxsxxxx", "projectName": "xc_blxxsxink_1", "queue": "root.xc_blxxxxxxink_1", "resourceGroupType": "DW", "specs": " {"cluster\\":\\"xxxxx\\",\\"bayesProjectName\\":\\"xc_blxxixxnk_1\\",\\"bayesProjectId\\":\\"ssxxsa\\",\\"name\\":\\"xsxsxxx\\",\\"queue\\":\\"root.sxxxsxs\\"}", "taskSameAsOwner": false }</pre> • HOLO

Parameter	Type	Example	Description
EngineInfo	Map	<pre>{ "pubEndpoint": "http://service.cn.maxcompute.aliyun.com/api", "endpoint": "http://service.cn.maxcompute.aliyun-inc.com/api", "resourceGroupType": "ODPS", "resourceGroupId": "361826516****", "vpcEndpoint": "http://service.cn.maxcompute.aliyun-inc.com/api", "projectName": "onefall_test_zjk", "taskSameAsOwner": "true" }</pre>	<pre>{ "endpoint": "hgprecn-cn-xsxsxsxs-cn-shanghai-internal.hologres.aliyuncs.com:80", "engineType": "ODPS", "odpsEndpoint": "hgprecn-cn-xsxsxsxs-cn-shanghai-internal.hologres.aliyuncs.com:80", "odpsProjectName": "xsxsxsxsxs", "projectName": "xsxsxsxsxs", "resourceGroupType": "DW", "specs": "{\n \"pubEndpoint\": \"hgprecn-cn-xsxsxsxs-cn-shanghai.hologres.aliyuncs.com:80\", \"commonBuyInstanceId\": \"hgprecn-cn-xsxsxsxs\", \"project\": \"holo_upxsxgrade1\", \"common_buy_instance_id\": \"hgprecn-cn-xsxsxs\", \"endpoint\": \"hgprecn-cn-xsxsxs-cn-shanghai-internal.hologres.aliyuncs.com:80\", \"port\": \"80\", \"host\": \"hgprecn-cn-xsxsxsxs-cn-shanghai-internal.hologres.aliyuncs.com\", \"vpcEndpoint\": \"hgprecn-cn-xsxsxsxs-cn-shanghai-vpc.hologres.aliyuncs.com:80\", \"authType\": 2, \"region\": \"cn-shanghai\" }", "taskSameAsOwner": false }</pre> - MaxGraph <pre>{ "endpoint": "http://pre-graphcompute.aliyuncs.com", "engineType": "ODPS", "odpsEndpoint": "http://pre-graphcompute.aliyuncs.com", "odpsProjectName": "xsxsxsxsxs", "projectName": "xsxsxsxsxs", "resourceGroupType": "DW", "taskSameAsOwner": false }</pre> - HYBRIDDB_FOR_POSTGRES

Parameter	Type	Example	Description
			<pre> "endpoint": "hybriddb_for_postgresql_mo12121ck_ endpoint", "engineType": "ODPS", "odpsEndpoint": "hybriddb_for_postgresql_m121212ock_ endpoint", "odpsProjectName": "sxasaxsaxaxas", "projectName": "sxasaxsaxaxas", "resourceGroupType": "DW", "specs": "{\"connectionString\":\"gp- xsxsxsxs.gpdb.rds.aliyuncs.com\", \"dat abase\":\"xsxsxsxs\", \"password\":\"xxx xxx\", \"instanceId\":\"gp- cdcdacdacda\", \"port\":\"3432\", \"owne rId\":\"12121212\", \"username\":\"sdasa ddsa\"}", "taskSameAsOwner": false } </pre> - ADB_MYSQL <pre> { "endpoint": "adb_mysql_mock_endpoint", "engineType": "ODPS", "odpsEndpoint": "adb_mysql_mock_endpoint", "odpsProjectName": "am-xsaxaxa", "projectName": "am-xsxsaxa", "resourceGroupType": "DW", "specs": "{\"connectionString\":\"am- xsaxsa.ads.aliyuncs.com:3306\", \"databa se\":\"xsaxsaxa\", \"password\":\"xsaxsax assxsa\", \"instanceId\":\"am- xsaxsasx\", \"username\":\"xsaxsadsd\"}" , "taskSameAsOwner": false } </pre> - HADOOP_CDH

Parameter	Type	Example	Description
			<pre> "bindingBaseId": "xsaxsaxs", "endpoint": "xsaaaaa", "engineType": "ODPS", "odpsEndpoint": "axsaxssxs", "odpsProjectName": "ssxxax", "projectName": "xsaxsaxsa", "resourceGroupId": 45208xxxxx, "resourceGroupType": "GATEWAY", "specs": "{ \"cluster\": { \"hive\": { \"hiveServer2Url\": \"jdbc:hive2://xxxxxer-1-cn-shanghai-pre-kerberos-1:10000\", \"hiveMetastore\": \"thrift://xxxxxr-1-cn-shanghai-pre-kerberos-1:9083\", \"version\": \"2.1.1\" }, \"configFiles\": { \"coreSite\": \"4534574xxxxx\", \"hdfsSite\": \"453457919xxxxx\", \"mapredSite\": \"45345750xxxxx\", \"yarnSite\": \"4534575xxxxx\", \"krb5Conf\": \"4534576xxxx1\", \"hiveSite\": \"453457xxxx20\" }, \"spark\": { \"version\": \"2.4.0\" }, \"cdh\": { \"version\": \"6.3.2\" }, \"hdfs\": { \"version\": \"3.0.0\" }, \"impala\": { \"impalaUrl\": \"jdbc:impala://cdh-xsxsxsxs-1-cn-shanghai-pre-kerberos-1:21050\", \"version\": \"3.2.0\" }, \"yarn\": { \"YarnUrl\": \"http://cdh-xsxsxsxs-1-cn-shanghai-pre-kerberos-1:8032\", \"webUrl\": \"http://cdh-xsxsxsxsxs-1-cn-shanghai-pre-kerberos-1:8088\", \"version\": \"3.0.0\" }, \"presto\": { \"prestoUrl\": \"jdbc:presto://cdh-xssxsxsxs-1-cn-shanghai-pre-kerberos-1:8080/hive/default\", \"version\": \"0.244.1\" } }, \"instanceId\": 161sdads733, \"authDetail\": { \"principal\": \"hive@HADOOP.COM\", \"keytabFileId\": \"45345815xsxsxs3\", \"type\": \"kerberos\", \"username\": \"xsxsxsxa@HADOOP.COM\", \"resGroupStatus\": \"\", \"hadoopAuthType\": \"kerberos\", \"clusterIdentifier\": \"xssxsxsxs\", \"clusterId\": \"xsxsx\", \"resGroupId\": 4520870619xsxsxs, \"accessMode\": \"security\", \"authType\": 2 } }, \"taskSameAsOwner\": false } </pre>

Parameter	Type	Example	Description
EnvType	String	PRD	The environment in which the compute engine instance runs. Valid values: - DEV: development environment - PRD: production environment
GmtCreate	String	Oct 10, 2019 3:42:44 PM	The time when the compute engine instance was created. It is a timestamp.
IsDefault	Boolean	true	Indicates whether the compute engine instance is the default instance of the same type.
Name	String	abc	The display name of the compute engine instance.
Region	String	cn-zhangjiakou	The region where the compute engine instance resides.
TaskAuthType	String	PROJECT	The identity that is assigned to access the compute engine instance. Valid values: - USER: current user - PROJECT: project executor - SUBACCOUNT: sub-account - STS_ROLE: STS role
TenantId	Long	1234567	The ID of the tenant.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	20	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	1	The total number of compute engine instances.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376****	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request to query compute engine instances is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListCalcEngines
&CalcEngineType=ODPS
&Name=abc
&ProjectId=27
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<data>
  <calcEngines>
    <bindingProjectId>27</bindingProjectId>
    <bindingProjectName>abc</bindingProjectName>
    <calcEngineType>ODPS</calcEngineType>
    <dwRegion>cn-zhangjiakou</dwRegion>
    <engineId>35</engineId>
    <engineInfo>
      <pubEndpoint>http://service.cn.maxcompute.aliyun.com/api</pubEndpoint>
      <endpoint>http://service.cn.maxcompute.aliyun-inc.com/api</endpoint>
      <resourceGroupType>ODPS</resourceGroupType>
      <resourceGroupId>361826516****</resourceGroupId>
      <vpcEndpoint>http://service.cn.maxcompute.aliyun-inc.com/api</vpcEndpoint>
      <projectName>abc</projectName>
      <taskSameAsOwner>true</taskSameAsOwner>
    </engineInfo>
    <envType>PROD</envType>
    <gmtCreate>Oct 10, 2019 3:42:44 PM</gmtCreate>
    <isDefault>true</isDefault>
    <name>abc</name>
    <region>cn-zhangjiakou</region>
    <taskAuthType>PROJECT</taskAuthType>
    <tenantId>1234567</tenantId>
  </calcEngines>
  <pageNumber>1</pageNumber>
  <pageSize>20</pageSize>
  <totalCount>1</totalCount>
</data>
<httpStatusCode>200</httpStatusCode>
<requestId>0bc1ec92159376****</requestId>
<success>true</success>
```

JSON format

```
{
  "data": {
    "calcEngines": [
      {
        "bindingProjectId": 27,
        "bindingProjectName": "abc",
        "calcEngineType": "ODPS",
        "dwRegion": "cn-zhangjiakou",
        "engineId": 35,
        "engineInfo": {
          "pubEndpoint": "http://service.cn.maxcompute.aliyun.com/api",
          "endpoint": "http://service.cn.maxcompute.aliyun-inc.com/api",
          "resourceGroupType": "ODPS",
          "resourceGroupId": "361826516****",
          "vpcEndpoint": "http://service.cn.maxcompute.aliyun-inc.com/api",
          "projectName": "abc",
          "taskSameAsOwner": "true"
        },
        "envType": "PROD",
        "gmtCreate": "Oct 10, 2019 3:42:44 PM",
        "isDefault": true,
        "name": "abc",
        "region": "cn-zhangjiakou",
        "taskAuthType": "PROJECT",
        "tenantId": 1234567
      }
    ],
    "pageNumber": 1,
    "pageSize": 20,
    "totalCount": 1
  },
  "httpStatusCode": 200,
  "requestId": "0bc1ec92159376****",
  "success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.

HttpCode	Error code	Error message	Description
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.2. CreateDataSource

Adds a data source in DataWorks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDataSource	The operation that you want to perform.
				The details of the data source. Example: - odps <pre>{ "accessId": "xssssss", "accessKey": "xsaxsaxsa", "authType": 2, "endpoint": "http://service.odps.aliyun.com/api", "project": "xsaxsax", "tag": "public" }</pre> - mysql

Parameter	Type	Required	Example	Description
				<pre>{ "database": "xsaxsa", "instanceName": "rm-xsaxsa", "password": "xsaxsa", "rdsOwnerId": "xsaxsa", "regionId": "cn-shanghai", "tag": "rds", "username": "xsaxsa" }</pre> - rds <pre>{ "configType": 1, "tag": "rds", "database": "xsaxsa", "username": "xsaxsa", "password": "xssaxsa\$32050", "instanceName": "rm-xsaxs", "rdsOwnerId": "11111111" }</pre> - oss <pre>{ "accessId": "sssssx", "accessKey": "xsaxxsaxs", "bucket": "xsa-xs-xs", "endpoint": "http://oss-cn-shanghai.aliyuncs.com", "tag": "public" }</pre> - sqlserver <pre>{ "jdbcUrl": "jdbc:sqlserver://xsaxsa- xsaxsa.database.xxx.cn:123;DatabaseName=xsxs-xsxs", "password": "sdasda\$fs", "tag": "public", "username": "sxaxacdacdd" }</pre> - polardb

Parameter	Type	Required	Example	Description
Content	String	Yes	<pre>{ "database": "dbname", "instanceName": "instancename", "password": "password", "rdsOwnerId": "123", "username": "username" }</pre>	<pre>{ "clusterId": "pc-sdadsadsa", "database": "dsadsadsa", "ownerId": "121212122", "password": "sdasdafssa", "region": "cn-shanghai", "tag": "polardb", "username": "asdadsads" }</pre> - oracle <pre>{ "jdbcUrl": "jdbc:oracle:saaa:@xxxxx:1521:PROD", "password": "sxasaxsa", "tag": "public", "username": "sasfadfa" }</pre> - mongodb <pre>{ "address": "[\"xsaxsa.mongodb.rds.aliyuncs.com:3717\"]", "database": "admin", "password": "sadsda@", "tag": "public", "username": "dsadsadas" }</pre> - emr

Parameter	Type	Required	Example	Description
				<pre>{ "accessId": "xsaxsa", "emrClusterId": "C-dsads", "emrResourceQueueName": "default", "emrEndpoint": "emr.aliyuncs.com", "accessKey": "dsadsad", "emrUserId": "224833315798889783", "name": "sasdsadsa", "emrAccessMode": "simple", "region": "cn-shanghai", "authType": "2", "emrProjectId": "FP-sdadsad" }</pre>
				• postgresql <pre>{ "jdbcUrl": "jdbc:postgresql://xxx:1921/sss", "password": "sdadsads", "tag": "public", "username": "sdsasda" }</pre> • analyticdb_for_mysql <pre>{ "instanceId": "am-sadsada", "database": "xsxsx", "username": "xsxsa", "password": "asdadsa", "connectionString": "am- xsxsxs.ads.aliyuncs.com:3306" }</pre> • hybriddb_for_postgresql

Parameter	Type	Required	Example	Description
				<pre>"connectionString": "gp-xsaxsaxa-master.gpdbmaster.rds.aliyuncs.com", "database": "xsaxsaxas", "password": "xsaxsaxsa@11", "instanceId": "gp-xsaxsaxsa", "port": "541132", "ownerId": "xsaxsaxsas", "username": "sadsad" }</pre> • holo <pre>{ "accessId": "xsaxsaxs", "accessKey": "xsaxsaxsa", "database": "xsaxsaxsa", "instanceId": "xsaxa", "tag": "aliyun" }</pre>
DataSourceType	String	Yes	rds	• kafka The type of the data source. Valid values: • "instanceId": "xsax-cn-xsaxsa", • "regionId": "cn-shanghai", • "tag": "aliyun", • "ownerId": "121212121212" • oss • sqlserver • polardb • oracle • mongoddb • emr • postgresql • analyticdb_for_mysql • hybriddb_for_postgresql • holo
EnvType	Integer	No	1	The environment in which the data source is used. Valid values: 0 and 1. The value 0 indicates the development environment. The value 1 indicates the production environment.

Parameter	Type	Required	Example	Description
Name	String	Yes	abc	The name of the data source.
ProjectId	Long	Yes	1	The ID of the DataWorks workspace to which the data source belongs. You can call the ListProjects operation to obtain the ID.
RegionId	String	Yes	cn-shanghai	The region ID of the data source. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
Description	String	No	abc	The description of the data source.
SubType	String	No	mysql	The subtype of the data source. This parameter takes effect only when the DataSourceType parameter is set to rds. When the DataSourceType parameter is set to rds, this parameter can be set to mysql, sqlserver, or postgresql.

Response parameters

Parameter	Type	Example	Description
Data	Long	123	The ID of the connection.
HttpStatusCode	String	200	The HTTP status code returned.
RequestId	String	0bc141151593763**	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateDataSource
&Content={"database":"dbname","instanceName":"instancename","password":"password","rdsOwnerId":
"123","username":"username"}
&DataSourceType=rds
&EnvType=1
&Name=abc
&ProjectId=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc141151593763****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>123</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc141151593763****",
  "HttpStatusCode": 200,
  "Data": 123,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.

HttpCode	Error code	Error message	Description
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.
400	ConnectionAlreadyExist	The connection with envType %s and name %s already exists.	The error message returned because the connection string already exists.
400	DataSource.CheckRamRoleFailure	Failed to check the RAM role. AccountId %s. roleArn %s.	The error message returned because the permissions on the data source have failed to be verified.
400	DataSource.ConnectionStrProperty.PatternNotMatch	The pattern value %s is invalid. The correct pattern is %s.	The error message returned because the connection string property of the data source does not match the specified rules.
400	Invalid.DataSource.DataSourceTypeNotSupport	The dataSourceType %s and subType %s are not supported.	The error message returned because the data source type is not supported.
400	Invalid.Param.Odps.EngineIdOrConnectStrEmpty	Either the compute engine ID or the connection string must be specified when data source type is ODPS.	The error message returned because the compute engine ID and connection string of the MaxCompute data source cannot be empty at the same time.
400	MissingParam.ConnectionStr.Property	You must specify property %s required by the data source connection string.	The error message returned because the required properties of the connection string of the data source are not specified.
400	Invalid.DataSource.DataSourceTypeNotExist	The data source type %s does not exist.	The error message returned because the specified data resource type does not exist.

2.1.6.1.3. ListDataSources

Queries the data sources of DataWorks.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataSources	The operation that you want to perform.

Parameter	Type	Required	Example	Description
Name	String	Yes	test	The name of the data source that you want to query.
ProjectId	Long	Yes	1	The ID of the DataWorks workspace to which the data source belongs. You can call the ListProjects operation to obtain the ID.
RegionId	String	Yes	cn-shanghai	The region ID of the data source. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
DataSourceType	String	No	rds	The type of the data source. Valid values: - odps - mysql - rds - oss - sqlserver - polardb - oracle - mongodb - emr - postgresql - analyticdb_for_mysql - hybridb_for_postgresql - holo
SubType	String	No	mysql	The subtype of the data source. This parameter takes effect only when the DataSourceType parameter is set to rds. If the DataSourceType parameter is set to rds, the parameter can be set to mysql, sqlserver, or postgresql.

Parameter	Type	Required	Example	Description
Status	String	No	ENABLED	The status of the data source. Valid values: - ENABLED: The data source is accessible. - DISABLED: The data source is inaccessible.
EnvType	Integer	No	1	The environment in which the data source runs. Valid values: 0 and 1. The value 0 indicates the development environment. The value 1 indicates the production environment.
PageSize	Integer	No	10	The number of the page to return. Default value: 10. Maximum value: 100.
PageNumber	Integer	No	1	The number of entries to return on each page. The value of this parameter must be a positive integer greater than or equal to 1.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The query result returned.
DataSources	Array of DataSources		The data sources.
BindingCalcEngineId	Integer	123	The ID of the compute engine with which the data source is associated.
ConnectStatus	Integer	1	The status of the data source. Valid values: 1: The data source is accessible. 2: The data source is inaccessible.

Parameter	Type	Example	Description
DataSourceType	String	rds	The type of the data source. Valid values: - odps - mysql - rds - oss - sqlserver - polardb - oracle - mongodb - emr - postgresql - analyticdb_for_mysql - hybriddb_for_postgresql - holo
DefaultEngine	Boolean	false	Indicates whether the engine that is associated with the data source is the default engine used by data sources of the same type.
Description	String	a connection	The description of the data source.
EnvType	Integer	1	The environment in which the data source runs. Valid values: 0: development environment. 1: production environment.
GmtCreate	String	Mar 17, 2021 4:09:32 PM	The time when the data source was created. Example: Mar 17, 2021 4:09:32 PM.
GmtModified	String	Mar 17, 2021 4:09:32 PM	The time when the data source was last modified. Example: Mar 17, 2021 4:09:32 PM.
Id	Integer	1	The ID of the connection.
Name	String	abc	The name of the data source.
Operator	String	193543050****	The ID of the Alibaba Cloud account that last modifies the data source.

Parameter	Type	Example	Description
ProjectId	Integer	123	The ID of the workspace with which the data source is associated.
Sequence	Integer	300	The sequence number of the data source. This parameter is used to sort data sources in descending order.
Shared	Boolean	false	Indicates whether the data source is shared.
Status	Integer	1	The status of the data source. Valid values: 1: The data source is accessible. 2: The data source is inaccessible.
SubType	String	mysql	The subtype of the data source. This parameter takes effect only when the DataSourceType parameter is set to rds. If the DataSourceType parameter is set to rds, the parameter can be set to mysql, sqlserver, or postgresql.
TenantId	Long	1234567	The ID of the tenant.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	20	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	233	The total number of data sources.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc14115159376359****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataSources
&Name=test
&ProjectId=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc14115159376359****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>233</TotalCount>
  <PageSize>20</PageSize>
  <PageNumber>1</PageNumber>
  <DataSources>
    <Status>1</Status>
    <Operator>193543050****</Operator>
    <ConnectStatus>1</ConnectStatus>
    <SubType>mysql</SubType>
    <Description>a connection</Description>
    <BindingCalcEngineId>123</BindingCalcEngineId>
    <ProjectId>123</ProjectId>
    <GmtModified>Mar 17, 2021 4:09:32 PM</GmtModified>
    <Sequence>300</Sequence>
    <Shared>false</Shared>
    <EnvType>1</EnvType>
    <Name>abc</Name>
    <DefaultEngine>false</DefaultEngine>
    <TenantId>1234567</TenantId>
    <GmtCreate>Mar 17, 2021 4:09:32 PM</GmtCreate>
    <DataSourceType>rds</DataSourceType>
    <Id>1</Id>
  </DataSources>
</Data>
<Success>true</Success>
```

JSON format


```
{
  "RequestId": "0bc14115159376359****",
  "HttpStatusCode": "200",
  "Data": {
    "TotalCount": "233",
    "PageSize": "20",
    "PageNumber": "1",
    "DataSources": [
      {
        "Status": "1",
        "Operator": "193543050****",
        "ConnectStatus": "1",
        "SubType": "mysql",
        "Description": "a connection",
        "BindingCalcEngineId": "123",
        "ProjectId": "123",
        "GmtModified": "Mar 17, 2021 4:09:32 PM",
        "Sequence": "300",
        "Shared": "false",
        "EnvType": "1",
        "Name": "abc",
        "DefaultEngine": "false",
        "TenantId": "1234567",
        "GmtCreate": "Mar 17, 2021 4:09:32 PM",
        "DataSourceType": "rds",
        "Id": "1"
      }
    ]
  },
  "Success": "true"
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.

HttpCode	Error code	Error message	Description
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.4. UpdateDataSource

Call the UpdateDataSource operation to update the data source.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateDataSource	The operation that you want to perform.
				The details of the data source. Example: - odps <pre>{ "accessId": "xssssss", "accessKey": "xsaxsaxsa", "authType": 2, "endpoint": "http://service.odps.aliyun.com/api", "project": "xsaxsax", "tag": "public" }</pre> - mysql <pre>{ "database": "xsaxsa", "instanceName": "rm-xsaxsa", "password": "xsaxsa", "rdsOwnerId": "xasxsa", "regionId": "cn-shanghai", "tag": "rds", "username": "xsaxsa" }</pre> - rds

Parameter	Type	Required	Example	Description
				<pre>"configType": 1, "tag": "rds", "database": "xsaxsa", "username": "xsaxsa", "password": "xssaxsa\$32050", "instanceName": "rm-xsaxs", "rdsOwnerId": "11111111" }</pre> - oss <pre>{ "accessId": "sssssx", "accessKey": "xsaxxsaxs", "bucket": "xsa-xs-xs", "endpoint": "http://oss-cn-shanghai.aliyuncs.com", "tag": "public" }</pre> - sqlserver <pre>{ "jdbcUrl": "jdbc:sqlserver://xsaxsa- xsaxsa.database.xxx.cn:123;DatabaseName=xsxs-xsxs", "password": "sdasda\$fs", "tag": "public", "username": "sxaxacdacdd" }</pre> - polardb <pre>{ "clusterId": "pc-sdadsadsa", "database": "dsadsadsa", "ownerId": "121212122", "password": "sdasdafssa", "region": "cn-shanghai", "tag": "polardb", "username": "asdadsads" }</pre> - oracle

Parameter	Type	Required	Example	Description
Content	String	Yes	<pre>{ "accessId": "xsaxsax", "accessKey": "xsaxsaxsa", "authType": 2, "endpoint": "http://service.o dps.aliyun.com/a pi", "project": "xsaxsax", "tag": "public" }</pre>	jdbcUrl: <pre>"jdbc:oracle:saaa:@xxxxx:1521:PRO D", "password": "sxasaxsa", "tag": "public", "username": "sasfadfa" }</pre> - mongodb <pre>{ "address": " [\"xsaxsaxsa.mongodb.rds.aliyuncs.c om:3717\"]", "database": "admin", "password": "sadsda@", "tag": "public", "username": "dsadsadas" }</pre> - emr <pre>{ "accessId": "xsaxsa", "emrClusterId": "C-dsads", "emrResourceQueueName": "default", "emrEndpoint": "emr.aliyuncs.com", "accessKey": "dsadsad", "emrUserId": "224833315798889783", "name": "sasdsadsa", "emrAccessMode": "simple", "region": "cn-shanghai", "authType": "2", "emrProjectId": "FP-sdadsad" }</pre> - postgresql

Parameter	Type	Required	Example	Description
				"jdbcUrl": "jdbc:postgresql://xxxx:1921/ssss", "password": "sdadsads", "tag": "public", "username": "sdsasda" } - analyticdb_for_mysql <pre>{ "instanceId": "am-sadsada", "database": "xsxsx", "username": "xsxsa", "password": "asdadsa", "connectionString": "am- xsxsxs.ads.aliyuncs.com:3306" }</pre> - hybriddb_for_postgresql <pre>{ "connectionString": "gp-xsaxsaxa- master.gpdbmaster.rds.aliyuncs.c om", "database": "xsaxsaxas", "password": "xsaxsaxsa@11", "instanceId": "gp-xsaxsaxsa", "port": "541132", "ownerId": "xsaxsaxsas", "username": "sadsad" }</pre> - holo <pre>{ "accessId": "xsaxsaxs", "accessKey": "xsaxsaxsa", "database": "xsaxsaxsa", "instanceId": "xsaxa", "tag": "aliyun" }</pre> - kafka

Parameter	Type	Required	Example	Description
				<pre> { "instanceId": "xsax-cn-xsaxsa", "regionId": "cn-shanghai", "tag": "aliyun", "ownerId": "121212121212" } </pre>
DataSourceId	Long	Yes	1	The ID of the data source. You can call the ListDataSources operation to obtain the ID.
EnvType	Integer	No	1	The environment in which the data source runs. Valid values: 0: development environment. 1: production environment.
RegionId	String	Yes	cn-shanghai	The region ID of the data source. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
Description	String	No	xxx	The description of the data source.
Status	String	No	ENABLED	The status of the data source. Valid values: - ENABLED indicates that the data source is in normal use. - DISABLED, indicating that the data source is disabled.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the data source was successfully updated.

Parameter	Type	Example	Description
HttpStatusCode	String	200	The HTTP status code returned.
RequestId	String	0bc14115159376359****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateDataSource
&Content={ "accessId": "xssssss", "accessKey": "xsaxsaxsa", "authType": 2, "endpoint": "http://service.o
dps.aliyun.com/api", "project": "xsaxsax", "tag": "public" }
&DataSourceId=1
&EnvType=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc14115159376359****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc14115159376359****",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.

HttpCode	Error code	Error message	Description
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.
400	MissingParam.ConnectionStr.Property	You must specify property %s required by the data source connection string.	The error message returned because the required properties of the connection string of the data source are not specified.
400	DataSource.CheckRamRoleFailure	Failed to check the RAM role. AccountId %s. roleArn %s.	The error message returned because the permissions on the data source have failed to be verified.
400	DataSource.ConnectionStrProperty.PatternNotMatch	The pattern value %s is invalid. The correct pattern is %s.	The error message returned because the connection string property of the data source does not match the specified rules.
400	DataSource.NotFound	The specified data source ID %s does not exist.	The error message returned because the data source does not exist.
400	Invalid.Connection.Empty	You must specify the connection parameter.	The error message returned because the data source parameter is left empty.
400	Invalid.DataSource.DataSourceTypeNotExist	The data source type %s does not exist.	The error message returned because the specified data resource type does not exist.
400	Invalid.DataSource.DataSourceTypeNotSupport	The dataSourceType %s and subType %s are not supported.	The error message returned because the data source type is not supported.
400	Invalid.Operation	You cannot delete the bindingCalcEngine that is bound to the data source.	The error message returned because you are not allowed to delete the compute engine instance with which the data source is associated.

HttpCode	Error code	Error message	Description
400	Invalid.Param.Odps.EngineIdOrConnectStr Empty	Either the compute engine ID or the connection string must be specified when data source type is ODPS.	The error message returned because the compute engine ID and connection string of the MaxCompute data source cannot be empty at the same time.
400	Required.Parameter.Empty	You must specify the parameter %s.	The error message returned because a required parameter is not specified.

2.1.6.1.5. DeleteDataSource

Deletes a data source.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteDataSource	The operation that you want to perform.
DataSourceId	Long	Yes	1	The ID of the data source. You can call the ListDataSources operation to obtain the ID.
RegionId	String	Yes	cn-shanghai	The region ID of the data source. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the request is successful.
HttpStatusCode	String	200	The HTTP status code returned.
RequestId	String	0bc1411515937635* ***	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteDataSource
&DataSourceId=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1411515937635****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1411515937635****",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.

HttpCode	Error code	Error message	Description
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.
400	Invalid.Operation	You cannot delete the bindingCalcEngine that is bound to the data source.	The error message returned because you are not allowed to delete the compute engine instance with which the data source is associated.

2.1.6.1.6. ListResourceGroups

Queries resource groups.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListResourceGroups	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the DataWorks workspace. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ResourceGroupType	Integer	No	3	The type of the resource group, including 0: DataWorks 2: MaxCompute 3: PAI 4: Data Integration 7: scheduling 8: App Studio 9: DataService Studio
Keyword	String	No	abc	The keyword. used for fuzzy queries by resource group name and ID.

Response parameters

Parameter	Type	Example	Description
Data	Array of Data		The list of resource groups.
BizExtKey	String	default	The extended field for the resource group.
Cluster	String	AY18G	The name of the cluster. This parameter is valid for MacCompute and Pai resource group types.
CreateTime	String	Jul 9, 2018 2:43:37 PM	The time when the cluster was created. Example: Jul 9, 2018 2:43:37 PM.
EnableKp	Boolean	false	Indicates whether the UID of an Alibaba Cloud account is used for access. 1 indicates that the MaxCompute engine uses the Alibaba Cloud Uid as the display name of the access account. 0 indicates that the MaxCompute engine uses the Alibaba Cloud account name as the display name. The remaining types are meaningless.
Id	Long	1234567	The ID of the resource group.
Identifier	String	e1815577-2f4e-4c5e-b29****	The identifier of the resource group.
IsDefault	Boolean	true	Indicates whether the resource group is the default resource group.
Mode	String	SHARE	The type of the resource group. Valid values: ISOLATE, SHARE, and DEVELOP. A value of ISOLATE indicates an exclusive resource group that adopts the subscription billing method. A value of SHARE indicates a shared resource group that adopts the pay-as-you-go billing method. A value of DEVELOP indicates a resource group for developers.
Name	String	abc	The name of the resource group.
ResourceGroup Type	String	PAI	The type of the resource group.

Parameter	Type	Example	Description
Sequence	Integer	300	The sequence.
Specs	Map	{}	The details of the resource group.
Status	Integer	0	The status of the resource group.
TenantId	Long	1234567	The ID of the tenant.
UpdateTime	String	Jul 9, 2018 2:43:37 PM	The last time when the resource group was updated.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1411515937635973****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListResourceGroups
&RegionId=cn-shanghai
&ResourceGroupType=3
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1411515937635973****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <IsDefault>true</IsDefault>
  <Status>0</Status>
  <Mode>SHARE</Mode>
  <CreateTime>Jul 9, 2018 2:43:37 PM</CreateTime>
  <Cluster>AY18G</Cluster>
  <Sequence>300</Sequence>
  <ResourceGroupType>PAI</ResourceGroupType>
  <Name>abc</Name>
  <TenantId>1234567</TenantId>
  <Identifier>e1815577-2f4e-4c5e-b29****</Identifier>
  <EnableKp>false</EnableKp>
  <BizExtKey>default</BizExtKey>
  <UpdateTime>Jul 9, 2018 2:43:37 PM</UpdateTime>
  <Id>1234567</Id>
  <Specs>{}</Specs>
</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1411515937635973****",
  "HttpStatusCode": 200,
  "Data": {
    "IsDefault": true,
    "Status": 0,
    "Mode": "SHARE",
    "CreateTime": "Jul 9, 2018 2:43:37 PM",
    "Cluster": "AY18G",
    "Sequence": 300,
    "ResourceGroupType": "PAI",
    "Name": "abc",
    "TenantId": 1234567,
    "Identifier": "e1815577-2f4e-4c5e-b29****",
    "EnableKp": false,
    "BizExtKey": "default",
    "UpdateTime": "Jul 9, 2018 2:43:37 PM",
    "Id": 1234567,
    "Specs": "{}"
  },
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.7. GetProject

Queries the information of a DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetProject	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace. You can call the ListProjects operation to obtain the ID.
RegionId	String	Yes	cn-shanghai	The region ID of the data source. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The description of the workspace.
Appkey	String	0	This parameter is deprecated.
BaseProject	Boolean	false	This parameter is deprecated.
DefaultDiResourceGroupIdentifier	String	group_280749521*** *	The identifier of the resource group that is allocated by default when you purchase an exclusive resource group for MaxCompute.
Destination	Integer	0	This parameter is deprecated.
DevStorageQuota	String	0	This parameter is deprecated.
DevelopmentType	Integer	4	The development type of the workspace. Currently, both are 4. This parameter is deprecated.
DisableDevelopment	Boolean	true	Indicates whether the development engineer role is disabled. Valid values: 0: The development engineer role is enabled. -1: The development engineer role is disabled. Default value: 0.
EnvTypes	List	["PROD"]	The environment of the workspace. Valid values: PROD and DEV. • PROD indicates a production environment. Workspaces in basic mode can run only in a production environment. • DEV indicates a development environment. Workspaces in standard mode can run in both a development environment and a production environment.
GmtCreate	String	Oct 10, 2019 3:42:53 PM	The time when the workspace was created. Example value: Dec 3, 2019 9:12:20 PM.

Parameter	Type	Example	Description
GmtModified	String	Dec 3, 2019 9:12:20 PM	The time when the workspace was modified. Example value: Dec 3, 2019 9:12:20 PM.
IsAllowDownload	Integer	1	Indicates whether the query result is allowed to be downloaded from DataStudio. Valid values: 1: The query result is allowed to be downloaded from DataStudio. 0: The query result is not allowed to be downloaded from DataStudio.
IsDefault	Integer	1	Indicates whether the workspace is a default workspace. Valid values: 1: The workspace is a default workspace. 0: The workspace is not a default workspace.
MaxFlowNode	Integer	1	This parameter is deprecated.
ProdStorageQuota	String	1	This parameter is deprecated.
ProjectDescription	String	abc	The description of the workspace.
ProjectId	Integer	27	The ID of the workspace.
ProjectIdentifier	String	abc	The unique identifier of the workspace.
ProjectMode	Integer	2	The mode in which the workspace runs. Valid values: 2: The workspace is in basic mode. 3: The workspace is in standard mode.
ProjectName	String	abc	The display name of the workspace.
ProjectOwnerBaseId	String	18229311****	The ID of the Alibaba Cloud account used by the workspace owner.

Parameter	Type	Example	Description
ProtectedMode	Integer	1	Indicates whether the workspace protection feature is enabled. Valid values: 1: Workspace protection is enabled. 0: Workspace protection is disabled.
ResidentArea	String	private	This parameter is deprecated.
SchedulerMaxRetryTimes	Integer	3	The default maximum number of automatic reruns that are allowed after an error occurs.
SchedulerRetryInterval	Integer	120000	The default interval at which an automatic rerun is performed. Unit: milliseconds. The maximum value of this parameter is 30 minutes. You must pay attention to the conversion between units.
Status	Integer	0	The status of the workspace. Valid values: - AVAILABLE(0): The workspace is normally running. - DELETED(1): The workspace is deleted. - INITIALIZING(2): The workspace is being initialized. - INIT_FAILED(3): The workspace fails to be initialized. - FORBIDDEN(4): The workspace is manually disabled. - DELETING(5): The workspace is being deleted. - DEL_FAILED(6): The workspace fails to be deleted. - FROZEN(7): The workspace is frozen due to overdue payments. - UPDATING(8): The workspace is being updated. The workspace enters this state after you associate a new compute engine with the workspace and the compute engine is being initialized. - UPDATE_FAILED(9): The workspace fails to be updated.

Parameter	Type	Example	Description
TablePrivacyMode	Integer	1	Indicates whether the MaxCompute tables in the workspace are visible to users within a tenant. Valid values: • 0: The MaxCompute tables are invisible to users within a tenant. • 1: The MaxCompute tables are visible to users within a tenant.
TenantId	Long	280749521	The ID of the tenant.
UseProxyOdpsAccount	Boolean	true	Indicates whether the MaxCompute compute engine is accessed by using a proxy account.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1411515937635973****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the information of the workspace is obtained.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetProject
&ProjectId=27
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1411515937635973****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <ProjectOwnerBaseld>18229311****</ProjectOwnerBaseld>
  <Destination>0</Destination>
  <ProjectName>abc</ProjectName>
  <IsAllowDownload>1</IsAllowDownload>
  <DevelopmentType>4</DevelopmentType>
  <GmtModified>Dec 3, 2019 9:12:20 PM</GmtModified>
  <DisableDevelopment>true</DisableDevelopment>
  <BaseProject>false</BaseProject>
  <ProjectDescription>abc</ProjectDescription>
  <ProjectMode>2</ProjectMode>
  <Status>0</Status>
  <ProjectIdentifier>abc</ProjectIdentifier>
  <IsDefault>1</IsDefault>
  <DevStorageQuota>0</DevStorageQuota>
  <SchedulerMaxRetryTimes>3</SchedulerMaxRetryTimes>
  <ProjectId>27</ProjectId>
  <ResidentArea>private</ResidentArea>
  <UseProxyOdpsAccount>true</UseProxyOdpsAccount>
  <TenantId>280749521</TenantId>
  <GmtCreate>Oct 10, 2019 3:42:53 PM</GmtCreate>
  <MaxFlowNode>1</MaxFlowNode>
  <ProtectedMode>1</ProtectedMode>
  <Appkey>0</Appkey>
  <TablePrivacyMode>1</TablePrivacyMode>
  <ProdStorageQuota>1</ProdStorageQuota>
  <DefaultDiResourceGroupId>group_280749521****</DefaultDiResourceGroupId>
  <SchedulerRetryInterval>120000</SchedulerRetryInterval>
  <EnvTypes>["PROD"]</EnvTypes>
</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1411515937635973****",
  "HttpStatusCode": 200,
  "Data": {
    "ProjectOwnerBaseId": "18229311****",
    "Destination": 0,
    "ProjectName": "abc",
    "IsAllowDownload": 1,
    "DevelopmentType": 4,
    "GmtModified": "Dec 3, 2019 9:12:20 PM",
    "DisableDevelopment": true,
    "BaseProject": false,
    "ProjectDescription": "abc",
    "ProjectMode": 2,
    "Status": 0,
    "ProjectIdentifier": "abc",
    "IsDefault": 1,
    "DevStorageQuota": 0,
    "SchedulerMaxRetryTimes": 3,
    "ProjectId": 27,
    "ResidentArea": "private",
    "UseProxyOdpsAccount": true,
    "TenantId": 280749521,
    "GmtCreate": "Oct 10, 2019 3:42:53 PM",
    "MaxFlowNode": 1,
    "ProtectedMode": 1,
    "Appkey": 0,
    "TablePrivacyMode": 1,
    "ProdStorageQuota": 1,
    "DefaultDiResourceGroupIdentifier": "group_280749521****",
    "SchedulerRetryInterval": 120000,
    "EnvTypes": "[\"PROD\"]"
  },
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

For a list of error codes, visit the [API Error Center](#).

2.1.6.1.8. AddProjectMemberToRole

Assigns a role to a user in a DataWorks workspace. Before you call this operation, you must add the user to the DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AddProjectMemberToRole	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The ID of the DataWorks workspace.
RoleCode	String	Yes	role_project_guest	Code for the DataWorks workspace role.
UserId	String	Yes	1234	The ID of the user.
ClientToken	String	No	1AFAE64E-D1BE-432B-A9****	The idempotent string. We recommend that you use a UUID.

Response parameters

Parameter	Type	Example	Description
RequestId	String	1AFAE64E-D1BE-432B-A9****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=AddProjectMemberToRole
&ProjectId=27
&RegionId=cn-shanghai
&RoleCode=role_project_guest
&UserId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>1AFAE64E-D1BE-432B-A9****</RequestId>
```

JSON format

```
{
  "RequestId": "1AFAE64E-D1BE-432B-A9****"
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.9. CreateProjectMember

Adds a user to a DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateProjectMember	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The ID of the DataWorks workspace.
UserId	String	Yes	1234	The ID of the user to be added.
ClientToken	String	No	1AFAE64E-D1BE-432B-A9****	The idempotent string. We recommend that you use a UUID.
RoleCode	String	No	role_project_guest	The role code is not required. If it is not empty, the user is added to this role.

Response parameters

Parameter	Type	Example	Description
RequestId	String	1AFAE64E-D1BE-432B-A9****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateProjectMember
&ProjectId=27
&RegionId=cn-shanghai
&UserId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>1AFAE64E-D1BE-432B-A9****</RequestId>
```

JSON format

```
{
  "RequestId": "1AFAE64E-D1BE-432B-A9****"
}
```


Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.
400	Invalid.Tenant.UserAlreadyInProject	The user already exists in the project.	The error message returned because the user to be added already exists in the workspace.
400	Invalid.Tenant.UserNotInTenant	The user is not in the tenant.	The error message returned because the member to be added is not in the tenant.

2.1.6.1.10. DeleteProjectMember

Removes a user from a DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteProjectMember	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The ID of the DataWorks workspace.
UserId	String	Yes	1234	The ID of the user.

Response parameters

Parameter	Type	Example	Description
RequestId	String	1AFAE64E-D1BE-432B-A9****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteProjectMember
&ProjectId=27
&RegionId=cn-shanghai
&UserId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>1AFAE64E-D1BE-432B-A9****</RequestId>
```

JSON format

```
{
  "RequestId": "1AFAE64E-D1BE-432B-A9****"
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.

HttpCode	Error code	Error message	Description
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.
400	Invalid.Tenant.UserAlreadyInProject	The user already exists in the project.	The error message returned because the user to be added already exists in the workspace.
400	Invalid.Tenant.UserNotInTenant	The user is not in the tenant.	The error message returned because the member to be added is not in the tenant.

2.1.6.1.11. ListProjectRoles

Queries roles in a DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListProjectRoles	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The ID of the DataWorks workspace.

Response parameters

Parameter	Type	Example	Description
ProjectRoleList	Array of Role		The list of roles in the workspace.
ProjectRoleCode	String	role_project_owner	The role code for the workspace.
ProjectRoleId	Integer	24	The role ID of the workspace.
ProjectRoleName	String	Project owner	The role name of the workspace.

Parameter	Type	Example	Description
ProjectRoleType	String	0	The role type of the workspace. Valid values: 0: default role type 1: user-defined role type
RequestId	String	1AFAE64E-D1BE-432B-A9****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListProjectRoles
&ProjectId=27
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>1AFAE64E-D1BE-432B-A9****</RequestId>
<ProjectRoleList>
  <ProjectRoleName> Project owner </ProjectRoleName>
  <ProjectRoleId>24</ProjectRoleId>
  <ProjectRoleCode>role_project_owner</ProjectRoleCode>
  <ProjectRoleType>0</ProjectRoleType>
</ProjectRoleList>
```

JSON format

```
{
  "RequestId": "1AFAE64E-D1BE-432B-A9****",
  "ProjectRoleList": [
    {
      "ProjectRoleName": "Project owner",
      "ProjectRoleId": "24",
      "ProjectRoleCode": "role_project_owner",
      "ProjectRoleType": "0"
    }
  ]
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.12. RemoveProjectMemberFromRole

Removes a role from a user in a DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RemoveProjectMemberFromRole	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The ID of the DataWorks workspace.

Parameter	Type	Required	Example	Description
RoleCode	String	Yes	role_project_guest	The role code for the DataWorks workspace. You can call the ListProjectRoles operation to obtain the list of character codes for the project. Valid values: - role_project_owner: project owners - role_project_admin: space administrators - role_project_dev: developers - role_project_pe: O&M personnel - role_project_deploy: deployers - role_project_guest: visitors - role_project_security: security administrators - role_project_tester: testers - role_project_erd: model designers
UserId	String	Yes	1234	The ID of the user.

Response parameters

Parameter	Type	Example	Description
RequestId	String	1AFAE64E-D1BE-432B-A9****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RemoveProjectMemberFromRole
&ProjectId=27
&RegionId=cn-shanghai
&RoleCode=role_project_guest
&UserId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>1AFAE64E-D1BE-432B-A9****</RequestId>
```

JSON format

```
{
  "RequestId": "1AFAE64E-D1BE-432B-A9*****"
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.13. ListProjectMembers

Queries members in a DataWorks workspace.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListProjectMembers	The operation that you want to perform.
ProjectId	Long	Yes	27	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region where the workspace resides.
PageNumber	Integer	No	1	The number of the page to return. Valid values: 1 to 50000

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The size of each page during a paged query. Valid values: 5 to 100

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned query results.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
ProjectMemberList	Array of ProjectMember		The list of members of the workspace.
Nick	String	zhangsan	The nickname of the workspace member.
ProjectMemberId	String	121	The ID of the user.
ProjectMemberName	String	Alice	The name of the member.
ProjectMemberType	String	0	The type of the member.
ProjectRoleList	Array of Role		The list of roles owned by members.

Parameter	Type	Example	Description
ProjectRoleCode	String	role_project_guest	The code of the role. Queries roles in a DataWorks workspace. Valid values: • role_project_owner: project owners • role_project_admin: space administrators • role_project_dev: developers • role_project_pe: O&M personnel • role_project_deploy: deployers • role_project_guest: visitors • role_project_security: security administrators • role_project_tester: testers • role_project_erd: model designers
ProjectRoleId	Integer	24	The ID of the role.
ProjectRoleName	String	Project owner	The name of the new role.
ProjectRoleType	String	0	The type of the role. Valid values: • 0: System default roles • 1: User-defined roles
Status	String	0	The status of the production studio.
TotalCount	Integer	3	The total number of entries returned.
RequestId	String	1AFAE64E-D1BE-432B-A9****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListProjectMembers
&ProjectId=27
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```

<TotalNum>5</TotalNum>
<PageNum>1</PageNum>
<PageSize>10</PageSize>
<RequestId>1AFAE64E-D1BE-432B-A9****</RequestId>
<ProjectMemberList>
  <CreateOn>2020-02-03 19:00:00</CreateOn>
  <Status>0</Status>
  <ProjectMemberType>5</ProjectMemberType>
  <ProjectMemberId>12312312</ProjectMemberId>
  <Nick>tom_xxx</Nick>
  <ProjectMemberName>Bob</ProjectMemberName>
</ProjectMemberList>
<ProjectMemberList>
  <ProjectRoleList>
    <ProjectRoleName>Visitors</ProjectRoleName>
    <ProjectRoleType>0</ProjectRoleType>
    <ProjectRoleCode>role_project_guest</ProjectRoleCode>
    <ProjectRoleId>1</ProjectRoleId>
  </ProjectRoleList>
</ProjectMemberList>

```

JSON format

```

{"RequestId":"1AFAE64E-D1BE-432B-A9****","Data":{"TotalCount":"3","PageSize":"10","ProjectMemberList":[{"Status":"0","ProjectMemberName":"Bob","ProjectMemberType":"0","ProjectMemberId":"121","Nick":"Bob"},{"ProjectRoleList":[{"ProjectRoleName":"Project owner","ProjectRoleId":"24","ProjectRoleCode":"role_project_guest","ProjectRoleType":"0"}]}],"PageNumber":"1"}}

```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.1.14. ListProjects

Queries DataWorks workspaces of the tenant to which a user belongs by page.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListProjects	The operation that you want to perform.
PageNumber	Integer	No	1	The number of the page to return.
RegionId	String	Yes	cn-shanghai	The ID of the region where the DataWorks workspace resides.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
PageResult	Struct		The result of the query.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ProjectList	Array of Project		The IDs of the DataWorks workspaces.
ProjectDescription	String	Test space	The description of the workspace.
ProjectId	Long	1212	The ID of the workspace.
ProjectIdentifier	String	test	The encoding of the workspace.
ProjectName	String	Test space	The name of the workspace.

Parameter	Type	Example	Description
ProjectOwnerBaseld	String	122222	The ID of the account used by the workspace owner.
ProjectStatus	Integer	0	The status of the workspace.
ProjectStatusCode	String	AVAILABLE	The status of the workspace.
TotalCount	Integer	123	The total number of entries returned.
RequestId	String	20658801***	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListProjects
&PageNumber=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>20658801***</RequestId>
<PageResult>
  <TotalCount>123</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <ProjectList>
    <ProjectIdentifier>test</ProjectIdentifier>
    <ProjectOwnerBaseld>122222</ProjectOwnerBaseld>
    <ProjectStatus>0</ProjectStatus>
    <ProjectStatusCode>AVAILABLE</ProjectStatusCode>
    <ProjectName>Test space</ProjectName>
    <ProjectId>1212</ProjectId>
    <ProjectDescription>Test space</ProjectDescription>
  </ProjectList>
</PageResult>
```

JSON format

```
{
  "RequestId": "20658801***",
  "PageResult": {
    "TotalCount": "123",
    "PageSize": "10",
    "PageNumber": "1",
    "ProjectList": [
      {
        "ProjectIdentifier": "test",
        "ProjectOwnerBaseId": "122222",
        "ProjectStatus": "0",
        "ProjectStatusCode": "AVAILABLE",
        "ProjectName": "Test Space",
        "ProjectId": "1212",
        "ProjectDescription": "Test space"
      }
    ]
  }
}
```

Error codes

HttpCode	Error code	Error message	Description
400	Invalid.Tenant.ConnectionNotExists	The connection does not exist.	The error message returned because the specified data source does not exist.
400	Invalid.Tenant.ProjectNotExists	The project does not exist.	The error message returned because the specified workspace does not exist.
400	Invalid.Tenant.UserAlreadyInProject	The user already exists in the project.	The error message returned because the user to be added already exists in the workspace.
403	Invalid.Tenant.UserIsNotProjectOwnerOrAdmin	The user is not a project administrator or owner.	The error message returned because the user is not the workspace owner or administrator.
400	Invalid.Tenant.UserNotInProject	The user is not in the project.	The error message returned because the user is not added to the workspace.
403	Invalid.Tenant.UserNotInTenant	The user is not in tenant.	The error message returned because the user does not belong to the tenant.

2.1.6.2. Metadata

2.1.6.2.1. GetMetaDBInfo

Queries the basic metadata information about a compute engine instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaDBInfo	The operation that you want to perform.
AppGuid	String	Yes	odps.some_engine_name	The ID of the compute engine instance. Specify the ID in the format of Compute engine type.Compute engine instance name.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
ClusterId	String	No	abc	The clusterId of the EMR cluster.
DatabaseName	String	No	abc	The database of the EMR cluster.
DataSourceType	String	No	emr	The data type. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The field information of the table.
AppGuid	String	odps.engine_name	The ID of the compute engine instance. The ID is in the format of Compute engine type.Compute engine instance name.
ClusterBizId	String	abc	The ID of the EMR cluster. This parameter is returned only when you query data in an EMR compute engine instance.
Comment	String	remarks	The remarks.

Parameter	Type	Example	Description
CreateTime	Long	1541576644000	The time when the compute engine instance was created.
Endpoint	String	http://service.odps-stg.aliyun-inc.com/	The endpoint of the service.
EnvType	Integer	1	The environment type of the metatable. Valid values: 0 and 1. A value of 0 indicates a non-production table. A value of 1 indicates a production table.
Location	String	hdfs://	The storage path of the metadatabase of the EMR cluster. This parameter is returned only when you query data in an EMR compute engine instance.
ModifyTime	Long	1541576644000	The last time when the compute engine instance was modified.
Name	String	abc	The name of the metadatabase.
OwnerId	String	23	The Alibaba Cloud account ID of the workspace owner.
OwnerName	String	323	The name of the workspace owner.
ProjectId	Long	22	The ID of the business process.
ProjectName	String	test	The name of the workspace.
ProjectNameCn	String	test	The display name of the workspace.
TenantId	Long	233	The ID of the tenant.
Type	String	hive	The type of the metadatabase.
RequestId	String	0bc1411515937****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaDBInfo
&AppGuid=odps.some_engine_name
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1411515937****</RequestId>
<Data>
  <ModifyTime>1541576644000</ModifyTime>
  <Comment>remarks</Comment>
  <OwnerName>323</OwnerName>
  <ProjectName>test</ProjectName>
  <ProjectNameCn>test</ProjectNameCn>
  <CreateTime>1541576644000</CreateTime>
  <ProjectId>22</ProjectId>
  <AppGuid>odps.engine_name</AppGuid>
  <EnvType>1</EnvType>
  <Name>abc</Name>
  <TenantId>233</TenantId>
  <Type>hive</Type>
  <Endpoint>http://service.odpsstg.aliyun-inc.com/</Endpoint>
  <OwnerId>23</OwnerId>
  <ClusterBizId>abc</ClusterBizId>
  <Location>hdfs://</Location>
</Data>
```

JSON format

```
{
  "RequestId": "0bc1411515937****",
  "Data": {
    "ModifyTime": "1541576644000",
    "Comment": "Remarks",
    "OwnerName": "323",
    "ProjectName": "test",
    "ProjectNameCn": "Test Item",
    "CreateTime": "1541576644000",
    "ProjectId": "22",
    "AppGuid": "odps.engine_name",
    "EnvType": "1",
    "Name": "abc",
    "TenantId": "233",
    "Type": "hive",
    "Endpoint": "http://service.odpsstg.aliyun-inc.com/",
    "OwnerId": "23",
    "ClusterBizId": "abc",
    "Location": "hdfs://"
  }
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeOut	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error code returned because an unknown error has occurred.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
400	Invalid.Meta.AppNotExist	The MaxCompute project does not exist.	The error message returned because the MaxCompute project does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.2. GetMetaDBTableList

Queries metatables in a compute engine instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaDBTableList	The operation that you want to perform.
AppGuid	String	Yes	odps.testProjectName	The ID of the project. Specify the ID in the format of <code>odps.{projectName}</code> . This parameter must be specified when you query metatables in a MaxCompute compute engine instance.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
PageNumber	Integer	No	1	The number of the page to return.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The number of the page to return. Default value: 10. Maximum value: 100.
ClusterId	String	No	abc	The ID of the E-MapReduce (EMR) cluster. This parameter must be specified when you query metatables in an EMR compute engine instance.
DatabaseName	String	No	abc	The name of the metadatabase.
DataSourceType	String	No	odps	The type of the compute engine instance. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The business data.
PageNumber	Integer	1	The number of the page to return.
PageSize	Integer	10	The number of the page to return. Default value: 10. Maximum value: 100.
TableEntityList	Array of TableEntityList		The list of metatables in the compute engine instance.
DatabaseName	String	abc	The name of the metadatabase.
TableGuid	String	odps.engine_name.table_name	The ID of the metatable.
TableName	String	tname	The name of the table.
TotalCount	Long	20	The total number of compute engine instances.
RequestId	String	0bc1ec92159376	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaDBTableList
&AppGuid=odps.testProjectName
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<Data>
  <TotalCount>20</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <TableEntityList>
    <TableName>1</TableName>
    <TableGuid>odps.engine_name.1</TableGuid>
    <DatabaseName>abc</DatabaseName>
  </TableEntityList>
</Data>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "Data": {
    "TotalCount": 20,
    "PageSize": 10,
    "PageNumber": 1,
    "TableEntityList": {
      "TableName": "tname",
      "TableGuid": "odps.engine_name.tname",
      "DatabaseName": "abc"
    }
  }
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.AppNotExist	The MaxCompute project does not exist.	The error message returned because the MaxCompute project does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Metadata.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.3. CheckMetaTable

Checks whether a table exists.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CheckMetaTable	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The region ID.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.
ClusterId	String	No	abc	The ID of the EMR cluster.
DatabaseName	String	No	abc	The name of the database of the EMR cluster.
TableName	String	No	abc	The name of the table of the EMR cluster.
DataSourceType	String	No	emr	The type of the data source. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the table exists.

Parameter	Type	Example	Description
RequestId	String	0bc1ec92159376	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CheckMetaTable
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.

HTTP status code	Error code	Error message	Description
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoComputeEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The specified table does not exist.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.2.4. CheckMetaPartition

Checks whether a partition exists.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CheckMetaPartition	The operation that you want to perform.
Partition	String	Yes	ds=202005	The name of the partition.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.
ClusterId	String	No	abc	The ID of the EMR cluster.
DatabaseName	String	No	abc	The name of the metadatabase.
TableName	String	No	abc	The name of the table.
DataSourceType	String	No	emr	The data type. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the partition exists.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CheckMetaPartition
&Partition=ds=202005
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the specified table does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.

HttpCode	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

For more information about how to use the CheckMetaPartition API, see [DataWorks OpenAPI example](#).

2.1.6.2.5. SearchMetaTables

Queries metatables based on specified conditions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SearchMetaTables	The operation that you want to perform.
AppGuid	String	Yes	odps.engine_name	The GUID of the workspace where the metatable resides.
DataSourceType	String	Yes	emr	The data type. Valid values: odps and emr.
Keyword	String	Yes	test	The keyword to specify.
RegionId	String	Yes	cn-shanghai	The region ID of the distributed instance.
PageNumber	Integer	No	1	The page number of the returned page.
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
EntityType	Integer	No	0	The type of the metatable. Valid values: 0 and 1. A value of 0 indicates a table. A value of 1 indicates a view. If you do not set this parameter, all types of metatables are queried.

Parameter	Type	Required	Example	Description
ClusterId	String	No	abc	The ID of the E-MapReduce (EMR) cluster. This parameter is returned only when you query data in an EMR compute engine instance.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The list of the entities.
ClusterId	String	abc	The ID of the EMR cluster.
DatabaseName	String	abc	The name of the metadatabase.
EntityType	Integer	0	The type of the metatable. Valid values: 0 and 1. A value of 0 indicates a table. A value of 1 indicates a view.
EnvType	Integer	1	The environment type. A value of 1 indicates the production environment. A value of 0 indicates the development environment.
OwnerId	String	123	The Alibaba Cloud account ID of the workspace owner.
ProjectId	Long	323	The ID of the workspace.
ProjectName	String	test	The name of the workspace.
TableGuid	String	odps.engine_name.table_name	The GUID of the metatable.
TableName	String	table_name	The name of the table.
TenantId	Long	12345	The ID of the tenant.

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
TotalCount	Long	100	The total number of returned entries.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=SearchMetaTables
&AppGuid=odps.engine_name.table_name
&DataSourceType=emr
&Keyword=test
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <DataEntityList>
    <EntityType>0</EntityType>
    <TableName>table_name</TableName>
    <TableGuid>odps.engine_name.table_name</TableGuid>
    <TenantId>12345</TenantId>
    <OwnerId>123</OwnerId>
    <ProjectName>test</ProjectName>
    <ClusterId>abc</ClusterId>
    <ProjectId>323</ProjectId>
    <DatabaseName>abc</DatabaseName>
    <EnvType>1</EnvType>
  </DataEntityList>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "DataEntityList": {
      "EntityType": 0,
      "TableName": "table_name",
      "TableGuid": "odps.engine_name.table_name",
      "TenantId": 12345,
      "OwnerId": 123,
      "ProjectName": "test",
      "ClusterId": "abc",
      "ProjectId": 323,
      "DatabaseName": "abc",
      "EnvType": 1
    }
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the specified table does not exist.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.6. GetMetaTableBasicInfo

Queries the basic information about a metatable.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableBasicInfo	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint that is used to call the operation.

Parameter	Type	Required	Example	Description
TableGuid	String	Yes	odps.engine_name.table_name	The globally unique identifier (GUID) of the MaxCompute table. The value of this parameter is in the odps.projectName.tableName format.
ClusterId	String	No	abc	The ID of the E-MapReduce (EMR) cluster. This parameter must be specified when DataSourceType is set to emr. You can log on to the EMR console to obtain the ID.
DatabaseName	String	No	abc	The name of the metadatabase. You can call the ListMetaDB operation to obtain the name.
TableName	String	No	abc	The name of the metatable.
DataSourceType	String	No	emr	The data type. Valid values: odps and emr.
Extension	Boolean	No	false	Specifies whether extended fields are included in query results. The extended fields include ReadCount, FavoriteCount, and ViewCount. This parameter only takes effect when DataSourceType is set to odps.

Response parameters

Parameter	Type	Example	Description
Data	Struct		Business data.
Caption	String	Display name	The display name of the metatable.
ClusterId	String	abc	The ID of the EMR cluster.
Comment	String	test table	The comment of the metatable.

Parameter	Type	Example	Description
CreateTime	Long	1589870294000	The time when the metatable was created.
DataSize	Long	10	The storage space that is occupied by the metatable. Unit: bytes.
DatabaseName	String	abc	The name of the metadatabase.
EnvType	Integer	1	The environment type. Valid values: 0: development environment 1: production environment
FavoriteCount	Long	6	The number of times the metatable was added to a favorite list. This parameter is returned only when the Extension parameter is set to true and only takes effect when the DataSourceType parameter is set to odps.
IsPartitionTable	Boolean	true	Indicates whether the metatable is a partitioned table.
IsVisible	Integer	1	The scope in which the metatable is visible. Valid values: 0: workspace members 1: users within the tenant 2: all tenants 3: the metatable owner
LastAccessTime	Long	1589870294000	The time when the metatable was last accessed.
LastDdlTime	Long	1589870294000	The time when the schema was last changed.
LastModifyTime	Long	1589870294000	The time when the metatable was last updated.
LifeCycle	Integer	5	The lifecycle of the metatable. Unit: days.
Location	String	hdfs://	The storage location of the Hive database.

Parameter	Type	Example	Description
OwnerId	String	123	The ID of the metatable owner.
PartitionKeys	String	ab	The Hive partition.
ProjectId	Long	232	The ID of the workspace.
ProjectName	String	test	The name of the workspace.
ReadCount	Long	3	The number of times the metatable was read. The parameter is returned only when the Extension parameter is set to true and only takes effect when the DataSourceType parameter is set to odps.
TableGuid	String	odps.engine_name.table_name	The GUID of the metatable.
TableName	String	table_name	The name of the table.
TenantId	Long	12345	The ID of the tenant.
ViewCount	Long	2	The number of times the metatable was viewed. The parameter is returned only when the Extension parameter is set to true and only takes effect when the DataSourceType parameter is set to odps.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1411515937	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableBasicInfo
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1411515937</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <LifeCycle>5</LifeCycle>
  <Comment>test table</Comment>
  <IsPartitionTable>true</IsPartitionTable>
  <ClusterId>abc</ClusterId>
  <ProjectName>test</ProjectName>
  <LastDdlTime>1589870294000</LastDdlTime>
  <CreateTime>1589870294000</CreateTime>
  <DatabaseName>abc</DatabaseName>
  <ProjectId>232</ProjectId>
  <ViewCount>2</ViewCount>
  <LastAccessTime>1589870294000</LastAccessTime>
  <LastModifyTime>1589870294000</LastModifyTime>
  <Caption>displayName</Caption>
  <EnvType>1</EnvType>
  <TableName>table_name</TableName>
  <TenantId>12345</TenantId>
  <TableGuid>odps.engine_name.table_name</TableGuid>
  <OwnerId>123</OwnerId>
  <DataSize>10</DataSize>
  <FavoriteCount>6</FavoriteCount>
  <PartitionKeys>ab</PartitionKeys>
  <IsVisible>1</IsVisible>
  <ReadCount>3</ReadCount>
  <Location>hdfs://</Location>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1411515937",
  "HttpStatusCode": 200,
  "Data": {
    "LifeCycle": 5,
    "Comment": "test table",
    "IsPartitionTable": true,
    "ClusterId": "abc",
    "ProjectName": "test",
    "LastDdlTime": 1589870294000,
    "CreateTime": 1589870294000,
    "DatabaseName": "abc",
    "ProjectId": 232,
    "ViewCount": 2,
    "LastAccessTime": 1589870294000,
    "LastModifyTime": 1589870294000,
    "Caption": "Display name",
    "EnvType": 1,
    "TableName": "table_name",
    "TenantId": 12345,
    "TableGuid": "odps.engine_name.table_name",
    "OwnerId": 123,
    "DataSize": 10,
    "FavoriteCount": 6,
    "PartitionKeys": "ab",
    "IsVisible": 1,
    "ReadCount": 3,
    "Location": "hdfs://"
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.7. GetMetaTableColumn

Queries the field information of a table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableColumn	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint that is used to call the operation.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table. You can call the GetMetaDBTableList operation to query the GUID of the table.
PageNum	Integer	No	1	The page number of the returned page.
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ClusterId	String	No	C-010A704DA760***	The ID of the E-MapReduce (EMR) cluster. You can log on to the EMR console to obtain the cluster ID.

Parameter	Type	Required	Example	Description
DatabaseName	String	No	abc	The name of the database of the EMR cluster. You can call the ListMetaDB operation to query the database name.
TableName	String	No	abc	The name of the table of the EMR cluster. You can call the GetMetaDBTableList operation to query the table name.
DataSourceType	String	No	emr	The type of the data source. Only EMR is supported.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The field information of the table.
ColumnList	Array of ColumnList		The fields returned.
Caption	String	data column	The description of the field.
ColumnGuid	String	odps.engine_name.table_name.name	The GUID of the field.
ColumnName	String	name	The name of the field.
ColumnType	String	string	The data type of the field.
Comment	String	comment	The remarks of the field.
IsForeignKey	Boolean	true	Indicates whether the field is a foreign key. Valid values: - true: The field is a foreign key. - false: The field is not a foreign key.
IsPartitionColumn	Boolean	false	Indicates whether the field is a partition field. Valid values: - true: The field is a partition field. - false: The field is not a partition field.

Parameter	Type	Example	Description
IsPrimaryKey	Boolean	false	Indicates whether the field is the primary key. Valid values: • true: The field is the primary key. • false: The field is not the primary key.
Position	Integer	1	The ordinal number of the field.
PageNum	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Long	100	The total number of fields.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableColumn
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageNum>1</PageNum>
  <PageSize>10</PageSize>
  <ColumnList>
    <ColumnName>name</ColumnName>
    <Comment>comment </Comment>
    <ColumnGuid>odps.engine_name.table_name.name</ColumnGuid>
    <ColumnType>string</ColumnType>
    <IsPartitionColumn>false</IsPartitionColumn>
    <IsForeignKey>true</IsForeignKey>
    <Caption>data column</Caption>
    <IsPrimaryKey>false</IsPrimaryKey>
  </ColumnList>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": "100",
    "PageNum": "1",
    "PageSize": "10",
    "ColumnList": {
      "ColumnName": "name",
      "Comment": "comment ",
      "ColumnGuid": "odps.engine_name.table_name.name",
      "ColumnType": "string",
      "IsPartitionColumn": "false",
      "IsForeignKey": "true",
      "Caption": "data column",
      "IsPrimaryKey": "false"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalError.Meta.TenantTimeOut	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.GuidFormat	The GUID format is invalid.	The error message returned because the GUID format is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Metadata.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Metadata.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Metadata.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Metadata.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.8. GetMetaTableFullInfo

Queries the complete information about a table, including the field information.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableFullInfo	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint that is used to call the operation.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table. You can call the GetMetaDBTableList operation to query the GUID of the table.

Parameter	Type	Required	Example	Description
PageNum	Integer	No	1	The page number of the returned page.
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ClusterId	String	No	C-010A704DA760*** *	The ID of the E-MapReduce (EMR) cluster. You can log on to the EMR console to obtain the cluster ID.
DatabaseName	String	No	abc	The name of the database of the EMR cluster. You can call the ListMetaDB operation to query the database name.
TableName	String	No	abc	The name of the table of the EMR cluster. You can call the GetMetaDBTableList operation to query the table name.
DataSourceType	String	No	emr	The type of the data source. Only EMR is supported.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The field information of the table.
ClusterId	String	C-010A704DA760****	The ID of the EMR cluster.
ColumnList	Array of ColumnList		The columns.
Caption	String	data comment	The description of the field.
ColumnGuid	String	odps.engine_name.table_name.1	The GUID of the field.
ColumnName	String	1	The name of the field.

Parameter	Type	Example	Description
ColumnType	String	string	The data type of the field.
Comment	String	comment	The remarks of the field.
IsForeignKey	Boolean	true	Indicates whether the field is a foreign key. Valid values: - true: The field is a foreign key. - false: The field is not a foreign key.
IsPartitionColumn	Boolean	true	Indicates whether the field is a partition field. Valid values: - true: The field is a partition field. - false: The field is not a partition field.
IsPrimaryKey	Boolean	false	Indicates whether the field is the primary key. Valid values: - true: The field is the primary key. - false: The field is not the primary key.
Position	Integer	1	The ordinal number of the field.
Comment	String	comment	The remarks of the metatable.
CreateTime	Long	1589870293000	The time when the metatable was created. You can convert the timestamp to the date based on the time zone that you use.
DataSize	Long	10	The storage space occupied by the table. Unit: byte.
DatabaseName	String	abc	The name of the metadatabase.
EnvType	Integer	1	The environment type. Valid values: 0: development environment 1: production environment

Parameter	Type	Example	Description
IsVisible	Integer	1	Indicates whether the table is visible. Valid values: 0: workspace members 1: tenants 2: among tenants 3: the metatable owner
LastAccessTime	Long	1589870294000	The time when the metatable was last accessed. You can convert the timestamp to the date based on the time zone that you use.
LastDdlTime	Long	1589870294000	The last time when the metatable schema was changed. You can convert the timestamp to the date based on the time zone that you use.
LastModifyTime	Long	1589870294000	The last time when the metatable was updated. You can convert the timestamp to the date based on the time zone that you use.
LifeCycle	Integer	5	The lifecycle of the table. Unit: days.
Location	String	hdfs://localhost:7777/user/hadoop/test.txt	The storage address of the Hive table.
OwnerId	String	123	The ID of the metatable owner.
PartitionKeys	String	abc	The partition key.
ProjectId	Long	22	The ID of the workspace to which the node belongs.
ProjectName	String	test	The GUID of the workspace where the metatable resides.
TableGuid	String	odps.engine_name.table_name	The GUID of the metatable.
TableName	String	table_name	The name of the table.

Parameter	Type	Example	Description
TenantId	Long	12345	The ID of the tenant.
TotalColumnCount	Long	22	The total number of fields.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1411515937****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableFullInfo
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1411515937****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <LifeCycle>5</LifeCycle>
  <Comment>comment</Comment>
  <ProjectName>test</ProjectName>
  <ClusterId>abc</ClusterId>
  <LastDdlTime>1589870294000</LastDdlTime>
  <CreateTime>1589870293000</CreateTime>
  <TotalColumnCount>22</TotalColumnCount>
  <ProjectId>22</ProjectId>
  <DatabaseName>abc</DatabaseName>
  <LastModifyTime>1589870294000</LastModifyTime>
  <LastAccessTime>1589870294000</LastAccessTime>
  <EnvType>1</EnvType>
  <TableName>table_name</TableName>
  <TableGuid>odps.engine_name.table_name</TableGuid>
  <TenantId>12345</TenantId>
  <OwnerId>123</OwnerId>
  <DataSize>10</DataSize>
  <PartitionKeys>abc</PartitionKeys>
  <ColumnList>
    <ColumnName>1</ColumnName>
    <Comment>comment</Comment>
    <ColumnGuid>odps.engine_name.table_name.1</ColumnGuid>
    <ColumnType>string</ColumnType>
    <IsPartitionColumn>true</IsPartitionColumn>
    <IsForeignKey>true</IsForeignKey>
    <Caption>data comment</Caption>
    <IsPrimaryKey>false</IsPrimaryKey>
  </ColumnList>
  <IsVisible>1</IsVisible>
  <Location>hdfs://localhost:777/user/hadoop/test.txt</Location>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1411515937****",
  "HttpStatusCode": 200,
  "Data": {
    "LifeCycle": 5,
    "Comment": "comment",
    "ProjectName": "test",
    "ClusterId": "abc",
    "LastDdlTime": 1589870294000,
    "CreateTime": 1589870293000,
    "TotalColumnCount": 22,
    "ProjectId": 22,
    "DatabaseName": "abc",
    "LastModifyTime": 1589870294000,
    "LastAccessTime": 1589870294000,
    "EnvType": 1,
    "TableName": "table_name",
    "TableGuid": "odps.engine_name.table_name",
    "TenantId": 12345,
    "OwnerId": 123,
    "DataSize": 10,
    "PartitionKeys": "abc",
    "ColumnList": {
      "ColumnName": 1,
      "Comment": "comment",
      "ColumnGuid": "odps.engine_name.table_name.1",
      "ColumnType": "string",
      "IsPartitionColumn": true,
      "IsForeignKey": true,
      "Caption": "data comment",
      "IsPrimaryKey": false
    },
    "IsVisible": 1,
    "Location": "hdfs://localhost:777/user/hadoop/test.txt"
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeOut	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Metadata.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.9. GetMetaTablePartition

Queries the partitions of a metatable.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTablePartition	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ClusterId	String	No	abc	The ID of the E-MapReduce (EMR) cluster. You can set this parameter only when you query data in an EMR compute engine instance.
DatabaseName	String	No	abc	The name of the metadatabase.
TableName	String	No	abc	The name of the metatable.

Parameter	Type	Required	Example	Description
DataSourceType	String	No	emr	The type of the data. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The list of the entities.
Comment	String	remarks	The remarks.
CreateTime	Long	1590032868000	The time when the partition was created.
DataSize	Long	19	Specifies the size of each part during multipart upload, download, or copy of large objects. Unit: bytes.
ModifiedTime	Long	1590032868000	The last time when the partition was modified.
PartitionGuid	String	odps.engine_name.table_name.pt=20170614	The GUID of the partition.
PartitionLocation	String	abc	The address of the Hive partition.
PartitionName	String	pt=20170614	The name of the partition.
PartitionPath	String	abc	The directory of the partition.
PartitionType	String	abc	The type of the partition.
RecordCount	Long	233	The number of entries in the partition.
TableGuid	String	odps.engine_name.table_name	The GUID of the table.

Parameter	Type	Example	Description
PageNumber	Integer	1	The number of the page to return.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Long	100	The total number of partitions.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTablePartition
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <DataEntityList>
    <PartitionName>pt=20170614</PartitionName>
    <PartitionGuid>odps.engine_name.table_name.pt=20170614</PartitionGuid>
    <Comment>remarks</Comment>
    <TableGuid>odps.engine_name.table_name</TableGuid>
    <ModifiedTime>1590032868000</ModifiedTime>
    <CreateTime>1590032868000</CreateTime>
    <DataSize>19</DataSize>
    <PartitionPath>abc</PartitionPath>
    <PartitionType>abc</PartitionType>
    <RecordCount>233</RecordCount>
    <PartitionLocation>abc</PartitionLocation>
  </DataEntityList>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "DataEntityList": {
      "PartitionName": "pt=20170614",
      "PartitionGuid": "odps.engine_name.table_name.pt=20170614",
      "Comment": "Remarks",
      "TableGuid": "odps.engine_name.table_name",
      "ModifiedTime": 1590032868000,
      "CreateTime": 1590032868000,
      "DataSize": 19,
      "PartitionPath": "abc",
      "PartitionType": "abc",
      "RecordCount": 233,
      "PartitionLocation": "abc"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}

```


Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.10. GetMetaTableOutput

Queries the node output information in a metatable.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableOutput	The operation that you want to perform.
EndDate	String	Yes	2020-05-27	The end of the time range to query.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
StartDate	String	Yes	2020-06-27	The beginning of the time range to query.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.
PageNumber	Integer	No	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The list of the entities.
EndTime	String	1511107200000	The end time of the node.
ProjectId	Long	333	The ID of the workspace.
StartTime	String	1511107200000	The start time of the node.
TableGuid	String	odps.engine_name.table_name	The GUID of the table.
TaskId	String	32	The ID of the node.
TaskInstanceId	Long	3521	The ID of the DTS instance.
WaitTime	String	323	The time spent waiting for available resources to run the node.
PageNumber	Integer	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Long	100	The total number of returned entries.
ErrorCode	String	1031203110005	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableOutput
&EndDate=2020-05-27
&RegionId=cn-shanghai
&StartDate=2020-06-27
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <DataEntityList>
    <TaskInstanceId>3521</TaskInstanceId>
    <TableGuid>odps.engine_name.table_name</TableGuid>
    <TaskId>32</TaskId>
    <EndTime>1511107200000</EndTime>
    <StartTime>1511107200000</StartTime>
    <ProjectId>333</ProjectId>
    <WaitTime>323</WaitTime>
  </DataEntityList>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "DataEntityList": {
      "TaskInstanceId": 3521,
      "TableGuid": "odps.engine_name.table_name",
      "TaskId": 32,
      "EndTime": 1511107200000,
      "StartTime": 1511107200000,
      "ProjectId": 333,
      "WaitTime": 323
    }
  },
  "ErrorCode": "1031203110005",
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HttpCode	Error code	Error message	Description
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.11. GetMetaTableChangeLog

Queries the change logs of a metatable.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableChangeLog	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint that is used to call the operation.
TableGuid	String	Yes	odps.engine_name.table_name	The globally unique identifier (GUID) of the table. Specify the GUID in the format of odps.projectName.tableName. You can call the GetMetaDBTableList operation to query the GUID of the table. To query the change logs of a MaxCompute table, you must call the GetMetaTableChangeLog operation.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ChangeType	String	No	ALTER_TABLE	The type of the change. Valid values: CREATE_TABLE, ALTER_TABLE, DROP_TABLE, ADD_PARTITION, and DROP_PARTITION.
ObjectType	String	No	TABLE	The entity on which the change is made. Valid values: TABLE and PARTITION.

Parameter	Type	Required	Example	Description
StartDate	String	No	2020-06-01 00:00:00	The beginning of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format. - By default, the system uses the current time as the value of this parameter if the time that you specify is invalid. - If both the values of the StartDate and EndDate parameters are invalid, the system automatically queries the change logs that are generated within the last 30 days.
EndDate	String	No	2020-06-02 00:00:00	The end of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format. - By default, the system uses the current time as the value of this parameter if the time that you specify is invalid. - If both the values of the StartDate and EndDate parameters are invalid, the system automatically queries the change logs that are generated within the last 30 days.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The list of the entities.
ChangeContent	String	<pre>" [{"action\":"ADD_COLUMN","value": [{"originName\":"", ,"originType\":"", ,"originComment\":"", "name\":"id", "type\":"struct", <name:string>","comment\":""}]]",</pre>	The content of the change.
ChangeType	String	CREATE_TABLE	The type of the change.

Parameter	Type	Example	Description
CreateTime	Long	1590722845000	The time when the table was created.
ModifiedTime	Long	1590722845000	The time when the table was modified.
ObjectType	String	TABLE	The entity on which the change was made. Valid values: TABLE and PARTITION.
Operator	String	abc	The name of the operator.
PageNumber	Integer	10	The number of the returned page.
PageSize	Integer	1	The number of entries returned on each page. Default value: 10. Maximum value: 100.
TotalCount	Long	10	The total number of returned entries.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableChangeLog
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>10</TotalCount>
  <PageSize>1</PageSize>
  <PageNumber>10</PageNumber>
  <DataEntityList>
    <Operator>abc</Operator>
    <ObjectType>TABLE</ObjectType>
    <ChangeContent>[{"action":"ADD_COLUMN","value":[{"originName":"","originType":"","originComment":"","name":"id","type":"struct<name:string>","comment":""}]}]</ChangeContent>
    <ModifiedTime>1590722845000</ModifiedTime>
    <ChangeType>CREATE_TABLE</ChangeType>
    <CreateTime>1590722845000</CreateTime>
  </DataEntityList>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 10,
    "PageSize": 1,
    "PageNumber": 10,
    "DataEntityList": {
      "Operator": "abc",
      "ObjectType": "TABLE",
      "ChangeContent": "[{"action": "ADD_COLUMN", "value": [{"originName": "", "originType": "", "originComment": "", "name": "id", "type": "struct<name:string>", "comment": ""}]}]",
      "ModifiedTime": "1590722845000",
      "ChangeType": "CREATE_TABLE",
      "CreateTime": "1590722845000"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}

```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.12. GetMetaTableIntroWiki

Queries the instructions on how to use a metatable.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableIntroWiki	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.
WikiVersion	Long	No	1	The version of the instructions.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The field information of the table.
Content	String	# businessRequirement 	The description of the metatable.

Parameter	Type	Example	Description
CreateTime	Long	1584444247000	The time when the metatable was created.
Creator	String	abc	The UID of the user who created the metatable.
CreatorName	String	abc	The name of the user who created the metatable.
ModifiedTime	Long	1584444247000	The time when the table was modified.
Version	Long	1	The version number.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableIntroWiki
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>0bc1ec92159376****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <ModifiedTime>1584444247000</ModifiedTime>
  <Version>1</Version>
  <Content># businessRequirement\n\n&lt;a name=&quot;xiw5n&quot;</Content>
  <CreateTime>1584444247000</CreateTime>
  <Creator>abc</Creator>
  <CreatorName>abc</CreatorName>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "ModifiedTime": 1584444247000,
    "Version": 1,
    "Content": "Content": "# businessRequirement\n\n&lt;a name=&quot;xiw5n&quot;",
    "CreateTime": 1584444247000,
    "Creator": "abc",
    "CreatorName": "abc"
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}

```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoComputeEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.

HttpCode	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.13. GetMetaTableLineage

Queries the lineage of a metadata table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableLineage	The operation that you want to perform.
Direction	String	Yes	up	Specifies whether to query the ancestor or descendant lineage of the field. A value of UP indicates the ancestor lineage. A value of DOWN indicates the descendant lineage.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.
NextPrimaryKey	String	No	odps.engine_name.table_name1	The key that is used to obtain the next page of query results. Set this parameter based on the value of the response parameter NextPrimaryKey when the value of the response parameter HasNext is true in the last query.
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ClusterId	String	No	abc	The ID of the EMR cluster.
DatabaseName	String	No	abc	The name of the metadata database.

Parameter	Type	Required	Example	Description
TableName	String	No	abc	The name of the table.
DataSourceType	String	No	emr	The data type. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The information about the table.
TableGuid	String	odps.tt.name	The GUID of the metatable.
TableName	String	name	The name of the table.
HasNext	Boolean	true	Indicates whether the next page has more query results.
NextPrimaryKey	String	odps.tt.2	The key that is used to obtain the next page of query results. This parameter is returned when the value of the response parameter HasNext is true.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableLineage
&Direction=up
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <DataEntityList>
    <TableName>1</TableName>
    <TableGuid>odps.tt.name</TableGuid>
  </DataEntityList>
  <HasNext>true</HasNext>
  <NextPrimaryKey>odps.tt.name</NextPrimaryKey>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "DataEntityList": {
      "TableName": 1,
      "TableGuid": "odps.tt.name"
    },
    "HasNext": true,
    "NextPrimaryKey": "odps.tt.name"
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.GuidFormat	The GUID format is invalid.	The error message returned because the GUID format is invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.

2.1.6.2.14. GetMetaColumnLineage

Queries the lineage of a field.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaColumnLineage	The operation that you want to perform.
ColumnGuid	String	Yes	odps.engine_name.table_name.column_name	The GUID of the field.
Direction	String	Yes	up	Specifies whether to query the ancestor or descendant lineage of the field. A value of UP indicates the ancestor lineage. A value of DOWN indicates the descendant lineage.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
PageNum	Integer	No	1	The number of the page to return.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
ClusterId	String	No	abc	The ID of the EMR cluster.
DatabaseName	String	No	abc	The name of the metadatabase.
TableName	String	No	abc	The name of the table.
ColumnName	String	No	abc	The name of the field.
DataSourceType	String	No	emr	The data type. Valid values: odps and emr.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The list of metatables in the compute engine instance.
ClusterId	String	abc	The ID of the EMR cluster.
ColumnGuid	String	odps.engine_name.table_name.1	The GUID of the field.
ColumnName	String	1	The field that is monitored.
DatabaseName	String	abc	The name of the metadatabase.
TableName	String	abc	The name of the metatable.
PageNum	Integer	1	The number of the page to return.

Parameter	Type	Example	Description
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Long	20	The total number of fields.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaColumnLineage
&ColumnGuid=odps.engine_name.table_name.column_name
&Direction=up
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>20</TotalCount>
  <PageNum>1</PageNum>
  <PageSize>10</PageSize>
  <DataEntityList>
    <ColumnName>1</ColumnName>
    <TableName>abc</TableName>
    <ColumnGuid>odps.engine_name.table_name.1</ColumnGuid>
    <ClusterId>abc</ClusterId>
    <DatabaseName>abc</DatabaseName>
  </DataEntityList>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 20,
    "PageNum": 1,
    "PageSize": 10,
    "DataEntityList": {
      "ColumnName": 1,
      "TableName": "abc",
      "ColumnGuid": "odps.engine_name.table_name.1",
      "ClusterId": "abc",
      "DatabaseName": "abc"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoComputeEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.15. CreateMetaCategory

Creates a category.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateMetaCategory	The operation that you want to perform.
Name	String	Yes	category_name	The name of the category.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
Comment	String	No	category 1	The remarks of the category.
ParentId	Long	No	0	The ID of the parent category.

Response parameters

Parameter	Type	Example	Description
Data	Struct		Category information.
CategoryId	Long	223	The ID of the category.

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateMetaCategory
&Name=category_name
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <CategoryId>223</CategoryId>
</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "CategoryId": 223
  },
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalError.Meta.Unknown	An unknown error occurred while processing your request.	The error message returned because an unknown error has occurred.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Invalid.Meta.CategoryForbidden	Failed to delete the category. The current category is associated with a table and cannot be operated.	The error message returned because the current category is associated to a table.
400	Invalid.Meta.CategoryIllegalDepth	A category can contain a maximum of 4 levels.	The error message returned because the category level exceeds 4.
403	Invalid.Meta.CategoryIllegalTenant	The tenant ID is different in the parent and child categories.	The error message returned because the tenant ID is different in the parent and child categories.
400	Invalid.Meta.CategoryNameDuplicate	The category name is duplicated at this level.	The error message returned because the category name is duplicated at this level.
500	Invalid.Meta.CategoryRelationDuplicate"	The table is already bound to the category.	The error message returned because the category is already associated with the table.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.16. DeleteMetaCategory

Deletes a category.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteMetaCategory	The operation that you want to perform.
CategoryId	Long	Yes	23	The ID of the category.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the category was deleted.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteMetaCategory
&CategoryId=23
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.

HttpCode	Error code	Error message	Description
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeOut	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error code returned because an unknown error has occurred.
403	Invalid.Meta.CategoryForbidden	Failed to delete the category. The current category is associated with a table and cannot be operated.	The error message returned because the current category is associated to a table.
400	Invalid.Meta.CategoryIllegalDepth	A category can contain a maximum of 4 levels.	The error message returned because the category level exceeds 4.
403	Invalid.Meta.CategoryIllegalTenant	The tenant ID is different in the parent and child categories.	The error message returned because the tenant ID is different in the parent and child categories.
400	Invalid.Meta.CategoryNameDuplicate	The category name is duplicated at this level.	The error message returned because the category name is duplicated at this level.
500	Invalid.Meta.CategoryRelationDuplicate"	The table is already bound to the category.	The error message returned because the category is already associated with the table.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.17. UpdateMetaCategory

Updates a category.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateMetaCategory	The operation that you want to perform.
CategoryId	Long	Yes	2	The ID of the category.
Name	String	Yes	category name	The name of the category.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
Comment	String	No	category name	The remarks of the category.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the category was updated.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateMetaCategory
&CategoryId=2
&Name=category name
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error code returned because an unknown error has occurred.
403	Invalid.Meta.CategoryForbidden	Failed to delete the category. The current category is associated with a table and cannot be operated.	The error message returned because the category contains associated tables and cannot be deleted.
400	Invalid.Meta.CategoryIllegalDepth	A category can contain a maximum of 4 levels.	The error message returned because the category can contain a maximum of 4 levels.
403	Invalid.Meta.CategoryIllegalTenant	The tenant ID is different in the parent and child categories.	The error message returned because the tenant ID is different in the parent and child categories.

HttpCode	Error code	Error message	Description
400	Invalid.Meta.CategoryNameDuplicate	The category name is duplicated at this level.	The error message returned because the category name is duplicated at this level.
500	Invalid.Meta.CategoryRelationDuplicate"	The table is already bound to the category.	The error message returned because the category is already associated with the table.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.18. GetMetaCategory

Queries a category tree.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaCategory	The operation that you want to perform.

Parameter	Type	Required	Example	Description
ParentCategoryId	Long	Yes	333	The ID of the category tree.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
PageNum	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The returned data.
DataEntityList	Array of DataEntityList		The list of data instances.
CategoryId	Long	133	The ID of the category.
Comment	String	category 1	The remarks of the category.
CreateTime	Long	1541576644000	The time when the category was created.
Depth	Integer	1	The number of category levels.
LastOperatorId	String	12345	The ID of the user who performed the last operation.
ModifiedTime	Long	1541576644000	The last time when the category was modified.
Name	String	category 1	The name of the category.
OwnerId	String	123	The ID of the category owner.

Parameter	Type	Example	Description
ParentCategoryId	Long	12	The ID of the parent category.
PageNum	Integer	1	The number of the page to return.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Long	10	The total number of categories.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaCategory
&ParentCategoryId=333
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>10</TotalCount>
  <PageNum>1</PageNum>
  <PageSize>10</PageSize>
  <DataEntityList>
    <CategoryId>133</CategoryId>
    <Comment>category 1</Comment>
    <ModifiedTime>1541576644000</ModifiedTime>
    <OwnerId>123</OwnerId>
    <LastOperatorId>12345</LastOperatorId>
    <CreateTime>1541576644000</CreateTime>
    <ParentCategoryId>12</ParentCategoryId>
    <Depth>1</Depth>
    <Name>category 1</Name>
  </DataEntityList>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 10,
    "PageNum": 1,
    "PageSize": 10,
    "DataEntityList": {
      "CategoryId": 133,
      "Comment": "category 1",
      "ModifiedTime": 1541576644000,
      "OwnerId": "123",
      "LastOperatorId": "12345",
      "CreateTime": 1541576644000,
      "ParentCategoryId": 12,
      "Depth": 1,
      "Name": "category 1"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalError.Meta.Unknown	An unknown error occurred while processing your request.	The error code returned because an unknown error has occurred.
403	Invalid.Meta.CategoryForbidden	Failed to delete the category. The current category is associated with a table and cannot be operated.	The error message returned because the category contains associated tables and cannot be deleted.
400	Invalid.Meta.CategoryIllegalDepth	A category can contain a maximum of 4 levels.	The error message returned because the category can contain a maximum of 4 levels.

HttpCode	Error code	Error message	Description
403	Invalid.Meta.CategoryIllegalTenant	The tenant ID is different in the parent and child categories.	The error message returned because the tenant ID is different in the parent and child categories.
400	Invalid.Meta.CategoryNameDuplicate	The category name is duplicated at this level.	The error message returned because the category name is duplicated at this level.
500	Invalid.Meta.CategoryRelationDuplicate"	The table is already bound to the category.	The error message returned because the category is already associated with the table.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.19. GetMetaTableListByCategory

Queries metatables in a specified category.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetMetaTableListByCategory	The operation that you want to perform.
CategoryId	Long	Yes	23	The ID of the class. You can call the GetMetaCategory to obtain the ID of the class.
RegionId	String	Yes	cn-shanghai	The region ID of the distributed instance.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The business data.
PageNumber	Integer	1	The number of the page to return.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TableGuidList	List	odps.engine_name.table_name	The list of metatables.
TotalCount	Long	20	The total number of returned entries.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetMetaTableListByCategory
&CategoryId=23
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TableGuidList>odps.engine_name.table_name</TableGuidList>
  <TotalCount>20</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": {
    "TableGuidList": "odps.engine_name.table_name",
    "TotalCount": 20,
    "PageSize": 10,
    "PageNumber": 1
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
500	InternalError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalError.Meta.TenantTimeout	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalError.Meta.Unknown	An unknown error occurred while processing your request.	The error code returned because an unknown error has occurred.
403	Invalid.Meta.CategoryForbidden	Failed to delete the category. The current category is associated with a table and cannot be operated.	The error message returned because the category contains associated tables and cannot be deleted.
400	Invalid.Meta.CategoryIllegalDepth	A category can contain a maximum of 4 levels.	The error message returned because the category can contain a maximum of 4 levels.

HttpCode	Error code	Error message	Description
403	Invalid.Meta.CategoryIllegalTenant	The tenant ID is different in the parent and child categories.	The error message returned because the tenant ID is different in the parent and child categories.
400	Invalid.Meta.CategoryNameDuplicate	The category name is duplicated at this level.	The error message returned because the category name is duplicated at this level.
500	Invalid.Meta.CategoryRelationDuplicate"	The table is already bound to the category.	The error message returned because the category is already associated with the table.
400	Invalid.Meta.TableNotExist	The table does not exist.	The error message returned because the metatable does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.
400	InvalidParameter.Meta.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.20. AddToMetaCategory

Adds a metatable to a specified category.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AddToMetaCategory	The operation that you want to perform.
CategoryId	Long	Yes	5	The ID of the category.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the metatable was added to the specified category.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=AddToMetaCategory
&CategoryId=5
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.21. DeleteFromMetaCategory

Removes a metatable from a specified category.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteFromMetaCategory	The operation that you want to perform.
CategoryId	Long	Yes	5	The ID of the category.
RegionId	String	Yes	cn-shanghai	The ID of the region where the streaming domain resides.
TableGuid	String	Yes	odps.engine_name.table_name	The GUID of the table.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the metatable was removed from the specified category.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0bc1ec92159376	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteFromMetaCategory
&CategoryId=5
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0bc1ec92159376</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0bc1ec92159376",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.22. CreateTable

Creates a MaxCompute table or view.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateTable	The operation that you want to perform.
Columns.N.ColumnName	String	Yes	columnName1	The field that is monitored.
Columns.N.ColumnType	String	Yes	string	The data type of the field.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
TableName	String	Yes	tableName1	The name of the table.
Columns.N.ColumnNameCn	String	No	Display name	The alias of the field.
Columns.N.SequenceNumber	Integer	No	1	The sequence number of the field. You can specify how fields are sorted in a table. By default, fields are sorted in the order requests are created. If the field is a partition field, this parameter is not supported.
Columns.N.Length	Integer	No	10	The length of the field.
Columns.N.IsPartitionCol	Boolean	No	0	Specifies whether the current field is a partition field. Valid values: 1: The current field is a partition field. 0: The current field is a non-partition field.

Parameter	Type	Required	Example	Description
Columns.N.Comment	String	No	The description of the manual review job.	The comment of the field.
IsView	Integer	No	0	Specifies whether to create a view or table. Valid values: 0: creates a table. 1: creates a view.
Visibility	Integer	No	1	Specifies whether the table or workspace is visible. Valid values: 0: Neither the table or workspace is visible. 1: The table and workspace are visible. 2: Only the workspace is visible.
LifeCycle	Integer	No	10	The lifecycle of the table. By default, this parameter is left empty, which indicates that the table is permanently stored.
CategoryId	Long	No	101	The ID of the category that can be associated. You can call the GetMetaCategory operation to query the ID of the category that can be associated.
LogicalLevelId	Long	No	101	The ID of the logical level.
PhysicsLevelId	Long	No	101	The ID of the physical level.
ExternalTableType	String	No	0	The storage type of the external table. Valid values: 0: OSS 1: Tablestore 2: Volume 3: MySQL
Location	String	No	location	The storage location of the external table.

Parameter	Type	Required	Example	Description
ProjectId	Long	No	23	The ID of the DataWorks workspace.
Endpoint	String	No	odps://abc	The endpoint of MaxCompute.
EnvType	Integer	No	1	The environment type of the DataWorks workspace. Valid values: 0: development environment 1: production environment
Themes.N.ThemeId	Long	No	101	The ID of the topic.
Themes.N.ThemeLevel	Integer	No	101	The level that corresponds to the topic ID. You can call the ListTableTheme operation to query the level.
AppGuid	String	No	odps.test	The ID of the MaxCompute project. The value of this parameter is in the odps.{projectName} format.
Comment	String	No	The description of the manual review job.	The comment.
HasPart	Integer	No	0	Specifies whether the MaxCompute table is a partitioned table. Valid values: 1 and 0. The value 1 indicates that the MaxCompute table is a partitioned table. The value 0 indicates that the MaxCompute table is not a partitioned table. This parameter is deprecated. Do not use this parameter. The code determines whether a MaxCompute table is a partitioned table based on the Column.N.isPartitionCol parameter. If the Column.N.isPartitionCol parameter is set to 1, the MaxCompute table is a partitioned table.

Parameter	Type	Required	Example	Description
ClientToken	String	No	Reserved parameter	A reserved parameter.

Response parameters

Parameter	Type	Example	Description
RequestId	String	abcdef	The ID of the request.
TaskInfo	Struct		The information about the request task. After a request task is submitted, it is divided into multiple subtasks that are run in sequence. After the current subtask is completed, the next subtask starts to run. After all subtasks are completed, the request task is completed. If a request task is aborted due to one of the following issues, address the issue based on the error code and initiate the request task again: • The request task fails to be submitted. • After the request task is submitted, a subtask fails to run.
Content	String	success	Details about the status of the current subtask. The following descriptions explain each value and provide examples: • If the current subtask succeeds, success is returned. • If the current subtask fails, the error details are displayed.
NextTaskId	String	abc1	The ID of the subtask to run. If this parameter is left empty, all subtasks are completed. You can call the GetDDLJobStatus operation to query the status of the subtask based on the subtask ID.

Parameter	Type	Example	Description
Status	String	success	The status of the current subtask. Valid values: • operating: The subtask is running. • success: The subtask succeeds. • failure: The subtask fails. For detailed error information, see the Content parameter.
TaskId	String	abc	The ID of the current subtask.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateTable
&Columns.1.ColumnName=columnName1
&Columns.1.ColumnType=string
&HasPart=0
&RegionId=cn-shanghai
&TableName=tableName1
&<Common request parameters>
```

Sample success responses

XML format

```
<TaskInfo>
  <Status>success</Status>
  <TaskId>abc</TaskId>
  <Content>success</Content>
  <NextTaskId>abc1</NextTaskId>
</TaskInfo>
<RequestId>abcdef</RequestId>
```

JSON format

```
{
  "TaskInfo": {
    "Status": "success",
    "TaskId": "abc",
    "Content": "success",
    "NextTaskId": "abc1"
  },
  "RequestId": "abcdef"
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.23. UpdateTable

Updates a MaxCompute table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateTable	The operation that you want to perform.
Columns.N.ColumnName	String	Yes	abc	The field that is monitored.

Parameter	Type	Required	Example	Description
Columns.N.ColumnType	String	Yes	string	The type of the field. For more information, see MaxCompute field types.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
TableName	String	Yes	abc	The name of the MaxCompute table.
Columns.N.ColumnNameCn	String	No	Display name	The display name of the field.
Columns.N.SequenceNumber	Integer	No	1	The sequence number of the field. If the field is a partition field, this parameter is not supported.
Columns.N.Length	Integer	No	10	The length of the field.
Columns.N.IsPartitionCol	Boolean	No	0	Specifies whether the field is a partition field. Valid values: 0 and 1. The value 0 indicates that the field is not a partition field. The value 1 indicates that the field is a partition field.
Columns.N.Comment	String	No	The description of the manual review job.	The comment of the field.
IsView	Integer	No	0	Specifies whether the table is a view. Valid values: 0 and 1. The value 0 indicates that the table is not a view. The value 1 indicates that the table is a view. The parameter is deprecated. Do not use this parameter.

Parameter	Type	Required	Example	Description
Visibility	Integer	No	1	The scope in which the table is visible. Valid values: 0, 1, and 2. The value 0 indicates that the table is not visible to all members. The value 1 indicates that the table is visible to all members. The value 2 indicates that the table is visible to workspace members.
LifeCycle	Integer	No	10	The lifecycle of the table. Unit: days. If this parameter is not specified, this table is permanently stored.
CategoryId	Long	No	101	The ID of the associated category.
LogicalLevelId	Long	No	101	The ID of the logical level.
PhysicsLevelId	Long	No	101	The ID of the physical level.
ExternalTableType	String	No	1	The type of the external table. Valid values: 0, 1, 2, and 3. The value 0 indicates the OSS external table. The value 1 indicates the Tablestore external table. The value 2 indicates the volume external table. The value 3 indicates the MySQL external table. The parameter is deprecated. Do not use this parameter.
Location	String	No	location	The storage location of the external table. The parameter is deprecated. Do not use this parameter.
ProjectId	Long	No	101	The ID of the DataWorks workspace. You can log on to the DataWorks console to obtain the ID.
Endpoint	String	No	odps://	The endpoint of MaxCompute. If this parameter is not specified, the endpoint of the MaxCompute project is automatically used.

Parameter	Type	Required	Example	Description
EnvType	Integer	No	1	The environment of the DataWorks workspace. Valid values: 0 and 1. The value 0 indicates the development environment. The value 1 indicates the production environment.
Themes.N.ThemeId	Long	No	101	The ID of the associated topic.
Themes.N.ThemeLevel	Integer	No	101	The level that corresponds to the topic ID.
AppGuid	String	No	odps.test	The globally unique identifier (GUID) of the MaxCompute project. The value of this parameter is in the odps.{projectName} format.
CreateIfNotExists	Boolean	No	true	Specifies whether the table exists. Valid values: - true: The table exists. - false: The table does not exist. The parameter is deprecated. Do not use this parameter.
HasPart	Integer	No	0	Specifies whether the table that you want to update is a partitioned table. Valid values: 0 and 1. The value 0 indicates that the table is not a partitioned table. The value 1 indicates that the table is a partitioned table. This parameter is deprecated. Do not use this parameter. The Column.N.isPartitionCol parameter is used to specify whether the MaxCompute table is a partitioned table. If the Column.N.isPartitionCol parameter is set to 1, the MaxCompute table is a partitioned table.
Comment	String	No	The description of the manual review job.	The comment.

Response parameters

Parameter	Type	Example	Description
RequestId	String	abcde	The ID of the request.
TaskInfo	Struct		The information about the request task. After a request task is submitted, it is divided into multiple subtasks that are run in sequence. After the current subtask is completed, the next subtask starts to run. After all subtasks are completed, the request task is completed. If a request task is aborted due to one of the following issues, address the issue based on the error code and initiate the request task again: • The request task fails to be submitted. • After the request task is submitted, a subtask fails to run.
Content	String	success	Details about the status of the current subtask. The following descriptions explain each value and provide examples: • If the current subtask succeeds, success is returned. • If the current subtask fails, the error details are displayed.
NextTaskId	String	abc1	The ID of the subtask to run. If this parameter is left empty, all subtasks are completed. You can call the <code>GetDDLJobStatus</code> operation to query the status of the subtask based on the subtask ID.
Status	String	success	The status of the current subtask. Valid values: • operating: The subtask is running. • success: The subtask succeeds. • failure: The subtask fails. For detailed error information, see the Content parameter.
TaskId	String	abc	The ID of the current subtask.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateTable
&Columns.1.ColumnName=abc
&Columns.1.ColumnType=string
&RegionId=cn-shanghai
&TableName=abc
&<Common request parameters>
```

Sample success responses

XML format

```
<TaskInfo>
  <Status>success</Status>
  <TaskId>abc</TaskId>
  <Content>success</Content>
  <NextTaskId>abc1</NextTaskId>
</TaskInfo>
<RequestId>abcde</RequestId>
```

JSON format

```
{
  "TaskInfo": {
    "Status": "success",
    "TaskId": "abc",
    "Content": "success",
    "NextTaskId": "abc1"
  },
  "RequestId": "abcde"
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.24. UpdateTableAddColumn

Updates the field data in a MaxCompute table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateTableAddColumn	The operation that you want to perform.
Column.N.Column Name	String	Yes	abc	The field that is monitored.
Column.N.Column Type	String	Yes	string	The type of the field. For more information, see MaxCompute field types.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint that is used to call the operation.
TableGuid	String	Yes	odps.engine_name.table_name	The globally unique identifier (GUID) of the MaxCompute table. Specify the GUID in the format of odps.projectName.tableName.

Parameter	Type	Required	Example	Description
Column.N.Column NameCn	String	No	Display name	The display name of the field.
Column.N.Comment	String	No	The description of the manual review job.	The comment of the field.

Response parameters

Parameter	Type	Example	Description
RequestId	String	abc	The ID of the request.
TaskInfo	Struct		The information about the request task. After a request task is submitted, it is divided into multiple subtasks that are run in sequence. After the current subtask is completed, the next subtask starts to run. After all subtasks are completed, the request task is completed. If a request task is aborted due to one of the following issues, address the issue based on the error code and initiate the request task again: - The request task fails to be submitted. - After the request task is submitted, a subtask fails to run.
Content	String	success	Details about the status of the current subtask. The following descriptions explain each value and provide examples: - If the current subtask succeeds, success is returned. - If the current subtask fails, the error details are displayed.
NextTaskId	String	abc1	The ID of the subtask to run. If this parameter is left empty, all subtasks are completed. You can call the GetDDLJobStatus operation to query the status of the subtask based on the subtask ID.

Parameter	Type	Example	Description
Status	String	success	The status of the current subtask. Valid values: • operating: The subtask is running. • success: The subtask succeeds. • failure: The subtask fails. For detailed error information, see the Content parameter.
TaskId	String	abc2	The ID of the current subtask.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateTableAddColumn
&Column.1.ColumnName=abc
&Column.1.ColumnType=string
&RegionId=cn-shanghai
&TableGuid=odps.engine_name.table_name
&<Common request parameters>
```

Sample success responses

XML format

```
<TaskInfo>
  <Status>success</Status>
  <TaskId>abc2</TaskId>
  <Content>success</Content>
  <NextTaskId>abc1</NextTaskId>
</TaskInfo>
<RequestId>abc</RequestId>
```

JSON format

```
{
  "TaskInfo": {
    "Status": "success",
    "TaskId": "abc2",
    "Content": "success",
    "NextTaskId": "abc1"
  },
  "RequestId": "abc"
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.25. DeleteTable

Deletes a MaxCompute table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteTable	The operation that you want to perform.

Parameter	Type	Required	Example	Description
EnvType	Integer	No	1	The environment of the DataWorks workspace. Valid values: 0 and 1. The value 0 indicates the development environment. The value 1 indicates the production environment.
ProjectId	Long	Yes	101	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TableName	String	Yes	table1	The name of the MaxCompute table.
AppGuid	String	No	odps.test	The globally unique identifier (GUID) of the MaxCompute project. The value of this parameter is in the odps.{projectName} format.

Response parameters

Parameter	Type	Example	Description
RequestId	String	abcde	The ID of the request.
TaskInfo	Struct		The information about the task of deleting a table.
Content	String	success	The content of the task.
NextTaskId	String	abc1	The ID of the running task.
Status	String	success	The status of the task that has ended.
TaskId	String	abc	The ID of the task that has ended.

Examples

Sample requests


```
http(s)://[Endpoint]/?Action=DeleteTable
&EnvType=1
&ProjectId=101
&RegionId=cn-shanghai
&TableName=table1
&<Common request parameters>
```

Sample success responses

XML format

```
<TaskInfo>
  <Status>success</Status>
  <TaskId>abc</TaskId>
  <Content>success</Content>
  <NextTaskId>abc1</NextTaskId>
</TaskInfo>
<RequestId>abcde</RequestId>
```

JSON format

```
{
  "TaskInfo": {
    "Status": "success",
    "TaskId": "abc",
    "Content": "success",
    "NextTaskId": "abc1"
  },
  "RequestId": "abcde"
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.26. GetDDLJobStatus

Queries the status of a table creation, update, or deletion task.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDDLJobStatus	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The region ID.
TaskId	String	Yes	abc	The ID of the DDL task.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the task.
Content	String	success	The content of the task.
NextTaskId	String	abc1	The ID of the ongoing task. If this parameter is empty, all tasks have ended.
Status	String	success	The status of the task.

Parameter	Type	Example	Description
TaskId	String	abc	The ID of the task.
RequestId	String	abc	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDDLJobStatus
&RegionId=cn-shanghai
&TaskId=abc
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abc</RequestId>
<Data>
  <Status>success</Status>
  <TaskId>abc</TaskId>
  <Content>success</Content>
  <NextTaskId>abc1</NextTaskId>
</Data>
```

JSON format

```
{
  "RequestId": "abc",
  "Data": {
    "Status": "success",
    "TaskId": "abc",
    "Content": "success",
    "NextTaskId": "abc1"
  }
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.27. UpdateMetaTable

Updates the metadata information about a table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateMetaTable	The operation that you want to perform.
EnvType	Integer	No	1	The environment of the DataWorks workspace. Valid values: 0 and 1. The value 0 indicates the development environment. The value 1 indicates the production environment.
ProjectId	Long	Yes	101	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.

Parameter	Type	Required	Example	Description
TableName	String	Yes	table1	The name of the table.
TableGuid	String	No	odps.test.table1	The unique identifier of the table. The format is odps.{projectName}.{tableName}.
NewOwnerId	String	No	12345	The new OwnerId, which is not updated when it is empty.
AddedLabels	String	No	a,b,c	The names of tags to be added. Separate them with commas (,).
RemovedLabels	String	No	a,b,c	The names of the tags to be deleted. Separate them with commas (,).
CategoryId	Long	No	101	The ID of the associated category.
Visibility	Integer	No	1	Edit the visibility of the table, including 0 (not visible), 1 (visible), and 2 (visible to the workspace).
Caption	String	No	Display name	The display name of the metatable.

Response parameters

Parameter	Type	Example	Description
RequestId	String	abcde	The ID of the request.
UpdateResult	Boolean	true	Indicates whether the subscriber is updated.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateMetaTable
&EnvType=1
&ProjectId=101
&RegionId=cn-shanghai
&TableName=table1
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcde</RequestId>
<UpdateResult>true</UpdateResult>
```

JSON format

```
{
  "RequestId": "abcde",
  "UpdateResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.28. UpdateMetaTableIntroWiki

Updates the instructions on how to use a table. If no instruction on how to use the table is available, the instructions that are specified by calling this operation are added.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateMetaTableIntroWiki	The operation that you want to perform.
Content	String	Yes	abc	The details of the table description.
RegionId	String	Yes	cn-shanghai	The region ID.
TableGuid	String	Yes	odps.test.table1	The unique identifier of the table. The format is odps.{projectName}.{tableName}.

Response parameters

Parameter	Type	Example	Description
RequestId	String	abcde	The ID of the request.
UpdateResult	Boolean	true	The result of the operation.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateMetaTableIntroWiki
&Content=abc
&RegionId=cn-shanghai
&TableGuid=odps.test.table1
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>abcde</RequestId>
<UpdateResult>true</UpdateResult>
```

JSON format

```
{
  "RequestId": "abcde",
  "UpdateResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.29. ListMetaDB

Queries metadatabases.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	ListMetaDB	The operation that you want to perform.
DataSourceType	String	Yes	emr	The type of the data source. Only E-MapReduce (EMR) is supported.
ProjectId	Long	Yes	123	The ID of the DataWorks workspace. You can call the ListProjects operation to obtain the ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
PageNum	Integer	No	1	The page number of the returned page.
PageSize	Integer	No	10	The number of the page to return. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
DatabaseInfo	Struct		The information of the metadatabases returned.
DbList	Array of DbList		The metadatabases returned.
CreateTimeStamp	Long	1388776825	The timestamp at which the metadatabase was created. This value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC. You can convert the timestamp to the date based on the time zone that you use.

Parameter	Type	Example	Description
Location	String	hdfs://localhost:7777/user/hadoop/test.txt	The URL of the metadatabase.
ModifiedTimeStamp	Long	1388776837	The timestamp at which the metadatabase was updated. This value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
Name	String	name	The name of the ApsaraDB RDS for MySQL database.
OwnerId	String	1232	The ID of the metadatabase owner.
Type	String	HIVE	The type of the metadatabase.
UUID	String	32342	The universally unique identifier (UUID) of the metadatabase.
TotalCount	Long	10	The total number of the metadatabases returned.
RequestId	String	abc	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListMetaDB
&DataSourceType=emr
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>abc</RequestId>
<DatabaseInfo>
  <TotalCount>10</TotalCount>
  <DbList>
    <Type>HIVE</Type>
    <OwnerId>1232</OwnerId>
    <ModifiedTimeStamp>1388776837</ModifiedTimeStamp>
    <UUID>32342</UUID>
    <CreateTimeStamp>1388776825</CreateTimeStamp>
    <Name>name</Name>
    <Location>hdfs://localhost:777/user/hadoop/test.txt</Location>
  </DbList>
</DatabaseInfo>
```

JSON format

```
{
  "RequestId": "abc",
  "DatabaseInfo": {
    "TotalCount": 10,
    "DbList": {
      "Type": "HIVE",
      "OwnerId": 1232,
      "ModifiedTimeStamp": 1388776837,
      "UUID": 32342,
      "CreateTimeStamp": 1388776825,
      "Name": "name",
      "Location": "hdfs://localhost:777/user/hadoop/test.txt"
    }
  }
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	InternalServerError.Meta.NoCalcEngine	An error occurred while retrieving the compute engine based on the MaxCompute project.	The error message returned because the system has failed to obtain a compute engine instance based on the specified MaxCompute project.
500	InternalServerError.Meta.TenantTimeOut	Failed to connect to the interface network.	The error message returned because the network cannot be connected.
500	InternalServerError.Meta.Unknown	An unknown error occurred while processing your request.	The error code returned because an unknown error has occurred.
403	Forbidden.Meta.NoTenant	The user is not the owner of any tenants.	The error message returned because the current user is not a tenant owner.
400	Invalid.Meta.AppNotExist	The MaxCompute project does not exist.	The error message returned because the MaxCompute project does not exist.
400	InvalidParameter.Meta.CommonError	The specified parameter is invalid.	The error message returned because the values specified for one or more request parameters are invalid.
400	InvalidParameter.Meta.Illegal	The app guid parameter is invalid.	The error message returned because the specified app GUID is invalid.
400	InvalidParameter.Meta.IllegalDate	The date format is invalid.	The error message returned because the date format is invalid.
400	InvalidParameter.Meta.IllegalPageError	The parameter for paging is invalid.	The error message returned because the value of the parameter for paging is invalid.

HttpCode	Error code	Error message	Description
400	InvalidParameter.Metadata.NotExist	The object does not exist.	The error message returned because the object you query does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.30. UpdateTableModelInfo

Updates the topic, level, and category of a table.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateTableModelInfo	The operation that you want to perform.
FirstLevelThemeId	Long	Yes	101	The ID of the level -1 topic.
LevelId	Long	Yes	101	The ID of the level.
RegionId	String	Yes	cn-shanghai	The region ID.
SecondLevelThemeId	Long	Yes	101	The ID of the second-level topic.
TableGuid	String	Yes	odps.test.table1	The unique identifier of the table, in the format of odps.{projectName}.{tableName}.
LevelType	Integer	No	1	The type of the level. Valid values: 1 and 2. The value 1 indicates the logical level. The value 2 indicates the physical level.

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
RequestId	String	abcde	The ID of the request.
UpdateResult	Boolean	true	Indicates whether the subscriber is updated.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateTableModelInfo
&FirstLevelThemeld=101
&LevelId=101
&RegionId=cn-shanghai
&SecondLevelThemeld=101
&TableGuid=odps.test.table1
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcde</RequestId>
<UpdateResult>true</UpdateResult>
```

JSON format

```
{
  "RequestId": "abcde",
  "UpdateResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.31. CreateTableTheme

Creates a topic.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateTableTheme	The operation that you want to perform.
Level	Integer	No	1	The level of the theme, including 1 (first-level theme) and 2 (second-level theme).
Name	String	Yes	The name of a topic.	The name of the subject.
ProjectId	Long	Yes	123	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
ParentId	Long	No	122	The ID of the parent class level.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	0	The error code returned.
ErrorMessage	String	success	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	ABCd	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.
Themeld	Long	123	The ID of the associated topic.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateTableTheme
&Level=1
&Name=Subject name
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Successful response examples

XML format

```
<Themeld>123</Themeld>
<RequestId>ABCd</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>0</ErrorCode>
<ErrorMessage>success</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "Themeld": 123,
  "RequestId": "ABCd",
  "HttpStatusCode": 200,
  "ErrorCode": 0,
  "ErrorMessage": "success",
  "Success": true
}
```


Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.32. DeleteTableTheme

Deletes a topic.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteTableTheme	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The region ID.

Parameter	Type	Required	Example	Description
Themeld	Long	Yes	123	The ID of the topic.
ProjectId	Long	No	123	The ID of the DataWorks workspace.

Response parameters

Parameter	Type	Example	Description
DeleteResult	Boolean	true	Indicates whether the sharding task was deleted.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	abcdef	The ID of the request.
Success	Boolean	true	Indicates whether the topic was deleted. Valid values: - true - false

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteTableTheme
&RegionId=cn-shanghai
&Themeld=123
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcdef</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>0</ErrorCode>
<ErrorMessage>success</ErrorMessage>
<Success>true</Success>
<DeleteResult>true</DeleteResult>
```

JSON format

```
{
  "RequestId": "abcdef",
  "HttpStatusCode": 200,
  "ErrorCode": 0,
  "ErrorMessage": "success",
  "Success": true,
  "DeleteResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HttpCode	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.33. UpdateTableTheme

Updates a topic.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateTableTheme	The operation that you want to perform.
Name	String	Yes	The name of a topic.	The name of the subject.
ProjectId	Long	Yes	123	The ID of the business process.
RegionId	String	Yes	cn-shanghai	The region ID.
ThemeId	Long	Yes	123	The ID of the topic.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	abcd	The ID of the request.

Parameter	Type	Example	Description
Success	Boolean	true	Whether the topic is updated successfully: • true • false
UpdateResult	Boolean	true	Indicates whether the update result is returned successfully: • true • false

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateTableTheme
&Name=Subject name
&ProjectId=123
&RegionId=cn-shanghai
&ThemeId=123
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcd</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
<UpdateResult>true</UpdateResult>
```

JSON format

```
{
  "RequestId": "abcd",
  "HttpStatusCode": 200,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true,
  "UpdateResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.34. ListTableTheme

Queries topics.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListTableTheme	The operation that you want to perform.
ProjectId	Long	Yes	123	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.

Parameter	Type	Required	Example	Description
ParentId	Long	No	121	The ID of the parent class.
PageNum	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
Data	Struct		Hierarchical data.
ThemeList	Array of ThemeList		The list of table levels.
CreateTimeStamp	Long	123432343243	The time when the hierarchy was created.
Creator	String	123455	The creator of the hierarchy.
Level	Integer	1	1 (level 1) and 2 (level 2).
Name	String	Layer	The name of the level.
ParentId	Long	122	The ID of the parent node.
ProjectId	Long	123	The ID of the business process.
Themeld	Long	123	The ID of the topic.
TotalCount	Long	10	The total number of content.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	abcde	The ID of the request.
Success	Boolean	true	Indicates whether the topic was successfully queried.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListTableTheme
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcde</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <ThemeList>
    <ParentId>122</ParentId>
    <ThemeId>123</ThemeId>
    <ProjectId>123</ProjectId>
    <Level>1</Level>
    <Creator>123455</Creator>
    <CreateTimeStamp>123432343243</CreateTimeStamp>
    <Name> Level name </Name>
  </ThemeList>
  <TotalCount>10</TotalCount>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format


```
{
  "RequestId": "abcde",
  "HttpStatusCode": 200,
  "Data": {
    "ThemeList": {
      "ParentId": 122,
      "Themeld": 123,
      "ProjectId": 123,
      "Level": 1,
      "Creator": 123455,
      "CreateTimeStamp": 123432343243,
      "Name": "Level Name"
    },
    "TotalCount": 10
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HttpCode	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.35. CreateTableLevel

Creates a table level.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateTableLevel	The operation that you want to perform.
LevelType	Integer	Yes	1	The type of the level. Valid values: 1 and 2. The value 1 indicates the logical level. The value 2 indicates the physical level.
Name	String	Yes	Level 1	The name of the level.
ProjectId	Long	Yes	123	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
Description	String	No	Level description	The description of the level.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.

Parameter	Type	Example	Description
LevelId	Long	123	The ID of the level.
RequestId	String	Request ID	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateTableLevel
&LevelType=1
&Name=Level 1
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>Request ID</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
<LevelId>123</LevelId>
```

JSON format

```
{
  "RequestId": "Request ID",
  "HttpStatusCode": 200,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true,
  "LevelId": 123
}
```

Error codes

HTTP status code	Error code	Error message	Description

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.2.36. DeleteTableLevel

Deletes a table level.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteTableLevel	The operation that you want to perform.
LevelId	Long	Yes	123	The ID of the level to be deleted. You can obtain it from the ListLevel interface.

Parameter	Type	Required	Example	Description
ProjectId	Long	Yes	123	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.

Response parameters

Parameter	Type	Example	Description
DeleteResult	Boolean	true	Indicates whether the sharding task was deleted.
RequestId	String	abcde	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteTableLevel
&LevelId=123
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcde</RequestId>
<DeleteResult>true</DeleteResult>
```

JSON format

```
{
  "RequestId": "abcde",
  "DeleteResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.37. UpdateTableLevel

Updates a table level.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateTableLevel	The operation that you want to perform.
LevelId	Long	Yes	123	The ID of the level to be updated. You can obtain it through ListLevel.
LevelType	Integer	No	1	The type of the level. Valid values: 1 and 2. The value 1 indicates the logical level. The value 2 indicates the physical level.

Parameter	Type	Required	Example	Description
Name	String	Yes	Layer	The name of the level.
ProjectId	Long	Yes	123	The ID of the DataWorks workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
Description	String	No	Level description	The description of the level.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	abcde	The ID of the request.
Success	Boolean	true	Indicates whether the category was updated.
UpdateResult	Boolean	true	Indicates whether the result is returned successfully.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateTableLevel
&LevelId=123
&LevelType=1
&Name=Level 1
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Successful response examples

XML format

```
<RequestId>abcde</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
<UpdateResult>true</UpdateResult>
```

JSON format

```
{
  "RequestId": "abcde",
  "HttpStatusCode": 200,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true,
  "UpdateResult": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HttpCode	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.2.38. ListTableLevel

Queries table levels.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListTableLevel	The operation that you want to perform.
LevelType	Integer	No	1	The type of the level. Valid values: 1 and 2. The value 1 indicates the logical level. The value 2 indicates the physical level.
ProjectId	Long	Yes	123	The ID of the DataWorks workspace. You can log on to the DataWorks console to obtain the ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
PageNum	Integer	No	1	The number of the page to return. Default value: 1.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	abc	The ID of the request.
Success	Boolean	true	Indicates whether the list of table levels is returned.
TableLevelInfo	Struct		The table level.
LevelList	Array of LevelList		The list of table levels.
Description	String	Description	The description of the table level.
LevelId	Long	123	The ID of the table level.
LevelType	Integer	1	The type of the level. Valid values: 1 and 2. The value 1 indicates the logical level. The value 2 indicates the physical level.
Name	String	Level 1	The name of the level.
ProjectId	Long	123	The ID of the DataWorks workspace.
TotalCount	Long	10	The total number of levels.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListTableLevel
&LevelType=1
&ProjectId=123
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<TableLevelInfo>
  <TotalCount>10</TotalCount>
  <LevelList>
    <Description> Hierarchy description </Description>
    <LevelType>1</LevelType>
    <ProjectId>123</ProjectId>
    <LevelId>123</LevelId>
    <Name> Level 1</Name>
  </LevelList>
</TableLevelInfo>
<RequestId>abc</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>>true</Success>
```

JSON format

```
{
  "TableLevelInfo": {
    "TotalCount": 10,
    "LevelList": {
      "Description": "Level description",
      "LevelType": 1,
      "ProjectId": 123,
      "LevelId": 123,
      "Name": "Level 1"
    }
  },
  "RequestId": "abc",
  "HttpStatusCode": 200,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3. DataStudio

2.1.6.3.1. CreateFile

Creates a file in DataStudio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateFile	The operation that you want to perform.
AutoRerunTimes	Integer	Yes	3	The number of automatic reruns that are allowed after an error occurs. Maximum value: 10.
FileDescription	String	Yes	File description	The description of the file.
FileFolderPath	String	Yes	Workflow/1/Data Integration/Folder 1/Folder 2	The path of the file.

Parameter	Type	Required	Example	Description
FileName	String	Yes	File name	The name of the file.
FileType	Integer	Yes	10	The type of the code in the file. Valid values: 6 (Shell), 10 (ODPS SQL), 11 (ODPS MR), 23 (Data Integration), 24 (ODPS Script), 99 (zero load), 221 (PyODPS 2), 225 (ODPS Spark), 227 (EMR Hive), 228 (EMR Spark), 229 (EMR Spark SQL), 230 (EMR MR), 239 (OSS object inspection), 257 (EMR Shell), 258 (EMR Spark Shell), 259 (EMR Presto), 260 (EMR Impala), 900 (real-time synchronization), 1089 (cross-tenant collaboration), 1091 (Hologres development), 1093 (Hologres SQL), 1100 (assignment), and 1221 (PyODPS 3).
InputList	String	Yes	project_root,project.file1,project.001_out	The output names of the parent file on which the current file depends. This parameter corresponds to the Parent Node Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console. Separate multiple output names with commas (,).
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Parameter	Type	Required	Example	Description
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
Owner	String	No	10000000000001	The ID of the Alibaba Cloud account used by the file owner. If this parameter is not specified, the ID of the Alibaba Cloud account of the user who calls the operation is automatically used.
Content	String	No	SHOW TABLES;	The code content of the file. The code formats vary based on the code type specified by FileType. To view the code format for a specific code type, go to Operation Center, right-click a node of the code type, and then select View Code.
AutoRerunInterval Millis	Integer	No	120000	The interval between automatic reruns after an error occurs. Unit: milliseconds. Maximum value: 1800000 (30 minutes). This parameter corresponds to the Auto Rerun Interval upon Error parameter in the Schedule section of the Properties tab in the DataWorks console. The interval that you specify in the DataWorks console is measured in minutes. Pay attention to the conversion between the units of time when you call the operation.

Parameter	Type	Required	Example	Description
RerunMode	String	No	ALL_ALLOWED	Specifies whether the node can be rerun. Valid values: ALL_ALLOWED, FAILURE_ALLOWED, and ALL_DENIED. The value ALL_ALLOWED indicates that the node can be rerun regardless of whether the last running is successful. The value FAILURE_ALLOWED indicates that the node can be rerun only if the last running fails. The value ALL_DENIED indicates that the node cannot be rerun. This parameter corresponds to the Rerun parameter in the Schedule section of the Properties tab in the DataWorks console.
Stop	Boolean	No	false	Specifies whether the scheduling for the node is suspended. This parameter corresponds to the Skip Execution parameter in the Schedule section of the Properties tab in the DataWorks console.
ParaValue	String	No	a=x b=y	The scheduling parameters of the node. This parameter corresponds to the Parameters parameter in the General section of the Properties tab in the DataWorks console.
StartEffectDate	Long	No	936923400000	The start time of automatic scheduling. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC. This parameter corresponds to the Validity Period parameter in the Schedule section of the Properties tab in the DataWorks console.

Parameter	Type	Required	Example	Description
EndEffectDate	Long	No	4155787800000	The end time of automatic scheduling. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC. This parameter corresponds to the Validity Period parameter in the Schedule section of the Properties tab in the DataWorks console.

Parameter	Type	Required	Example	Description
CronExpress	String	No	00 05 00 * * ?	The cron expression that represents the automatic scheduling policy of the node. This parameter corresponds to the Cron Expression parameter in the Schedule section of the Properties tab in the DataWorks console. - Cron expression for a node that is scheduled to run at 05:30 every day: <code>00 30 05 * * ?</code> - Cron expression for a node that is scheduled to run at the fifteenth minute of each hour: <code>00 15 * * * ?</code> - Cron expression for a node that is scheduled to run every 10 minutes: <code>00 00/10 * * * ?</code> - Cron expression for a node that is scheduled to run every 10 minutes from 08:00 to 17:00 every day: <code>00 00-59/10 8-23 * * * ?</code> - Cron expression for a node that is scheduled to run at 00:20 on the first day of each month: <code>00 20 00 1 * ?</code> - Cron expression for a node that is scheduled to run every three months starting from 00:10 on January 1: <code>0 10 00 1 1-12/3 ?</code> - Cron expression for a node that is scheduled to run at 00:05 every Tuesday and Friday: <code>00 05 00 * * 2,5</code> The scheduling system of DataWorks imposes the following limits on cron expressions: - A node can be scheduled to run at a minimum interval of 5 minutes. - A node can be scheduled to run at 00:05 every day at the earliest.

Parameter	Type	Required	Example	Description
CycleType	String	No	DAY	The type of the scheduling cycle. Valid values: NOT_DAY and DAY. The value NOT_DAY indicates that the node is scheduled to run by minute or hour. The value DAY indicates that the node is scheduled to run by day, week, or month. This parameter corresponds to the Scheduling Cycle parameter in the Schedule section of the Properties tab in the DataWorks console.
DependentType	String	No	NONE	The way in which the node depends on last-cycle instances. Valid values: NONE, SELF, CHILD, and USER_DEFINE. The value NONE indicates that the node does not depend on last-cycle instances. The value SELF indicates that the node depends on its own last-cycle instance. The value CHILD indicates that the node depends on the last-cycle instances of its descendant nodes. The value USER_DEFINE indicates that the node depends on the last-cycle instances of specific nodes.
DependentNodeIDList	String	No	abc	The IDs of the nodes that generate instances in the last cycle on which the current node depends.
ProjectIdentifier	String	No	dw_project	The unique identifier of the DataWorks workspace. You can view the identifier at the top of the DataStudio page. You can also click the identifier to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Parameter	Type	Required	Example	Description
ResourceGroupIdentifier	String	No	group_375827434852437	The identifier of the resource group that is used to run the node. This parameter corresponds to the Resource Group parameter in the Resource Group section of the Properties tab in the DataWorks console. You can call the ListResourceGroups operation to query the available resource groups in the workspace. When you call the operation, set ResourceType to 1. The response parameter Identifier indicates the identifier of an available resource group.
ResourceGroupId	Long	No	375827434852437	The ID of the resource group that is used to run the node. This parameter corresponds to the Resource Group parameter in the Resource Group section of the Properties tab in the DataWorks console. You must specify either this parameter or the ResourceGroupIdentifier parameter to determine the resource group. You can call the ListResourceGroups operation to query the available resource groups in the workspace. When you call the operation, set ResourceType to 1. The response parameter Id indicates the ID of an available resource group.
ConnectionName	String	No	odps_first	The name of the connected data source that is used to run the node. This parameter corresponds to the Connection parameter in the Source column of the Connections section on the configuration tab of the node. You can call the ListDataSources operation to obtain the available data sources of the workspace. The response parameter Name indicates the name of an available data source.

Response parameters

Parameter	Type	Example	Description
Data	Long	1000001	The ID of the file that was created.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateFile
&AutoRerunTimes=3
&FileDescription=File description
&FileFolderPath=Workflow/1/Data Integration/Folder 1/Folder 2
&FileName=File name
&FileType=10
&InputList=project_root,project.file1,project.001_out
&ProjectId=10000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>1000001</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG",
  "HttpStatusCode": 200,
  "Data": 1000001,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

Only default resource groups for Data Integration can be used to run the batch synchronization node that corresponds to the file created by using the CreateFile operation. You cannot specify a resource group for Data Integration for the node by setting the related parameter.

2.1.6.3.2. CreateUdfFile

Creates a file for a function in DataStudio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateUdfFile	The operation that you want to perform.
ClassName	String	Yes	com.alibaba.dataworks.api.udf.StringConcat	The name of the class where the function is defined. This parameter corresponds to the Class Name parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
FileFolderPath	String	Yes	Workflow/1/Function/String Processing Function	The path of the folder where the file for the function is stored.
FileName	String	Yes	StringConcat	The name of the file for the function.
FunctionType	String	Yes	STRING	The type of the function. Valid values: MATH, AGGREGATE, STRING, DATE, ANALYTIC, and OTHER. This parameter corresponds to the Function Type parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
RegionId	String	Yes	cn-zhangjiakou	The region ID.
Resources	String	Yes	string-concat-1.0.0.jar,commons-lang-2.6.jar	The names of the resources that are referenced by the function. This parameter corresponds to the Resources parameter in the Register Function section of the configuration tab of the function in the DataWorks console. Separate multiple resource names with commas (,).
UdfDescription	String	Yes	Concatenate multiple strings to generate a new string	The description of the function. This parameter corresponds to the Description parameter in the Register Function section of the configuration tab of the function in the DataWorks console.

Parameter	Type	Required	Example	Description
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
CmdDescription	String	No	StringConcat(String... substrs)	The expression syntax for calling the function. This parameter corresponds to the Expression Syntax parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
ParameterDescription	String	No	Strings to be concatenated	The description of the input parameters of the function. This parameter corresponds to the Parameter Description parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
ReturnValue	String	No	New string generated after all strings are concatenated in the order of appearance	The description for the return value of the function. This parameter corresponds to the Return Value parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
Example	String	No	StringConcat('a', 'b', 'c')	The example for calling the function. This parameter corresponds to the Example parameter in the Register Function section of the configuration tab of the function in the DataWorks console.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw_project	The unique identifier of the DataWorks workspace. You can view the identifier at the top of the DataStudio page. You can also click the identifier to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	Long	100000002	The ID of the file that was created.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateUdfFile
&ClassName=com.alibaba.dataworks.api.udf.StringConcat
&FileFolderPath=Workflow/1/Function/String Processing Function
&FileName=StringConcat
&FunctionType=STRING
&ProjectId=10000
&RegionId=cn-zhangjiakou
&Resources=string-concat-1.0.0.jar,commons-lang-2.6.jar
&UdfDescription=Concatenate multiple strings to generate a new string
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>100000002</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": 100000002,
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.3. CreateBusiness

Creates a workflow in DataStudio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateBusiness	The operation that you want to perform. Set the value to CreateBusiness.
BusinessName	String	Yes	My workflow	The name of the workflow. The name must be unique in a workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
ProjectIdentifier	String	No	dw_project	The unique English identifier of the DataWorks workspace. The identifier is displayed in the top navigation bar of the DataStudio page. You can click the identifier to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.
Description	String	No	A workflow created by using the operation	The description of the workflow.

Parameter	Type	Required	Example	Description
Owner	String	No	10000000000001	The ID of the Alibaba Cloud account used by the owner of the workflow. If this parameter is not specified, the ID of the Alibaba Cloud account of the user who calls the operation is used by default.
UseType	String	No	NORMAL	The functional module to which the workflow belongs. Valid values: NORMAL and MANUAL_BIZ.

Response parameters

Parameter	Type	Example	Description
BusinessId	Long	100001	The ID of the workflow.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

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

Sample success responses

JSON format

```
{"RequestId":"0000-ABCD-EFG****","HttpStatusCode":"200","ErrorCode":"Invalid.Tenant.ConnectionNotExists","ErrorMessage":"The connection does not exist.","BusinessId":"100001","Success":"true"}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.4. DeleteBusiness

Deletes a workflow.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteBusiness	The operation that you want to perform.
BusinessId	Long	Yes	1000001	The ID of the workflow. You can call the ListBusiness operation to query the ID.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteBusiness
&BusinessId=1000001
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.5. CreateFolder

Creates a folder.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateFolder	The operation that you want to perform.
FolderPath	String	Yes	Business Flow/systemData/MaxCompute/import	The path to the folder. You can refer to the "Directory structure on the DataStudio page after a folder is created" section in Examples of DataStudio API operation calls to create a path.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	String	bdfd68****	The unique identifier of the folder.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateFolder
&FolderPath=Business Flow/System Data/MaxCompute/Import
&RegionId=cn-shanghai
&<Common request parameters>
```


Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>bdfd68****</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG",
  "HttpStatusCode": 200,
  "Data": "bdfd68****",
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.6. UpdateBusiness

Updates a workflow.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateBusiness	The operation that you want to perform.
BusinessId	Long	Yes	300000	The ID of the workflow. You can call the ListBusiness operation to query the ID.
BusinessName	String	Yes	MyBusiness	The name of the workflow. You can call the ListBusiness operation to obtain the data.
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or the ProjectId parameter to determine the DataWorks workspace to which the operation is applied.

Parameter	Type	Required	Example	Description
Description	String	No	modified from my first business	The description of the workflow.
Owner	String	No	348428****	The owner of the workflow. You can call the ListBusiness operation to obtain the data.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateBusiness
&BusinessId=300000
&BusinessName=MyBusiness
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.7. ListFiles

Queries files.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListFiles	The operation that you want to perform.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
FileFolderPath	String	No	Workflow/My first workflow/MaxCompute/ods	The directory in which the files are stored.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.
Keyword	String	No	ods	The keyword in the file names. The keyword is used to perform a fuzzy match.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or the ProjectId parameter to determine the DataWorks workspace to which the operation is applied.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of the page to return. Default value: 10. Maximum value: 100.

Parameter	Type	Required	Example	Description
UseType	String	No	NORMAL	The functional module to which the files belong. Valid values: NORMAL, MANUAL, MANUAL_BIZ, SKIP, ADHOCQUERY, and COMPONENT. The value SKIP indicates that the nodes corresponding to the files are run in dry-run mode.
FileTypes	String	No	10,23	The types of the code in the files. Valid values: 6 (Shell), 10 (ODPS SQL), 11 (ODPS MR), 23 (Data Integration), 24 (ODPS Script), 97 (PAI), 98 (node group), 99 (zero load), 221 (PyODPS 2), 225 (ODPS Spark), 227 (EMR Hive), 228 (EMR Spark), 229 (EMR Spark SQL), 230 (EMR MR), 239 (OSS object inspection), 257 (EMR Shell), 258 (EMR Spark Shell), 259 (EMR Presto), 260 (EMR Impala), 900 (real-time synchronization), 1002 (PAI), 1089 (cross-tenant collaboration), 1091 (Hologres development), 1093 (Hologres SQL), 1100 (assignment), 1106 (for-each), and 1221 (PyODPS 3).
Owner	String	No	3726346****	The owner of the files.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The files returned.
Files	Array of File		The details of the file.
BizId	Long	300000	The ID of the workflow to which the file belongs. This parameter is deprecated and replaced by the BusinessId parameter.

Parameter	Type	Example	Description
BusinessId	Long	300000	The ID of the workflow where the file resides.
CommitStatus	Integer	1	Indicates whether the latest code is committed. Valid values: 0 and 1. A value of 0 indicates that the latest code is not committed. A value of 1 indicates that the latest code is committed.
ConnectionName	String	odps_first	The ID of the compute engine instance that is used to run the node corresponding to the file.
Content	String	SHOW TABLES;	The code in the file.
CreateTime	Long	1593950832000	The time when the file was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CreateUser	String	382762****	The Alibaba Cloud account ID of the user who created the file.
CurrentVersion	Integer	2	The version of the file.
FileDescription	String	my test datastudio file	The description of the file.
FileFolderId	String	2735c2****	The ID of the folder to which the file belongs.
FileId	Long	10000001	The ID of the file.
FileName	String	ods_user_info_d	The name of the file.

Parameter	Type	Example	Description
FileType	Integer	10	The type of the code in the file. Valid values: 6 (Shell), 10 (ODPS SQL), 11 (ODPS MR), 23 (Data Integration), 24 (ODPS Script), 99 (zero load), 221 (PyODPS 2), 225 (ODPS Spark), 227 (EMR Hive), 228 (EMR Spark), 229 (EMR Spark SQL), 230 (EMR MR), 239 (OSS object inspection), 257 (EMR Shell), 258 (EMR Spark Shell), 259 (EMR Presto), 260 (EMR Impala), 900 (real-time synchronization), 1089 (cross-tenant collaboration), 1091 (Hologres development), 1093 (Hologres SQL), 1100 (assignment), and 1221 (PyODPS 3).
IsMaxCompute	Boolean	false	Indicates whether the file needs to be uploaded to MaxCompute if the file is a MaxCompute resource file. This parameter is returned only if the file is a MaxCompute resource file.
LastEditTime	Long	1593950832000	The time when the file was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
LastEditUser	String	38748246285727	The ID of the Alibaba Cloud account used to last modify the file.
NodeId	Long	300001	The ID of the auto triggered node that is generated in the scheduling system after the file is committed.
Owner	String	3872572****	The ID of the Alibaba Cloud account used by the file owner.
ParentId	Long	-1	The ID of the node group file to which the current file belongs. This parameter is returned only if the current file is an inner file of the node group file.
UseType	String	NORMAL	The functional module to which the file belongs. Valid values: NORMAL, MANUAL, MANUAL_BIZ, SKIP, ADHOCQUERY, and COMPONENT. The value SKIP indicates that the node corresponding to the file is run in dry-run mode.

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
TotalCount	Integer	13	The total number of entries returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListFiles
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>13</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Files>
    <ParentId>-1</ParentId>
    <Owner>3872572****</Owner>
    <CurrentVersion>2</CurrentVersion>
    <UseType>NORMAL</UseType>
    <FileName>ods_user_info_d</FileName>
    <CreateTime>1593950832000</CreateTime>
    <CreateUser>382762****</CreateUser>
    <IsMaxCompute>false</IsMaxCompute>
    <ConnectionName>odps_first</ConnectionName>
    <LastEditTime>1593950832000</LastEditTime>
    <FileDescription>my test datastudio file</FileDescription>
    <FileType>10</FileType>
    <LastEditUser>38748246285727</LastEditUser>
    <Content>SHOW TABLES;</Content>
    <CommitStatus>1</CommitStatus>
    <NodeId>300001</NodeId>
    <FileId>10000001</FileId>
    <FileFolderId>2735c2****</FileFolderId>
    <BizId>300000</BizId>
    <BusinessId>300000</BusinessId>
  </Files>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 13,
    "PageSize": 10,
    "PageNumber": 1,
    "Files": {
      "ParentId": -1,
      "Owner": "3872572****",
      "CurrentVersion": 2,
      "UseType": "NORMAL",
      "FileName": "ods_user_info_d",
      "CreateTime": 1593950832000,
      "CreateUser": "382762****",
      "IsMaxCompute": false,
      "ConnectionName": "odps_first",
      "LastEditTime": 1593950832000,
      "FileDescription": "my test datastudio file",
      "FileType": 10,
      "LastEditUser": 38748246285727,
      "Content": "SHOW TABLES;",
      "CommitStatus": 1,
      "NodeId": 300001,
      "FileId": 10000001,
      "FileFolderId": "2735c2****",
      "BizId": 300000,
      "BusinessId": 300000
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.

HttpCode	Error code	Error message	Description
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.8. EstablishRelationTableToBusiness

Imports a table to a workflow. To perform this operation in the DataWorks console, click a workflow, right-click Table under MaxCompute, and then select Import Table on the DataStudio page.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	EstablishRelationTableToBusiness	The operation that you want to perform.
BusinessId	String	Yes	1000001	The ID of the workflow. You can call the ListBusiness operation to query the workflow ID.
FolderId	String	Yes	2eb6f9****	The ID of the folder. You can call the GetFolder or ListFolders operation to query the folder ID.
RegionId	String	Yes	cn-zhangjiakou	The region ID.
TableGuid	String	Yes	odps.dw_project.tb1	The universally unique identifier (UUID) of the table. You can call the SearchMetaTables operation to query the UUID of the table.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw	The GUID of the DataWorks workspace. You can view the GUID at the top of the DataStudio page. You can also click the GUID to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=EstablishRelationTableToBusiness
&BusinessId=1000001
&FolderId=2eb6f9****
&ProjectId=10000
&RegionId=cn-zhangjiakou
&TableGuid=odps.dw_project.tb1
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG",
  "HttpStatusCode": 200,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.9. SubmitFile

Commits a file to the development environment of the scheduling system to generate a node. This API operation processes your request in asynchronous mode. You can call the GetDeployment operation to check whether the file is committed.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SubmitFile	The operation that you want to perform.
FileId	Long	Yes	1000000	The ID of the file. You can call the ListFiles operation to query the ID.
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.
Comment	String	No	Commit for the first time	The description of the commit operation.

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
Data	Long	3000001	The ID of the deployment task. The ID is used as the value of a specific request parameter when you call the GetDeployment operation to query the details of the deployment task.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=SubmitFile
&FileId=1000000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>3000001</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format


```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": 3000001,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.10. ListBusiness

Queries workflows.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	ListBusiness	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.
Keyword	String	No	my	The keyword of the workflow that is used to perform a fuzzy match.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The workflows returned.

Parameter	Type	Example	Description
Business	Array of Business		The details of the workflow.
BusinessId	Long	3000001	The ID of the workflow.
BusinessName	String	1	The name of the workflow.
Description	String	1	The description of the workflow.
Owner	String	34824327****	The owner of the workflow.
ProjectId	Long	10000	The ID of the workspace to which the workflow belongs.
UseType	String	NORMAL	The functional module to which the workflow belongs. Valid values: NORMAL and MANUAL_BIZ.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
TotalCount	Integer	13	The total number of entries returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListBusiness
&PageNumber=1
&PageSize=10
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>13</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Business>
    <Owner>34824327****</Owner>
    <Description>1</Description>
    <BusinessName>1</BusinessName>
    <UseType>NORMAL</UseType>
    <ProjectId>10000</ProjectId>
    <BusinessId>3000001</BusinessId>
  </Business>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 13,
    "PageSize": 10,
    "PageNumber": 1,
    "Business": {
      "Owner": "34824327****",
      "Description": "1",
      "BusinessName": "1",
      "UseType": "NORMAL",
      "ProjectId": 10000,
      "BusinessId": 3000001
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.11. GetFileVersion

Queries the details of a file version.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetFileVersion	The operation that you want to perform.
FileId	Long	Yes	1000001	The ID of the file. You can call the ListFiles operation to query the file ID.
FileVersion	Integer	No	2	The version of the file that you want to query.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The region ID.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
ProjectIdentifier	String	No	dw_project	The GUID of the DataWorks workspace. You can view the GUID at the top of the DataStudio page. You can also click the GUID to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the file version.
ChangeType	String	UPDATE	The type of the change to the file of the current version. Valid values: CREATE, UPDATE, and DELETE.
Comment	String	Second version	The description of the file version.
CommitTime	Long	1593881265000	The time when the file version was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CommitUser	String	7384234****	The Alibaba Cloud account ID of the user who created the file version.
FileContent	String	SHOW TABLES;	The code in the file of the current version.
FileName	String	ods_user_info_d	The name of the file of the current version.

Parameter	Type	Example	Description
FilePropertyContent	String	<code>{"fileName":"ods_user_info_d","fileType":10}</code>	The basic information about the file of the current version.
FileVersion	Integer	2	The version of the file.
IsCurrentProd	Boolean	true	Indicates whether the file version is the same as the latest file version in the production environment.
NodeContent	String	<code>{"cycleType":0,"cronExpress":"00 05 00 * * ?"}</code>	The scheduling configurations for the node corresponding to the file of the current version.
NodeId	Long	3000001	The ID of the node corresponding to the file of the current version.
Status	String	COMMITTED	The status of the file of the current version. Valid values: COMMITTING, COMMITTED, CHECK_OK, PACKAGED, DEPLOYING, DEPLOYED, and CANCELLED.
UseType	String	NORMAL	The feature to which the file applies. Features include data analytics (NORMAL), manually triggered node (MANUAL), manually triggered workflow (MANUAL_BIZ), dry-run (SKIP), ad hoc query (ADHOCQUERY), and script template management (COMPONENT).
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetFileVersion
&FileId=1000001
&FileVersion=2
&ProjectId=10000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Status>COMMITTED</Status>
  <Comment>Second version</Comment>
  <CommitUser>7384234****</CommitUser>
  <FilePropertyContent>{"fileName":"ods_user_info_d","fileType":10}</FilePropertyContent>
  <FileName>ods_user_info_d</FileName>
  <CommitTime>1593881265000</CommitTime>
  <UseType>NORMAL</UseType>
  <IsCurrentProd>true</IsCurrentProd>
  <FileContent>SHOW TABLES;</FileContent>
  <ChangeType>UPDATE</ChangeType>
  <FileVersion>2</FileVersion>
  <NodeId>3000001</NodeId>
  <NodeContent>{"cycleType":0,"cronExpress":"00 05 00 * * ?"}</NodeContent>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format


```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "Status": "COMMITTED",
    "Comment": "Second version submission",
    "CommitUser": "7384234****",
    "FilePropertyContent": "{\"fileName\":\"ods_user_info_d\",\"fileType\":\"10\"}",
    "FileName": "ods_user_info_d",
    "CommitTime": 1593881265000,
    "UseType": "NORMAL",
    "IsCurrentProd": true,
    "FileContent": "SHOW TABLES;",
    "ChangeType": "UPDATE",
    "FileVersion": 2,
    "NodeId": 3000001,
    "NodeContent": "{\"cycleType\":\"0\",\"cronExpress\":\"00 05 00 * * ?\"}"
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HttpCode	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.12. DeleteFolder

Deletes a folder from DataStudio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteFolder	The operation that you want to perform.
FolderId	String	Yes	2eb6f9****	The ID of the folder. You can call the ListFolders operation to query the folder ID.
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteFolder
&FolderId=2eb6f9****
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.13. DeleteFile

Deletes a file from DataStudio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteFile	The operation that you want to perform.
FileId	Long	Yes	10000001	The ID of the file. You can call the ListFiles operation to query the file ID.
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteFile
&FileId=10000001
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.14. GetDeployment

Queries the details of a deployment task.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDeployment	The operation that you want to perform.
DeploymentId	Long	Yes	3000001	The ID of the deployment task. A deployment task ID is generated when you perform the SubmitFile or DeployFile operation.
RegionId	String	Yes	cn-zhangjiakou	The region ID.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw_project	The GUID of the DataWorks workspace. You can view the GUID at the top of the DataStudio page. You can also click the GUID to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the release package.
Deployment	Struct		The details of the release package.
CreateTime	Long	1593877765000	The time when the deployment task was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CreatorId	String	20030****	The Alibaba Cloud account ID of the user who created the deployment task.
ErrorMessage	String	success	The error message that was returned when the deployment task failed. In this case, the value of the Status parameter is 2.
ExecuteTime	Long	1593877765000	The time when the deployment task was run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
FromEnvironment	Integer	0	The environment in which the deployment task was run. Valid values: 0 and 1. A value of 0 indicates the on-premises environment. A value of 1 indicates the development environment.

Parameter	Type	Example	Description
HandlerId	String	2003****	The Alibaba Cloud account ID of the user who ran the deployment task.
Name	String	ods_user_info_d-2020-07-04_20030****	The name of the deployment task. The value is the same as the name of the specific deployment task that is displayed on the Release Package page in the Deploy module.
Status	Integer	1	The status of the deployment task. Valid values: 0, 1, and 2. A value of 0 indicates that the task is ready. A value of 1 indicates that the task was successful. A value of 2 indicates that the task failed.
ToEnvironment	Integer	1	The environment to which the file was committed. Valid values: 1 and 2. A value of 1 indicates the development environment. A value of 2 indicates the production environment.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDeployment
&DeploymentId=3000001
&ProjectId=10000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Deployment>
    <Status>1</Status>
    <HandlerId>2003****</HandlerId>
    <ExecuteTime>1593877765000</ExecuteTime>
    <CreatorId>20030****</CreatorId>
    <CreateTime>1593877765000</CreateTime>
    <ToEnvironment>1</ToEnvironment>
    <FromEnvironment>0</FromEnvironment>
    <ErrorMessage>success</ErrorMessage>
    <Name>ods_user_info_d-2020-07-04_20030****</Name>
  </Deployment>
</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "Deployment": {
      "Status": 1,
      "HandlerId": "2003****",
      "ExecuteTime": 1593877765000,
      "CreatorId": "20030****",
      "CreateTime": 1593877765000,
      "ToEnvironment": 1,
      "FromEnvironment": 0,
      "ErrorMessage": "Success",
      "Name": "ods_user_info_d-2020-07-04_20030****"
    }
  },
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.15. UpdateFile

Updates a file.

When you debug or call this operation, the specified parameters must be different from the original file configuration of the node. For example, if the value of a parameter of the source node is A, you need to change it to B before submitting it. If you still enter A, invalid is reported.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateFile	The operation that you want to perform.
AutoRerunTimes	Integer	No	3	The maximum number of automatic reruns that are allowed after an error occurs.
FileDescription	String	Yes	File description	The description of the file.
FileFolderPath	String	Yes	Workflow/1/Data Integration/Folder 1/Folder 2	The path of the file.
FileId	Long	Yes	100000001	The ID of the file.

Parameter	Type	Required	Example	Description
FileName	String	Yes	ods_user_info_d	The name of the file. You can set the FileName parameter to a new value to change the file name. You can call the ListFiles operation to query the ID of the file whose name you want to change. Then, you can set the Field parameter to the ID and set FileName to a new value when you call the UpdateFile operation.
InputList	String	Yes	project_root,project.file1,project.001_out	The output name of the parent file on which the current file depends. This parameter corresponds to the Parent Node Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console. If you specify multiple output names, separate them with commas (,).
RegionId	String	Yes	cn-zhangjiakou	The region ID.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
Content	String	No	SELECT "1";	The code content of the file. The code formats vary based on the file type. To view the code format for a specific file type, go to Operation Center, right-click a node of the file type, and then select View Code.

Parameter	Type	Required	Example	Description
AutoRerunInterval Millis	Integer	No	120000	The interval between automatic reruns after an error occurs. Unit: milliseconds. This parameter corresponds to the Error automatic rerun interval parameter in the Schedule section of the Properties tab in the DataWorks console. The interval that you specify in the DataWorks console is measured in minutes. Pay attention to the conversion between the units of time when you call the operation.
RerunMode	String	No	ALL_ALLOWED	Specifies whether the node can be rerun. Valid values: ALL_ALLOWED, FAILURE_ALLOWED, and ALL_DENIED. The value ALL_ALLOWED indicates that the node can be rerun regardless of whether the last running is successful. The value FAILURE_ALLOWED indicates that the node can be rerun only if the last running fails. The value ALL_DENIED indicates that the node cannot be rerun. This parameter corresponds to the Rerun parameter in the Schedule section of the Properties tab in the DataWorks console.
Stop	Boolean	No	false	Specifies whether the scheduling for the node is suspended. This parameter corresponds to the Skip Execution parameter in the Schedule section of the Properties tab in the DataWorks console.

Parameter	Type	Required	Example	Description
ParaValue	String	No	x=a y=b z=c	The scheduling parameters of the node. This parameter corresponds to the Arguments parameter in the General section of the Properties tab in the DataWorks console.
StartEffectDate	Long	No	936923400000	The start time of automatic scheduling. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC. This parameter corresponds to the Validity Period parameter in the Schedule section of the Properties tab in the DataWorks console.
EndEffectDate	Long	No	4155787800000	The end time of automatic scheduling. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC. This parameter corresponds to the Validity Period parameter in the Schedule section of the Properties tab in the DataWorks console.

Parameter	Type	Required	Example	Description
CronExpress	String	No	00 00-59/5 1-23 * * ?	The cron expression that represents the automatic scheduling policy of the node. This parameter corresponds to the Cron Expression parameter in the Schedule section of the Properties tab in the DataWorks console. - Cron expression for a node that is scheduled to run at 05:30 every day: <code>00 30 05 * * ?</code> - Cron expression for a node that is scheduled to run at the fifteenth minute of each hour: <code>00 15 * * * ?</code> - Cron expression for a node that is scheduled to run every 10 minutes: <code>00 00/10 * * * ?</code> - Cron expression for a node that is scheduled to run every 10 minutes from 08:00 to 17:00 every day: <code>00 00-59/10 8-23 * * * ?</code> - Cron expression for a node that is scheduled to run at 00:20 on the first day of each month: <code>00 20 00 1 * ?</code> - Cron expression for a node that is scheduled to run every three months starting from 00:10 on January 1: <code>0 10 00 1 1-12/3 ?</code> - Cron expression for a node that is scheduled to run at 00:05 every Tuesday and Friday: <code>00 05 00 * * 2,5</code> The scheduling system of DataWorks imposes the following limits on cron expressions: - A node can be scheduled to run at a minimum interval of 5 minutes. - A node can be scheduled to run at 00:05 every day at the earliest.

Parameter	Type	Required	Example	Description
CycleType	String	No	NOT_DAY	The type of the scheduling cycle. Valid values: NOT_DAY and DAY. The value NOT_DAY indicates that the node is scheduled to run by minute or hour. The value DAY indicates that the node is scheduled to run by day, week, or month. This parameter corresponds to the Scheduling Cycle parameter in the Schedule section of the Properties tab in the DataWorks console.
DependentType	String	No	USER_DEFINE	The way in which the node depends on last-cycle instances. Valid values: NONE, SELF, CHILD, and USER_DEFINE. The value NONE indicates that the node does not depend on last-cycle instances. The value SELF indicates that the node depends on its own last-cycle instance. The value CHILD indicates that the node depends on the last-cycle instances of its descendant nodes. The value USER_DEFINE indicates that the node depends on the last-cycle instances of specific nodes.
DependentNodeList	String	No	5,10,15,20	The IDs of the auto triggered nodes on which the current node depends. Separate multiple IDs with commas (,). Specify this parameter when the DependentType parameter is set to USER_DEFINE. This parameter corresponds to the field that appears after Cross-Cycle Dependencies is selected and the Depend On parameter is set to Instances of Custom Nodes in the DataWorks console.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw_project	The GUID of the DataWorks workspace. You can view the GUID at the top of the DataStudio page. You can also click the GUID to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.
OutputList	String	No	dw_project.ods_user_info_d	The output name of the current file. This parameter corresponds to the Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console. If you specify multiple output names, separate them with commas (,).
ResourceGroupIdentifier	String	No	default_group	The name of the resource group that is used to run the node corresponding to the file. You can call the ListResourceGroups operation and set the resourceGroupType parameter to 1 to query available resource groups.
ConnectionName	String	No	odps_first	The name of the data source that is used to run the node corresponding to the file. You can call the ListConnections operation to query available data sources.
Owner	String	No	18023848927592	The ID of the file owner.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.

Parameter	Type	Example	Description
RequestId	String	0000-ABCD-EFGH-IJKLMNOPQ	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateFile
&AutoRerunTimes=3
&FileDescription=File description
&FileFolderPath=Business Flow/1/Data Integration/Folder1/Folder2
&FileId=100000001
&FileName=ods_user_info_d
&InputList=project_root,project.file1,project.001_out
&ProjectId=10000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFGH-IJKLMNOPQ</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFGH-IJKLMNOPQ",
  "HttpStatusCode": 200,
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HttpCode	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.16. GetFolder

Queries the details of a folder.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetFolder	The operation that you want to perform.
FolderId	String	Yes	273****	The ID of the folder. You can call the ListFolders operation to query the ID. You must specify either this parameter or FolderPath.
FolderPath	String	Yes	Workflow/1/Max Compute/ods	The path of the folder. You can call the ListFolders operation to query the path. You must specify either this parameter or FolderId.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the folder.
FolderId	String	2735****	The ID of the folder.
FolderPath	String	Workflow/1/MaxCompute/ods	The path of the folder.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.

Parameter	Type	Example	Description
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetFolder
&FolderId=273****
&FolderPath=Workflow/1/MaxCompute/ods
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <FolderPath>Workflow/1/MaxCompute/ods</FolderPath>
  <FolderId>2735****</FolderId>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "FolderPath": "Workflow/1/MaxCompute/ods",
    "FolderId": "2735****"
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.17. ListFileVersions

Queries file versions.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListFileVersions	The operation that you want to perform.
FileId	Long	Yes	100000001	The ID of the file. You can call the ListFiles operation to query the ID.

Parameter	Type	Required	Example	Description
ProjectId	Long	Yes	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
RegionId	String	Yes	cn-zhangjiakou	The region ID.
ProjectIdentifier	String	No	dw_project	The GUID of the DataWorks workspace. You can view the GUID at the top of the DataStudio page. You can also click the GUID to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.
PageNumber	Integer	No	1	The number of the page to return.
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The file versions returned.
FileVersions	Array of FileVersion		The details of the file version.
ChangeType	String	UPDATE	The type of the change to the file of the current version. Valid values: CREATE, UPDATE, and DELETE.
Comment	String	Second version	The description of the file version.

Parameter	Type	Example	Description
CommitTime	Long	1593881265000	The time when the file version was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CommitUser	String	73842342****	The ID of the Alibaba Cloud account used to create the file of the current version.
FileContent	String	SHOW TABLES;	The code in the file of the current version.
FileName	String	ods_user_info_d	The name of the file of the current version.
FilePropertyContent	String	{"fileName": "ods_user_info_d", "fileType": 10}	The basic information about the file of the current version.
FileVersion	Integer	2	The version of the file.
IsCurrentProd	Boolean	false	Indicates whether the file version is the same as the latest file version in the production environment.
NodeContent	String	{"cycleType": 0, "cronExpress": "00 05 00 * * ?"}	The scheduling configurations for the node that corresponds to the file of the current version.
NodeId	Long	1234	The ID of the auto triggered node that corresponds to the file of the current version.
Status	String	COMMITTED	The status of the file of the current version. Valid values: COMMITTING, COMMITTED, CHECK_OK, PACKAGED, DEPLOYING, DEPLOYED, and CANCELLED.
UseType	String	NORMAL	The functional module to which the file belongs. Valid values: NORMAL, MANUAL, MANUAL_BIZ, SKIP, ADHOCQUERY, and COMPONENT. The value SKIP indicates that the node corresponding to the file is run in dry-run mode.

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
TotalCount	Integer	13	The total number of entries that meet the query conditions.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListFileVersions
&FileId=100000001
&ProjectId=10000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>0000-ABCD-EFG****Q</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>13</TotalCount>
  <FileVersions>
    <Status>COMMITTED</Status>
    <CommitUser>73842342****</CommitUser>
    <Comment>Second version</Comment>
    <CommitTime>1593881265000</CommitTime>
    <FileName>ods_user_info_d</FileName>
    <FilePropertyContent>{"fileName":"ods_user_info_d","fileType":10}</FilePropertyContent>
    <UseType>NORMAL</UseType>
    <IsCurrentProd>false</IsCurrentProd>
    <FileContent>SHOW TABLES;</FileContent>
    <ChangeType>UPDATE</ChangeType>
    <FileVersion>2</FileVersion>
    <NodeId>3000001</NodeId>
    <NodeContent>{"cycleType":0,"cronExpress":"00 05 00 * * ?"}</NodeContent>
  </FileVersions>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
</Data>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****Q",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 13,
    "FileVersions": {
      "Status": "COMMITTED",
      "CommitUser": "73842342****",
      "Comment": "Second version submission",
      "CommitTime": 1593881265000,
      "FileName": "ods_user_info_d",
      "FilePropertyContent": "{\"fileName\":\"ods_user_info_d\",\"fileType\":10}",
      "UseType": "NORMAL",
      "IsCurrentProd": false,
      "FileContent": "SHOW TABLES;",
      "ChangeType": "UPDATE",
      "FileVersion": 2,
      "NodeId": 3000001,
      "NodeContent": "{\"cycleType\":0,\"cronExpress\":\"00 05 00 * * ?\"}"
    },
    "PageSize": 10,
    "PageNumber": 1
  },
  "Success": true
}

```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

2.1.6.3.18. GetFile

Queries the details of a file.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetFile	The operation that you want to perform.
ProjectId	Long	Yes	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or the ProjectId parameter to determine the DataWorks workspace to which the operation is applied.
FileId	Long	No	100000001	The ID of the file. You can call the ListFiles operation to query the ID.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the file.
File	Struct		The basic information of the file.
BizId	Long	1000001	The ID of the workflow to which the file belongs. This parameter is deprecated and replaced by the BusinessId parameter.
CommitStatus	Integer	0	Indicates whether the latest code in the file is committed. Valid values: 0 and 1. The value 0 indicates that the latest code in the file is not committed. The value 1 indicates that the latest code in the file is committed.
ConnectionName	String	odps_first	The ID of the compute engine instance that is used to run the node that corresponds to the file.

Parameter	Type	Example	Description
Content	String	SHOW TABLES;	The code in the file.
CreateTime	Long	1593879116000	The time when the file was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
CreateUser	String	424732****	The ID of the Alibaba Cloud account used to create the file.
CurrentVersion	Integer	3	The latest version number of the file.
FileDescription	String	My first DataWorks file	The description of the file.
FileFolderId	String	2735c2****	The ID of the folder to which the file belongs.
FileName	String	ods_user_info_d	The name of the file.
FileType	Integer	10	The type of the code in the file. Valid values: 6 (Shell), 10 (ODPS SQL), 11 (ODPS MR), 23 (Data Integration), 24 (ODPS Script), 99 (zero load), 221 (PyODPS 2), 225 (ODPS Spark), 227 (EMR Hive), 228 (EMR Spark), 229 (EMR Spark SQL), 230 (EMR MR), 239 (OSS object inspection), 257 (EMR Shell), 258 (EMR Spark Shell), 259 (EMR Presto), 260 (EMR Impala), 900 (real-time synchronization), 1089 (cross-tenant collaboration), 1091 (Hologres development), 1093 (Hologres SQL), 1100 (assignment), and 1221 (PyODPS 3).
IsMaxCompute	Boolean	true	Indicates whether the file needs to be uploaded to MaxCompute. This parameter is returned only if the file is a MaxCompute resource file.

Parameter	Type	Example	Description
LastEditTime	Long	1593879116000	The time when the file was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
LastEditUser	String	62465892****	The ID of the Alibaba Cloud account used to last modify the file.
NodeId	Long	300001	The ID of the auto triggered node that is generated in the scheduling system after the file is committed.
Owner	String	7775674356****	The ID of the Alibaba Cloud account used by the file owner.
ParentId	Long	-1	The ID of the node group file to which the current file belongs. This parameter is returned only if the current file is an inner file of the node group file.
UseType	String	0	The functional module to which the file belongs. Valid values: NORMAL, MANUAL, MANUAL_BIZ, SKIP, ADHOCQUERY, and COMPONENT. The value SKIP indicates that the node corresponding to the file is run in dry-run mode.
NodeConfiguration	Struct		The scheduling configurations of the file.
AutoRerunIntervalMillis	Integer	120000	The interval between automatic reruns after an error occurs. Unit: milliseconds. This parameter corresponds to the Auto Rerun Interval upon Error parameter in the Schedule section of the Properties tab in the DataWorks console. The interval that you specify in the DataWorks console is measured in minutes. Pay attention to the conversion between the units of time when you call the operation.

Parameter	Type	Example	Description
AutoRerunTimes	Integer	3	The maximum number of automatic reruns that are allowed after an error occurs.
CronExpress	String	00 05 00 * * ?	The cron expression that represents the time-based scheduling policy of the node.
CycleType	String	DAY	The type of the scheduling cycle. Valid values: NOT_DAY and DAY. This parameter corresponds to the Scheduling Cycle parameter in the Schedule section of the Properties tab in the DataWorks console.
DependentNodeList	String	5,10,15,20	The IDs of the nodes on which the node corresponding to the file depends when the DependentType parameter is set to USER_DEFINE. Multiple IDs are separated by commas (,). This parameter corresponds to the field that appears after Dependency on Last Cycle is selected and the Depend On parameter is set to Instances of Custom Nodes in the Schedule section of the Properties tab in the DataWorks console.
DependentType	String	USER_DEFINE	The way in which the node that corresponds to the file depends on last-cycle instances. Valid values: NONE, SELF, CHILD, and USER_DEFINE. The value NONE indicates that the node does not depend on last-cycle instances. The value SELF indicates that the node depends on its own last-cycle instance. The value CHILD indicates that the node depends on the last-cycle instances of its descendant nodes. The value USER_DEFINE indicates that the node depends on the last-cycle instances of specific nodes.
EndEffectDate	Long	4155787800000	The end time of automatic scheduling. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC. This parameter corresponds to the Validity Period parameter in the Schedule section of the Properties tab in the DataWorks console.

Parameter	Type	Example	Description
InputList	Array of NodeInputOutput		The output names of the parent files on which the current file depends. This parameter corresponds to the Parent Node Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console.
Input	String	project.001_out	The output name of the parent file on which the current file depends. This parameter corresponds to the Parent Node Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console.
OutputList	Array of NodeInputOutput		The output names of the current file. This parameter corresponds to the Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console.
Output	String	dw_project.002_out	The output name of the current file. This parameter corresponds to the Output Name parameter in the Dependencies section of the Properties tab in the DataWorks console.
RefTableName	String	ods_user_info_d	The output name of the current file. This parameter corresponds to the Output Table Name parameter in the Dependencies section of the Properties tab in the DataWorks console.
ParaValue	String	a=x b=y	The scheduling parameters. This parameter corresponds to the Parameters parameter in the General section of the Properties tab in the DataWorks console.

Parameter	Type	Example	Description
RerunMode	String	ALL_ALLOWED	Indicates whether the node can be rerun. Valid values: ALL_ALLOWED, FAILURE_ALLOWED, and ALL_DENIED. The value ALL_ALLOWED indicates that the node can be rerun regardless of whether the last running is successful. The value FAILURE_ALLOWED indicates that the node can be rerun only if the last running fails. The value ALL_DENIED indicates that the node cannot be rerun. This parameter corresponds to the Rerun parameter in the Schedule section of the Properties tab in the DataWorks console.
ResourceGroupId	Long	375827434852437	The ID of the resource group that is used to run the node. You can call the ListResourceGroups operation to query the available resource groups in the workspace.
StartEffectDate	Long	936923400000	The start time of automatic scheduling. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC. This parameter corresponds to the Validity Period parameter in the Schedule section of the Properties tab in the DataWorks console.
Stop	Boolean	false	Indicates whether the scheduling for the node that corresponds to the file is suspended. This parameter corresponds to the Skip Execution parameter in the Schedule section of the Properties tab in the DataWorks console.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetFile
&ProjectId=10000
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <File>
    <ParentId>-1</ParentId>
    <Owner>7775674356****</Owner>
    <CurrentVersion>3</CurrentVersion>
    <UseType>0</UseType>
    <FileName>ods_user_info_d</FileName>
    <CreateTime>1593879116000</CreateTime>
    <CreateUser>424732****</CreateUser>
    <IsMaxCompute>true</IsMaxCompute>
    <ConnectionName>odps_first</ConnectionName>
    <LastEditTime>1593879116000</LastEditTime>
    <FileDescription>My first DataWorks file</FileDescription>
    <FileType>10</FileType>
    <LastEditUser>62465892****</LastEditUser>
    <Content>SHOW TABLES;</Content>
    <CommitStatus>0</CommitStatus>
    <NodeId>300001</NodeId>
    <FileFolderId>2735c2****</FileFolderId>
    <BizId>1000001</BizId>
  </File>
  <NodeConfiguration>
    <EndEffectDate>4155787800000</EndEffectDate>
    <CycleType>DAY</CycleType>
    <ResourceGroupId>375827434852437</ResourceGroupId>
    <DependentNodeIdList>5,10,15,20</DependentNodeIdList>
    <AutoRerunIntervalMillis>120000</AutoRerunIntervalMillis>
    <StartEffectDate>936923400000</StartEffectDate>
    <Stop>false</Stop>
    <DependentType>USER_DEFINE</DependentType>
    <AutoRerunTimes>3</AutoRerunTimes>
    <RerunMode>ALL_ALLOWED</RerunMode>
    <ParaValue>a=x b=y</ParaValue>
    <CronExpress>00 05 00 * * ?</CronExpress>
    <InputList>
      <Input>project.001_out</Input>
    </InputList>
    <OutputList>
      <Output>dw_project.002_out</Output>
      <RefTableName>ods_user_info_d</RefTableName>
    </OutputList>
  </NodeConfiguration>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "File": {
      "ParentId": -1,
      "Owner": "7775674356****",
      "CurrentVersion": 3,
      "UseType": 0,
      "FileName": "ods_user_info_d",
      "CreateTime": 1593879116000,
      "CreateUser": "424732****",
      "IsMaxCompute": true,
      "ConnectionName": "odps_first",
      "LastEditTime": 1593879116000,
      "FileDescription": "My First Dataworks File",
      "FileType": 10,
      "LastEditUser": "62465892****",
      "Content": "SHOW TABLES;",
      "CommitStatus": 0,
      "NodeId": 300001,
      "FileFolderId": "2735c2****",
      "BizId": 1000001
    },
    "NodeConfiguration": {
      "EndEffectDate": 4155787800000,
      "CycleType": "DAY",
      "ResourceGroupId": 375827434852437,
      "DependentNodeIdList": "5,10,15,20",
      "AutoRerunIntervalMillis": 120000,
      "StartEffectDate": 936923400000,
      "Stop": false,
      "DependentType": "USER_DEFINE",
      "AutoRerunTimes": 3,
      "RerunMode": "ALL_ALLOWED",
      "ParaValue": "a=x b=y",
      "CronExpress": "00 05 00 * * ?",
      "InputList": {
        "Input": "project.001_out"
      },
      "OutputList": {
        "Output": "dw_project.002_out",
        "RefTableName": "ods_user_info_d"
      }
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}

```

Error codes

HttpCode	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.19. ListFolders

Queries folders.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListFolders	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ParentFolderPath	String	Yes	Workflow/1/Max Compute	The path of the parent folder.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The information of the folders.
Folders	Array of Folders		The folders returned.
FolderId	String	2735c2****	The ID of the folder.
FolderPath	String	Workflow/1/MaxCompute/ods	The path of the folder.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.

Parameter	Type	Example	Description
TotalCount	Integer	13	The total number of entries returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListFolders
&PageNumber=1
&PageSize=10
&ParentFolderPath=Workflow/1/MaxCompute
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>13</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Folders>
    <FolderPath>Workflow/1/MaxCompute/ods</FolderPath>
    <FolderId>2735c2****</FolderId>
  </Folders>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 13,
    "PageSize": 10,
    "PageNumber": 1,
    "Folders": {
      "FolderPath": "Workflow/1/MaxCompute/ods",
      "FolderId": "2735c2****"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.20. DeployFile

Deploys a file to the production environment.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeployFile	The operation that you want to perform.
FileId	Long	Yes	10000001	The ID of the file. You can call the ListFiles operation to query the ID. You must specify either this parameter or NodeId.
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.
Comment	String	No	Deploy for the first time	The description of the commit operation.

Parameter	Type	Required	Example	Description
NodeId	Long	No	2000001	The ID of the node in the scheduling system that corresponds to the file to be deployed. You must specify either this parameter or FileId.

Response parameters

Parameter	Type	Example	Description
Data	Long	30000001	The ID of the deployment task. The ID is used as the value of a specific request parameter when you call the GetDeployment operation to query the details of the deployment task.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeployFile
&FileId=10000001
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>30000001</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": 30000001,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HttpCode	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.3.21. UpdateUdfFile

Updates the file information about a function.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateUdfFile	The operation that you want to perform.
ClassName	String	Yes	com.alibaba.dataworks.api.udf.StringConcat	The name of the class where the function is defined. This parameter corresponds to the Class Name parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
FileFolderPath	String	Yes	Workflow/1/Function/String Processing Function	The path of the folder where the function is stored.
FileId	String	Yes	10000001	The ID of the file.
FunctionType	String	Yes	STRING	The type of the function. Valid values: MATH, AGGREGATE, STRING, DATE, ANALYTIC, and OTHER. This parameter corresponds to the Function Type parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
RegionId	String	Yes	cn-zhangjaikou	The region ID.
Resources	String	Yes	string-concat-1.0.0.jar,commons-lang-2.6.jar	The names of the resources that are referenced by the function. This parameter corresponds to the Resources parameter in the Register Function section of the configuration tab of the function in the DataWorks console. Separate multiple resource names with commas (,).

Parameter	Type	Required	Example	Description
UdfDescription	String	Yes	Concatenate multiple strings to generate a new string	The description of the function. This parameter corresponds to the Description parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can click the Workspace Manage icon in the upper-right corner of the DataStudio page to go to the Workspace Management page and view the workspace ID.
CmdDescription	String	No	StringConcat(String... substrs)	The expression syntax for calling the function. This parameter corresponds to the Expression Syntax parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
ParameterDescription	String	No	Strings to be concatenated	The description of the input parameters of the function. This parameter corresponds to the Parameter Description parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
ReturnValue	String	No	New string generated after all strings are concatenated in the order of appearance	The description for the return value of the function. This parameter corresponds to the Return Value parameter in the Register Function section of the configuration tab of the function in the DataWorks console.
Example	String	No	StringConcat('a', 'b', 'c')	The example for calling the function. This parameter corresponds to the Example parameter in the Register Function section of the configuration tab of the function in the DataWorks console.

Parameter	Type	Required	Example	Description
ProjectIdentifier	String	No	dw_project	The unique identifier of the DataWorks workspace. You can view the identifier at the top of the DataStudio page. You can also click the identifier to switch to another workspace. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```

http(s)://[Endpoint]/?Action=UpdateUdfFile
&ClassName=com.alibaba.dataworks.api.udf.StringConcat
&FileFolderPath=Workflow/1/Function/String Processing Function
&FileId=10000001
&FunctionType=STRING
&ProjectId=10000
&RegionId=cn-zhangjaikou
&Resources=string-concat-1.0.0.jar,commons-lang-2.6.jar
&UdfDescription=Concatenate multiple strings to generate a new string
&<Common request parameters>

```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.22. GetBusiness

Queries the details of a workflow.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetBusiness	The operation that you want to perform.
BusinessId	Long	Yes	1000000001	The ID of the workflow. You can call the ListBusiness operation to query the ID.
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the workflow.
BusinessId	Long	1000001	The ID of the workflow.
BusinessName	String	1	The name of the workflow.

Parameter	Type	Example	Description
Description	String	My first workflow	The description of the workflow.
Owner	String	20000****	The ID of the Alibaba Cloud account used by the workflow owner.
ProjectId	String	10000	The ID of the workspace to which the workflow belongs.
UseType	String	NORMAL	The functional module to which the workflow belongs. Valid values: NORMAL and MANUAL_BIZ.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetBusiness
&BusinessId=1000000001
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Owner>20000****</Owner>
  <Description>My first workflow</Description>
  <BusinessName>1</BusinessName>
  <UseType>NORMAL</UseType>
  <ProjectId>10000</ProjectId>
  <BusinessId>1000001</BusinessId>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "Owner": "20000****",
    "Description": "My first workflow",
    "BusinessName": "1",
    "UseType": "NORMAL",
    "ProjectId": 10000,
    "BusinessId": 1000001
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.

HTTP status code	Error code	Error message	Description
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.3.23. UpdateFolder

Updates the information about a folder.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateFolder	The operation that you want to perform.
FolderId	String	Yes	2735c2c19d58	The ID of the folder. You can call the ListFolders operation to query the ID.
FolderName	String	Yes	MySecondFolder	The name of the folder.
RegionId	String	Yes	cn-zhangjiakou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Parameter	Type	Required	Example	Description
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the ID. You must specify either this parameter or ProjectIdentifier to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the name. You must specify either this parameter or ProjectId to determine the DataWorks workspace to which the operation is applied.

Response parameters

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateFolder
&FolderId=2735c2c19d58
&FolderName=MySecondFolder
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.4. Operation Center

2.1.6.4.1. ListAlertMessages

Queries alerts.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListAlertMessages	The operation that you want to perform.
BeginTime	String	Yes	2020-04-02T00:00:00+0800	The beginning of the time range to report an alert. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
EndTime	String	Yes	2020-04-04T00:00:00+0800	The end of the time range to report an alert. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
PageNumber	Integer	Yes	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RemindId	Long	Yes	9527	The ID of the custom alert rule. This parameter takes effect if the AlertRuleTypes parameter is set to USER_DEFINE.

Parameter	Type	Required	Example	Description
AlertMethods	String	No	SMS,MAIL,PHONE	The notification method. Valid values: MAIL, SMS, and PHONE. A value of MAIL indicates that the notification is sent by email. A value of SMS indicates that the notification is sent by text message. A value of PHONE indicates that the notification is sent by call, which is supported only by DataWorks Professional Edition and more advanced editions. You can specify multiple notification methods. Separate them with commas (,).
AlertUser	String	No	123456	The ID of the Apsara Stack tenant account used by the alert recipient.
AlertRuleTypes	String	No	GLOBAL,USER_DEFINE,OTHER	The type of the alert rule. Valid values: GLOBAL, USER_DEFINE, and OTHER. A value of GLOBAL indicates that the alert rule is a global alert rule. A value of USER_DEFINE indicates that the alert rule is customized by a user. A value of OTHER indicates that the alert rule is a rule of another type. You can specify multiple types. Separate them with commas (,).

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned alerts.
AlertMessages	Array of AlertMessages	N/A	The list of alerts.
AlertId	Long	1234	The ID of the alert.

Parameter	Type	Example	Description
AlertMessageStatus	String	READY_TO_SEND	The sending status of the notification. Valid values: READY_TO_SEND, SEND_FAIL, SEND_SUCCESS, and SEND_OVERLIMIT. A value of READY_TO_SEND indicates that the notification is waiting to be sent. A value of SEND_FAIL indicates that the notification fails to be sent. A value of SEND_SUCCESS indicates that the notification is sent. A value of SEND_OVERLIMIT indicates that the number of notifications that are sent exceeds the upper limit.
AlertMethod	String	SMS	The notification method. Valid values: MAIL, SMS, and PHONE. A value of MAIL indicates that the notification is sent by email. A value of SMS indicates that the notification is sent by text message. A value of PHONE indicates that the notification is sent by call, which is supported only by DataWorks Professional Edition and more advanced editions.
AlertTime	Long	1553531401000	The timestamp when the alert was reported.
AlertUser	String	9527952795****	The ID of the Apsara Stack tenant account used by the alert recipient.
Content	String	Errors on a node	The content of the alert.
Instances	Array of Instances	N/A	The instances that triggered the custom alert rule. This parameter is returned if the Source parameter is set to REMIND_ALERT. This parameter is left empty if the Source parameter is set to other values.
InstanceId	Long	12345	The ID of the instance.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.

Parameter	Type	Example	Description
Status	String	NOT_RUN	The status of the instance. Valid values: NOT_RUN, WAIT_TIME, WAIT_RESOURCE, RUNNING, CHECKING, CHECKING_CONDITION, FAILURE, and SUCCESS. A value of NOT_RUN indicates that the instance is not run. A value of WAIT_TIME indicates that the instance is waiting to be run. A value of WAIT_RESOURCE indicates that the instance is waiting for resources. A value of RUNNING indicates that the instance is running. A value of CHECKING indicates that data quality is being checked for the node. A value of CHECKING_CONDITION indicates that branch conditions are being checked for the node. A value of FAILURE indicates that the instance fails to run. A value of SUCCESS indicates that the instance is run.
Nodes	Array of Nodes	N/A	The nodes returned for different alert sources. • The nodes that form a loop are returned if the Source parameter is set to NODE_CYCLE_ALERT. • The nodes that are isolated from each other are returned if the Source parameter is set to NODE_LONELY_ALERT.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	95279527952****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.

Parameter	Type	Example	Description
RemindId	Long	1234	The ID of the custom alert rule that was triggered. This parameter is returned if the Source parameter is set to REMIND_ALERT.
RemindName	String	Name of the custom alert rule	The name of the custom alert rule that was triggered. This parameter is returned if the Source parameter is set to REMIND_ALERT.
SlaAlert	Struct	N/A	The basic information of the baseline instance that triggered an alert. This parameter is returned if the Source parameter is set to SLA_ALERT. This parameter is left empty if the Source parameter is set to other values.
BaselineId	Long	1234	The ID of the baseline.
BaselineName	String	Baseline name	The name of the baseline.
BaselineOwner	String	952795279****	The ID of the Apsara Stack tenant account used by the baseline owner. Multiple IDs are separated by commas (,).
Bizdate	Long	1553443200000	The data timestamp of the baseline instance.
InGroupId	Integer	1	The ID of the cycle of the baseline instance. Valid values of the ID of an hour-level cycle: 1 to 24. The ID of a day-level cycle is 1.
ProjectId	Long	1234	The ID of the workspace to which the baseline belongs.

Parameter	Type	Example	Description
Status	String	SAFE	The status of the baseline. Valid values: ERROR, SAFE, DANGROUS, and OVER. A value of ERROR indicates that no nodes are associated with the baseline, or all nodes associated with the baseline are suspended. A value of SAFE indicates that nodes are run before the alert duration begins. A value of DANGROUS indicates that nodes are still running after the alert duration ends but the committed time does not arrive. A value of OVER indicates that nodes are still running after the committed time.
Source	String	REMIND_ALERT	The type of the alert. Valid values: REMIND_ALERT, TOPIC_ALERT, SLA_ALERT, NODE_CYCLE_ALERT, and NODE_LONELY_ALERT. A value of REMIND_ALERT indicates that the alert is a custom alert. A value of TOPIC_ALERT indicates that the alert is an event alert. A value of SLA_ALERT indicates that the alert is a baseline alert. A value of NODE_CYCLE_ALERT indicates that the alert is a nodes-into-loop alert. A value of NODE_LONELY_ALERT indicates that the alert is a node isolation alert.
Topics	Array of Topics	N/A	The events that triggered alerts. This parameter is returned if the Source parameter is set to TOPIC_ALERT. This parameter is left empty if the Source parameter is set to other values.
Instanceld	Long	12345	The ID of the instance that triggered the event.
Nodeld	Long	1234	The ID of the node that triggered the event.
TopicId	Long	1234	The ID of the event.
TopicName	String	9527 error	The name of the event.
TopicOwner	String	9527952795****	The ID of the Apsara Stack tenant account used by the event owner.

Parameter	Type	Example	Description
TopicStatus	String	FIXING	The status of the event. Valid values: IGNORE, NEW, FIXING, and RECOVER. A value of IGNORE indicates that the event is ignored. A value of NEW indicates that the event is a new event. A value of FIXING indicates that the event is being processed. A value of RECOVER indicates that the event is processed.
PageNumber	String	1	The page number of the returned page.
PageSize	String	10	The number of entries returned per page.
TotalCount	String	100	The total number of alerts returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```

http(s)://[Endpoint]/?Action=ListAlertMessages
&BeginTime=2020-04-02T00:00:00+0800
&EndTime=2020-04-04T00:00:00+0800
&PageNumber=1
&PageSize=10
&RegionId=cn-shanghai
&RemindId=9527
&<Common request parameters>

```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <AlertMessages>
    <AlertUser>12345</AlertUser>
    <AlertMessageStatus>READY_TO_SEND</AlertMessageStatus>
    <RemindId>9527</RemindId>
    <AlertId>9527</AlertId>
    <Content>Errors on a node</Content>
    <AlertMethod>SMS</AlertMethod>
    <AlertTime>1553531401000</AlertTime>
    <Source>REMIND_ALERT</Source>
    <RemindName>Name of the custom alert rule</RemindName>
  </AlertMessages>
  <AlertMessages>
    <Instances>
      <Status>NOT_RUN</Status>
      <NodeName>Node name</NodeName>
      <InstanceId>95279527</InstanceId>
      <NodeId>1234</NodeId>
      <ProjectId>1234</ProjectId>
    </Instances>
    <Instances>
      <TopicOwner>9527952795****</TopicOwner>
      <InstanceId>12345</InstanceId>
      <NodeId>1234</NodeId>
      <TopicId>1234</TopicId>
      <TopicStatus>FIXING</TopicStatus>
      <TopicName>9527 error</TopicName>
    </Instances>
    <Instances>
      <Owner>12345</Owner>
      <NodeName>Node name</NodeName>
      <NodeId>1234</NodeId>
      <ProjectId>1234</ProjectId>
    </Instances>
    <Topics>
      <Status>NOT_RUN</Status>
      <NodeName>Node name</NodeName>
      <InstanceId>12345</InstanceId>
      <NodeId>1234</NodeId>
      <ProjectId>1234</ProjectId>
    </Topics>
    <Topics>
      <TopicOwner>95279527****</TopicOwner>
      <InstanceId>95279527</InstanceId>
      <NodeId>9527</NodeId>
      <TopicId>9527</TopicId>
      <TopicStatus>FIXING</TopicStatus>
      <TopicName>9527 error</TopicName>
    </Topics>
```

```

<Topics>
  <Owner>95279527****</Owner>
  <NodeName>Node name</NodeName>
  <NodeId>9527</NodeId>
  <ProjectId>9527</ProjectId>
</Topics>
<Nodes>
  <Status>NOT_RUN</Status>
  <NodeName>Node name</NodeName>
  <InstanceId>95279527</InstanceId>
  <NodeId>9527</NodeId>
  <ProjectId>9527</ProjectId>
</Nodes>
<Nodes>
  <TopicOwner>95279527****</TopicOwner>
  <InstanceId>95279527</InstanceId>
  <NodeId>9527</NodeId>
  <TopicId>9527</TopicId>
  <TopicStatus>FIXING</TopicStatus>
  <TopicName>9527 error</TopicName>
</Nodes>
<Nodes>
  <Owner>95279527****</Owner>
  <NodeName>Node name</NodeName>
  <NodeId>9527</NodeId>
  <ProjectId>9527</ProjectId>
</Nodes>
</AlertMessages>
<AlertMessages>
  <SlaAlert>
    <BaselineId>9527</BaselineId>
    <Status>SAFE</Status>
    <BaselineOwner>95279527****</BaselineOwner>
    <BaselineName>Baseline name</BaselineName>
    <ProjectId>9527</ProjectId>
    <Bizdate>1553443200000</Bizdate>
    <InGroupId>1</InGroupId>
  </SlaAlert>
</AlertMessages>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "AlertMessages": [

```

```
{
  "AlertUser": 12345,
  "AlertMessageStatus": "READY_TO_SEND",
  "RemindId": 9527,
  "AlertId": 9527,
  "Content": "Errors on a node",
  "AlertMethod": "SMS",
  "AlertTime": 1553531401000,
  "Source": "REMIND_ALERT",
  "RemindName": "Name of the custom alert rule"
},
{
  "Instances": [
    {
      "Status": "NOT_RUN",
      "NodeName": "Node name",
      "InstancelId": 95279527,
      "NodeId": 1234,
      "ProjectId": 1234
    },
    {
      "TopicOwner": "9527952795****",
      "InstancelId": 12345,
      "NodeId": 1234,
      "TopicId": 1234,
      "TopicStatus": "FIXING",
      "TopicName": "9527 error"
    },
    {
      "Owner": 12345,
      "NodeName": "Node name",
      "NodeId": 1234,
      "ProjectId": 1234
    }
  ],
  "Topics": [
    {
      "Status": "NOT_RUN",
      "NodeName": "Node name",
      "InstancelId": 12345,
      "NodeId": 1234,
      "ProjectId": 1234
    },
    {
      "TopicOwner": "95279527****",
      "InstancelId": 95279527,
      "NodeId": 9527,
      "TopicId": 9527,
      "TopicStatus": "FIXING",
      "TopicName": "9527 error"
    },
    {
      "Owner": "95279527****",
      "NodeName": "Node name",
```

```

        "NodeId": 9527,
        "ProjectId": 9527
    },
    ],
    "Nodes": [
        {
            "Status": "NOT_RUN",
            "NodeName": "Node name",
            "InstanceId": 95279527,
            "NodeId": 9527,
            "ProjectId": 9527
        },
        {
            "TopicOwner": "95279527****",
            "InstanceId": 95279527,
            "NodeId": 9527,
            "TopicId": 9527,
            "TopicStatus": "FIXING",
            "TopicName": "9527 error"
        },
        {
            "Owner": "95279527****",
            "NodeName": "Node name",
            "NodeId": 9527,
            "ProjectId": 9527
        }
    ]
},
{
    "SlaAlert": {
        "BaselineId": 9527,
        "Status": "SAFE",
        "BaselineOwner": "95279527****",
        "BaselineName": "Baseline name",
        "ProjectId": 9527,
        "Bizdate": 1553443200000,
        "InGroupId": 1
    }
}
],
},
"ErrorCode": "Invalid.Tenant.ConnectionNotExists",
"ErrorMessage": "The specified parameters are invalid.",
"Success": true
}

```

Error codes

HTTP status code	Error code	Error message	Description

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Activate DataWorks first.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

2.1.6.4.2. GetBaselineConfig

Queries the configurations of a baseline.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetBaselineConfig	The operation that you want to perform.
BaselineId	Long	Yes	1234	The ID of the baseline.
RegionId	String	Yes	cn-shanghai	The region ID.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The details of the baseline.
BaselineId	Long	1234	The ID of the baseline.
BaselineName	String	Baseline name	The name of the baseline.
BaselineType	String	DAILY	The type of the baseline. Valid values: DAILY and HOURLY. The value DAILY indicates that the baseline is a day-level baseline. The value HOURLY indicates that the baseline is an hour-level baseline.
ExpHour	Integer	7	The hour in the alert time of the daily baseline. Valid values: [0,47].

Parameter	Type	Example	Description
ExpMinu	Integer	30	The minute in the alert time of the daily baseline. Valid values: [0,59].
HourExpDetail	String	{ "1": "03:28", "2": "04:28", "3": "05:28", "4": "06:28", "5": "07:28", "6": "08:28", "7": "09:28", "8": "10:28", "9": "11:28", "10": "12:28", "11": "13:28", "12": "14:28", "13": "15:28", "14": "16:28", "15": "17:28", "16": "18:28", "17": "19:28", "18": "20:28", "19": "21:28", "20": "22:28", "21": "23:28", "22": "24:28", "23": "25:28", "24": "26:28"} }	The alert time of the hourly baseline. This parameter is presented as key-value pairs in JSON format. The key is the ID of the cycle, and the value is presented in the hh:mm format. Valid values of hh: [0,47]. Valid values of mm: [0,59].
HourSlaDetail	String	{ "1": "03:58", "2": "04:58", "3": "05:58", "4": "06:58", "5": "07:58", "6": "08:58", "7": "09:58", "8": "10:58", "9": "11:58", "10": "12:58", "11": "13:58", "12": "14:58", "13": "15:58", "14": "16:58", "15": "17:58", "16": "18:58", "17": "19:58", "18": "20:58", "19": "21:58", "20": "22:58", "21": "23:58", "22": "24:58", "23": "25:58", "24": "26:58"} }	The committed time of the hourly baseline. This parameter is presented as key-value pairs in JSON format. The key is the ID of the cycle, and the value is presented in the hh:mm format. Valid values of hh: [0,47]. Valid values of mm: [0,59].
IsDefault	Boolean	true	Indicates whether the baseline is a default baseline of the workspace. Valid values: true and false. The value true indicates that the baseline is a default baseline of the workspace, and the value false indicates that the baseline is not a default baseline of the workspace.

Parameter	Type	Example	Description
Owner	String	9527952****	The ID of the Alibaba Cloud account used by the owner of the baseline. Multiple IDs are separated by commas (,).
Priority	Integer	1	The priority of the baseline. Valid values: 1, 3, 5, 7, and 8.
ProjectId	Long	1234	The ID of the workspace.
SlaHour	Integer	9	The hour in the committed time of the daily baseline. Valid values: [0,47].
SlaMinu	Integer	30	The minute in the committed time of the daily baseline. Valid values: [0,59].
UseFlag	Boolean	true	Indicates whether the baseline is enabled. Valid values: true and false. The value true indicates that the baseline is enabled, and the value false indicates that the baseline is disabled.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	401	The HTTP status code returned.
RequestId	String	ecb967ec-c137-48a5-860****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetBaselineConfig
&BaselineId=1234
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>ecb967ec-c137-48a5-860****</RequestId>
<HttpStatusCode>401</HttpStatusCode>
<Data>
  <BaselineId>1234</BaselineId>
  <IsDefault>true</IsDefault>
  <ExpHour>7</ExpHour>
  <HourSlaDetail>{"1":"03:58","2":"04:58","3":"05:58","4":"06:58","5":"07:58","6":"08:58","7":"09:58","8":"10:58","9":"11:58","10":"12:58","11":"13:58","12":"14:58","13":"15:58","14":"16:58","15":"17:58","16":"18:58","17":"19:58","18":"20:58","19":"21:58","20":"22:58","21":"23:58","22":"24:58","23":"25:58","24":"26:58"}</HourSlaDetail>
  <Owner>9527952****</Owner>
  <ExpMinu>30</ExpMinu>
  <SlaHour>9</SlaHour>
  <Priority>1</Priority>
  <SlaMinu>30</SlaMinu>
  <ProjectId>1234</ProjectId>
  <HourExpDetail>{"1":"03:28","2":"04:28","3":"05:28","4":"06:28","5":"07:28","6":"08:28","7":"09:28","8":"10:28","9":"11:28","10":"12:28","11":"13:28","12":"14:28","13":"15:28","14":"16:28","15":"17:28","16":"18:28","17":"19:28","18":"20:28","19":"21:28","20":"22:28","21":"23:28","22":"24:28","23":"25:28","24":"26:28"}</HourExpDetail>
  <UseFlag>true</UseFlag>
  <BaselineType>DAILY</BaselineType>
  <BaselineName>Baseline name</BaselineName>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "ecb967ec-c137-48a5-860****",
  "HttpStatusCode": 401,
  "Data": {
    "BaselineId": 1234,
    "IsDefault": true,
    "ExpHour": 7,
    "HourSlaDetail": "{\"1\":\"03:58\",\"2\":\"04:58\",\"3\":\"05:58\",\"4\":\"06:58\",\"5\":\"07:58\",\"6\":\"08:58\",\"7\":\"09:58\",\"8\":\"10:58\",\"9\":\"11:58\",\"10\":\"12:58\",\"11\":\"13:58\",\"12\":\"14:58\",\"13\":\"15:58\",\"14\":\"16:58\",\"15\":\"17:58\",\"16\":\"18:58\",\"17\":\"19:58\",\"18\":\"20:58\",\"19\":\"21:58\",\"20\":\"22:58\",\"21\":\"23:58\",\"22\":\"24:58\",\"23\":\"25:58\",\"24\":\"26:58\"}",
    "Owner": "9527952****",
    "ExpMinu": 30,
    "SlaHour": 9,
    "Priority": 1,
    "SlaMinu": 30,
    "ProjectId": 1234,
    "HourExpDetail": "{\"1\":\"03:28\",\"2\":\"04:28\",\"3\":\"05:28\",\"4\":\"06:28\",\"5\":\"07:28\",\"6\":\"08:28\",\"7\":\"09:28\",\"8\":\"10:28\",\"9\":\"11:28\",\"10\":\"12:28\",\"11\":\"13:28\",\"12\":\"14:28\",\"13\":\"15:28\",\"14\":\"16:28\",\"15\":\"17:28\",\"16\":\"18:28\",\"17\":\"19:28\",\"18\":\"20:28\",\"19\":\"21:28\",\"20\":\"22:28\",\"21\":\"23:28\",\"22\":\"24:28\",\"23\":\"25:28\",\"24\":\"26:28\"}",
    "UseFlag": true,
    "BaselineType": "DAILY",
    "BaselineName": "Baseline name"
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Server	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	Forbidden.Wkbench.BaselineNoPermission	You are not authorized to access the baseline.	The error message returned because you are not authorized to access the baseline.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because an error has occurred on the service.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for the resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has been reached. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.4.3. ListNodesByBaseline

Queries nodes in a baseline.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListNodesByBaseline	The operation that you want to perform.

Parameter	Type	Required	Example	Description
BaselineId	Long	Yes	1234	The ID of the baseline. You can call the GetNode operation to query the baseline ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Array of Data	N/A	The nodes in the baseline.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	9527952****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-E****	The ID of the request. You can use the ID to troubleshoot issues.

Parameter	Type	Example	Description
Success	String	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListNodesByBaseline
&BaselineId=1234
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-E****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Owner>9527952****</Owner>
  <NodeName>Node name</NodeName>
  <ProjectId>1234</ProjectId>
  <NodeId>1234</NodeId>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-E****",
  "HttpStatusCode": 200,
  "Data": {
    "Owner": "9527952****",
    "NodeName": "Node name",
    "ProjectId": 1234,
    "NodeId": 1234
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	Forbidden.Wkbench.BaselineNoPermission	You are not authorized to access the baseline.	The error message returned because you are not authorized to access the baseline.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Invalid.Wkbench.BaselineNotExist	The baseline does not exist.	The error message returned because the baseline does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

For a list of error codes, visit the [API Error Center](#).

2.1.6.4.4. ListBaselineConfigs

Queries baselines.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListBaselineConfigs	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Valid values: 1 to 100. Default value: 10.
ProjectId	Long	Yes	1234	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
Owner	String	No	95279527****	The ID of the Apsara Stack tenant account used by the baseline owner.

Parameter	Type	Required	Example	Description
Priority	String	No	1,3,5,7,8	The priority of the baseline. Valid values: 1, 3, 5, 7, and 8. You can specify multiple priorities. Separate them with commas (,).
Useflag	Boolean	No	true	Specifies whether to enable the baseline. Valid values: true and false.
BaselineTypes	String	No	DAILY,HOURLY	The type of the baseline. Valid values: DAILY and HOURLY. A value of DAILY indicates that the baseline is a day-level baseline. A value of HOURLY indicates that the baseline is an hour-level baseline. You can specify multiple types. Separate them with commas (,).
SearchText	String	No	Keyword	The keyword in the baseline name used to search for the baseline.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned baselines.
Baselines	Array of Baselines	N/A	The list of baselines.
BaselineId	Long	1234	The ID of the baseline.
BaselineName	String	Baseline name	The name of the baseline.
BaselineType	String	DAILY	The type of the baseline. Valid values: DAILY and HOURLY. A value of DAILY indicates that the baseline is a day-level baseline. A value of HOURLY indicates that the baseline is an hour-level baseline.
ExpHour	Integer	7	The hour in the alert time of the daily baseline. Valid values: 0 to 47.

Parameter	Type	Example	Description
ExpMinu	Integer	30	The minute in the alert time of the daily baseline. Valid values: 0 to 59.
HourExpDetail	String	null	The alert time of the hourly baseline. This parameter is presented as key-value pairs in the JSON format. The key indicates the ID of the cycle, and the value is presented in the hh:mm format. Valid values of hh: 0 to 47. Valid values of mm: 0 to 59.
HourSlaDetail	String	null	The committed time of the hourly baseline. This parameter is presented as key-value pairs in the JSON format. The key indicates the ID of the cycle, and the value is presented in the hh:mm format. Valid values of hh: 0 to 47. Valid values of mm: 0 to 59.
IsDefault	Boolean	true	Indicates whether the baseline is a default baseline of the workspace. Valid values: true and false.
Owner	String	952795****	The ID of the Apsara Stack tenant account used by the baseline owner. Multiple IDs are separated by commas (,).
Priority	Integer	1	The priority of the baseline. Valid values: 1, 3, 5, 7, and 8.
ProjectId	Long	9527	The ID of the workspace to which the baseline belongs.
SlaHour	Integer	9	The hour in the committed time of the daily baseline. Valid values: 0 to 47.
SlaMinu	Integer	30	The minute in the committed time of the daily baseline. Valid values: 0 to 59.
UseFlag	Boolean	true	Indicates whether the baseline is enabled. Valid values: true and false.
PageNumber	Integer	1	The page number of the returned page.

Parameter	Type	Example	Description
PageSize	Integer	10	The number of entries returned per page.
TotalCount	Integer	100	The total number of baselines returned.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListBaselineConfigs
&PageNumber=1
&PageSize=10
&ProjectId=1234
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Baselines>
    <BaselineId>1234</BaselineId>
    <IsDefault>true</IsDefault>
    <ExpHour>7</ExpHour>
    <HourSlaDetail>{"1":"03:58","2":"04:58","3":"05:58","4":"06:58","5":"07:58","6":"08:58","7":"09:58","8":"
10:58","9":"11:58","10":"12:58","11":"13:58","12":"14:58","13":"15:58","14":"16:58","15":"17:58","16":"18:58
","17":"19:58","18":"20:58","19":"21:58","20":"22:58","21":"23:58","22":"24:58","23":"25:58","24":"26:58"}</
HourSlaDetail>
    <Owner>952795****</Owner>
    <ExpMinu>30</ExpMinu>
    <SlaHour>9</SlaHour>
    <Priority>1</Priority>
    <SlaMinu>30</SlaMinu>
    <ProjectId>1234</ProjectId>
    <HourExpDetail>{"1":"03:28","2":"04:28","3":"05:28","4":"06:28","5":"07:28","6":"08:28","7":"09:28","8":
"10:28","9":"11:28","10":"12:28","11":"13:28","12":"14:28","13":"15:28","14":"16:28","15":"17:28","16":"18:2
8","17":"19:28","18":"20:28","19":"21:28","20":"22:28","21":"23:28","22":"24:28","23":"25:28","24":"26:28"}<
/HourExpDetail>
    <UseFlag>true</UseFlag>
    <BaselineType>DAILY</BaselineType>
    <BaselineName>Baseline name</BaselineName>
  </Baselines>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "Baselines": {
      "BaselineId": 1234,
      "IsDefault": true,
      "ExpHour": 7,
      "HourSlaDetail": "{ \"1\": \"03:58\", \"2\": \"04:58\", \"3\": \"05:58\", \"4\": \"06:58\", \"5\": \"07:58\", \"6\": \"08:58\", \"7\": \"09:58\", \"8\": \"10:58\", \"9\": \"11:58\", \"10\": \"12:58\", \"11\": \"13:58\", \"12\": \"14:58\", \"13\": \"15:58\", \"14\": \"16:58\", \"15\": \"17:58\", \"16\": \"18:58\", \"17\": \"19:58\", \"18\": \"20:58\", \"19\": \"21:58\", \"20\": \"22:58\", \"21\": \"23:58\", \"22\": \"24:58\", \"23\": \"25:58\", \"24\": \"26:58\" }",
      "Owner": "952795****",
      "ExpMinu": 30,
      "SlaHour": 9,
      "Priority": 1,
      "SlaMinu": 30,
      "ProjectId": 1234,
      "HourExpDetail": "{ \"1\": \"03:28\", \"2\": \"04:28\", \"3\": \"05:28\", \"4\": \"06:28\", \"5\": \"07:28\", \"6\": \"08:28\", \"7\": \"09:28\", \"8\": \"10:28\", \"9\": \"11:28\", \"10\": \"12:28\", \"11\": \"13:28\", \"12\": \"14:28\", \"13\": \"15:28\", \"14\": \"16:28\", \"15\": \"17:28\", \"16\": \"18:28\", \"17\": \"19:28\", \"18\": \"20:28\", \"19\": \"21:28\", \"20\": \"22:28\", \"21\": \"23:28\", \"22\": \"24:28\", \"23\": \"25:28\", \"24\": \"26:28\" }",
      "UseFlag": true,
      "BaselineType": "DAILY",
      "BaselineName": "Baseline name"
    }
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Serve	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Wkbench.NoProject	You have not joined any project.	The error message returned because you have not joined a workspace.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.5. GetBaselineKeyPath

Queries the key path of a baseline instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetBaselineKeyPath	The operation that you want to perform.
BaselineId	Long	Yes	1234	The ID of the baseline.
Bizdate	String	Yes	2020-07-07T00:00:00+0800	The data timestamp of the baseline instance. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
InGroupId	Integer	Yes	1	The ID of the cycle of the baseline instance. Valid values of the ID of an hour-level cycle: 1 to 24. The ID of a day-level cycle is 1.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Response parameters

Parameter	Type	Example	Description
Data	Array of Data	N/A	The information about the key path.
Bizdate	Long	1553443200000	The data timestamp of the baseline instance.
InGroupId	Integer	1	The ID of the cycle of the baseline instance. Valid values: 1 to 288.
InstanceId	Long	12345	The ID of the instance.
NodeId	Long	1234	The ID of the node.

Parameter	Type	Example	Description
NodeName	String	Node name	The name of the node.
Owner	String	9527952****	The ID of the Apsara Stack tenant account used by the node owner.
PrgType	Integer	10	The type of the node. Valid values: 23, 10, 6, and 99. A value of 23 indicates that the node is a Data Integration node. A value of 10 indicates that the node is a MaxCompute SQL node. A value of 6 indicates that the node is a Shell node. A value of 99 indicates that the node is a zero load node.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.
Runs	Array of Runs	N/A	The information about the baseline instance that was run.
AbsTime	Long	1553531402000	The timestamp obtained by combining the predicted time when the baseline instance started to run with the historical average duration that the baseline instance was running.
BeginCast	Long	1553531686000	The timestamp of the predicted time when the baseline instance started to run.
BeginRunningTime	Long	1553531401000	The timestamp of the actual time when the baseline instance started to run.
BeginWaitResTime	Long	1553531401000	The timestamp when the baseline instance started to wait for resources.
BeginWaitTime	Long	1553531400000	The timestamp when the baseline instance started to wait to be run.
Bizdate	Long	1553443200000	The data timestamp of the baseline instance.

Parameter	Type	Example	Description
CycTime	Long	1553443200000	The timestamp of the scheduled time at which the baseline instance is run.
EndCast	Long	1553531687000	The timestamp of the predicted time when the baseline instance stopped running.
FinishTime	Long	1553531401000	The timestamp of the actual time when the baseline instance stopped running.
InGroupId	Integer	1	The ID of the cycle of the baseline instance. Valid values: 1 to 288.
InstanceId	Long	1234	The ID of the instance.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	9527952****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.
Status	String	SUCCESS	The status of the instance. Valid values: NOT_RUN, WAIT_TIME, WAIT_RESOURCE, RUNNING, CHECKING, CHECKING_CONDITION, FAILURE, and SUCCESS. A value of NOT_RUN indicates that the instance is not run. A value of WAIT_TIME indicates that the instance is waiting to be run. A value of WAIT_RESOURCE indicates that the instance is waiting for resources. A value of RUNNING indicates that the instance is running. A value of CHECKING indicates that data quality is being checked for the node. A value of CHECKING_CONDITION indicates that branch conditions are being checked for the node. A value of FAILURE indicates that the instance fails to run. A value of SUCCESS indicates that the instance is run.

Parameter	Type	Example	Description
Topics	Array of Topics	N/A	The information about the events that are associated with the baseline instance.
AddTime	Long	1553531401000	The timestamp when the event was found.
InstanceId	Long	1234	The ID of the instance.
TopicId	Long	1234	The ID of the event.
TopicName	Long	1234	The name of the event.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetBaselineKeyPath
&BaselineId=1234
&Bizdate=2020-07-07T00:00:00+0800
&InGroupId=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <PrgType>10</PrgType>
  <Owner>9527952****</Owner>
```

```

<Instanceld>1234</Instanceld>
<NodeName>Node name</NodeName>
<ProjectId>1234</ProjectId>
<NodeId>1234</NodeId>
<Bizdate>1553443200000</Bizdate>
<InGroupId>1</InGroupId>
</Data>
<Data>
  <Topics>
    <Status>SUCCESS</Status>
    <Owner>9527952****</Owner>
    <Instanceld>1234</Instanceld>
    <NodeName>Node name</NodeName>
    <ProjectId>1234</ProjectId>
    <BeginWaitTimeTime>1553531400000</BeginWaitTimeTime>
    <CycTime>1553443200000</CycTime>
    <BeginRunningTime>1553531401000</BeginRunningTime>
    <Bizdate>1553443200000</Bizdate>
    <InGroupId>1</InGroupId>
    <BeginCast>1553531686000</BeginCast>
    <AbsTime>1553531402000</AbsTime>
    <FinishTime>1553531401000</FinishTime>
    <NodeId>9527</NodeId>
    <BeginWaitResTime>1553531401000</BeginWaitResTime>
    <EndCast>1553531687000</EndCast>
  </Topics>
  <Topics>
    <Instanceld>1234</Instanceld>
    <TopicId>1234</TopicId>
    <AddTime>1553531401000</AddTime>
    <TopicName>1234</TopicName>
  </Topics>
  <Runs>
    <Status>SUCCESS</Status>
    <Owner>9527952****</Owner>
    <Instanceld>1234</Instanceld>
    <NodeName>Node name</NodeName>
    <ProjectId>1234</ProjectId>
    <BeginWaitTimeTime>1553531400000</BeginWaitTimeTime>
    <CycTime>1553443200000</CycTime>
    <BeginRunningTime>1553531401000</BeginRunningTime>
    <Bizdate>1553443200000</Bizdate>
    <InGroupId>1</InGroupId>
    <BeginCast>1553531686000</BeginCast>
    <AbsTime>1553531402000</AbsTime>
    <FinishTime>1553531401000</FinishTime>
    <NodeId>1234</NodeId>
    <BeginWaitResTime>1553531401000</BeginWaitResTime>
    <EndCast>1553531687000</EndCast>
  </Runs>
  <Runs>
    <Instanceld>1234</Instanceld>
    <TopicId>1234</TopicId>
    <AddTime>1553531401000</AddTime>

```

```
<TopicName>1234</TopicName>
</Runs>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": [
    {
      "PrgType": 10,
      "Owner": "9527952****",
      "InstanceId": 1234,
      "NodeName": "Node name",
      "ProjectId": 1234,
      "NodeId": 1234,
      "Bizdate": 1553443200000,
      "InGroupId": 1
    },
    {
      "Topics": [
        {
          "Status": "SUCCESS",
          "Owner": "9527952****",
          "InstanceId": 1234,
          "NodeName": "Node name",
          "ProjectId": 1234,
          "BeginWaitTime": 1553531400000,
          "CycTime": 1553443200000,
          "BeginRunningTime": 1553531401000,
          "Bizdate": 1553443200000,
          "InGroupId": 1,
          "BeginCast": 1553531686000,
          "AbsTime": 1553531402000,
          "FinishTime": 1553531401000,
          "NodeId": 9527,
          "BeginWaitResTime": 1553531401000,
          "EndCast": 1553531687000
        },
        {
          "InstanceId": 1234,
          "TopicId": 1234,
          "AddTime": 1553531401000,
          "TopicName": 1234
        }
      ],
      "Runs": [
        {
          "Status": "SUCCESS",
          "Owner": "9527952****"
```

```

    "OwnerId": 10021002,
    "InstanceId": 1234,
    "NodeName": "Node name",
    "ProjectId": 1234,
    "BeginWaitTime": 1553531400000,
    "CycTime": 1553443200000,
    "BeginRunningTime": 1553531401000,
    "Bizdate": 1553443200000,
    "InGroupId": 1,
    "BeginCast": 1553531686000,
    "AbsTime": 1553531402000,
    "FinishTime": 1553531401000,
    "NodeId": 1234,
    "BeginWaitResTime": 1553531401000,
    "EndCast": 1553531687000
  },
  {
    "InstanceId": 1234,
    "TopicId": 1234,
    "AddTime": 1553531401000,
    "TopicName": 1234
  }
]
},
"ErrorCode": 1031203110005,
"ErrorMessage": "The specified parameters are invalid.",
"Success": true
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Wkbench.BaselineNoPermission	You are not authorized to access the baseline.	The error message returned because you are not authorized to access the baseline.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	Forbidden.Not.Server	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Invalid.Wkbench.BaselineStatusNotExist	The BaselineStatus does not exist.	The error message returned because the baseline instance does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.6. GetBaselineStatus

Queries the details of a baseline instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetBaselineStatus	The operation that you want to perform.
BaselineId	Long	Yes	1234	The ID of the baseline.
Bizdate	String	Yes	2020-07-07T00:00:00+0800	The data timestamp of the baseline instance. Specify the time in the ISO 8601 standard in the yyyy-MM-ddThh:mm:ssZ format. The time must be in UTC.
InGroupId	Integer	Yes	1	The ID of the cycle of the baseline instance. Valid values of the ID of an hour-level cycle: 1 to 24. The ID of a day-level cycle is 1.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the baseline instance.
BaselineId	Long	1234	The ID of the baseline.
BaselineName	String	Baseline name	The name of the baseline.
Bizdate	Long	1553443200000	The data timestamp of the baseline instance.
BlockInstance	Struct	N/A	The information about the key path.
EndCast	Long	1553443200000	The timestamp of the predicted time when the baseline instance stopped running.
FinishTime	Long	1553443200000	The timestamp of the actual time when the baseline instance stopped running.

Parameter	Type	Example	Description
InstanceId	Long	12345	The ID of the instance.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	9527952795****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.
Status	String	SUCCESS	The status of the instance. Valid values: NOT_RUN, WAIT_TIME, WAIT_RESOURCE, RUNNING, CHECKING, CHECKING_CONDITION, FAILURE, and SUCCESS. A value of NOT_RUN indicates that the instance is not run. A value of WAIT_TIME indicates that the instance is waiting to be run. A value of WAIT_RESOURCE indicates that the instance is waiting for resources. A value of RUNNING indicates that the instance is running. A value of CHECKING indicates that data quality is being checked for the node. A value of CHECKING_CONDITION indicates that branch conditions are being checked for the node. A value of FAILURE indicates that the instance fails to run. A value of SUCCESS indicates that the instance is run.
Buffer	Float	1200	The margin of the baseline instance. Unit: seconds.
EndCast	Long	1553443200000	The timestamp of the predicted time when the baseline instance stopped running.
ExpTime	Long	1553443200000	The timestamp of the alert time of the baseline instance.

Parameter	Type	Example	Description
FinishStatus	String	UNFINISH	The status of the baseline instance. Valid values: UNFINISH and FINISH. A value of UNFINISH indicates that the baseline instance is still running. A value of FINISH indicates that the baseline instance is run.
FinishTime	Long	1553443200000	The timestamp of the actual time when the baseline instance stopped running. This parameter is returned if the FinishStatus parameter is set to FINISH.
InGroupId	Integer	1	The ID of the cycle of the baseline instance. Valid values of the ID of an hour-level cycle: 1 to 24. The ID of a day-level cycle is 1.
LastInstance	Struct	N/A	The latest instance information.
EndCast	Long	1553443200000	The timestamp of the predicted time when the baseline instance stopped running.
FinishTime	Long	1553443200000	The timestamp of the actual time when the baseline instance stopped running.
InstanceId	Long	12345	The ID of the instance.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	9527952795****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.

Parameter	Type	Example	Description
Status	String	SUCCESS	The status of the instance. Valid values: NOT_RUN, WAIT_TIME, WAIT_RESOURCE, RUNNING, CHECKING, CHECKING_CONDITION, FAILURE, and SUCCESS. A value of NOT_RUN indicates that the instance is not run. A value of WAIT_TIME indicates that the instance is waiting to be run. A value of WAIT_RESOURCE indicates that the instance is waiting for resources. A value of RUNNING indicates that the instance is running. A value of CHECKING indicates that data quality is being checked for the node. A value of CHECKING_CONDITION indicates that branch conditions are being checked for the node. A value of FAILURE indicates that the instance fails to run. A value of SUCCESS indicates that the instance is run.
Owner	String	9527952795****	The ID of the Apsara Stack tenant account used by the baseline owner. Multiple IDs are separated by commas (,).
Priority	Integer	1	The priority of the baseline. Valid values: 1, 2, 5, 7, and 8.
ProjectId	Long	1234	The ID of the workspace to which the baseline belongs.
SlaTime	Long	1553443200000	The timestamp of the committed time of the baseline instance.

Parameter	Type	Example	Description
Status	String	SAFE	The status of the baseline. Valid values: ERROR, SAFE, DANGROUS, and OVER. A value of ERROR indicates that no nodes are associated with the baseline, or all nodes associated with the baseline are suspended. A value of SAFE indicates that nodes are run before the alert duration begins. A value of DANGROUS indicates that nodes are still running after the alert duration ends but the committed time does not arrive. A value of OVER indicates that nodes are still running after the committed time.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetBaselineStatus
&BaselineId=1234
&Bizdate=2020-07-07T00:00:00+0800
&InGroupId=1
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <BaselineId>1234</BaselineId>
  <Buffer>1200</Buffer>
  <Status>SAFE</Status>
  <Owner>9527952795****</Owner>
  <Priority>1</Priority>
  <FinishStatus>UNFINISH</FinishStatus>
  <ProjectId>9527</ProjectId>
  <Bizdate>1553443200000</Bizdate>
  <ExpTime>1553443200000</ExpTime>
  <InGroupId>1</InGroupId>
  <FinishTime>1553443200000</FinishTime>
  <BlockInstance>
    <Status>SUCCESS</Status>
    <Owner>9527952795****</Owner>
    <FinishTime>1553443200000</FinishTime>
    <InstanceId>12345</InstanceId>
    <NodeName>Node name</NodeName>
    <ProjectId>1234</ProjectId>
    <NodeId>1234</NodeId>
    <EndCast>1553443200000</EndCast>
  </BlockInstance>
  <BaselineName>Baseline name</BaselineName>
  <SlaTime>1553443200000</SlaTime>
  <EndCast>1553443200000</EndCast>
  <LastInstance>
    <Status>SUCCESS</Status>
    <Owner>9527952795****</Owner>
    <FinishTime>1553443200000</FinishTime>
    <InstanceId>12345</InstanceId>
    <NodeName>Node name</NodeName>
    <ProjectId>1234</ProjectId>
    <NodeId>1234</NodeId>
    <EndCast>1553443200000</EndCast>
  </LastInstance>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "BaselineId": 1234,
    "Buffer": 1200,
    "Status": "SAFE",
    "Owner": "9527952795****",
    "Priority": 1,
    "FinishStatus": "UNFINISH",
    "ProjectId": 9527,
    "Bizdate": 1553443200000,
    "ExpTime": 1553443200000,
    "InGroupId": 1,
    "FinishTime": 1553443200000,
    "BlockInstance": {
      "Status": "SUCCESS",
      "Owner": "9527952795****",
      "FinishTime": 1553443200000,
      "InstanceId": 12345,
      "NodeName": "Node name",
      "ProjectId": 1234,
      "NodeId": 1234,
      "EndCast": 1553443200000
    },
    "BaselineName": "Baseline name",
    "SlaTime": 1553443200000,
    "EndCast": 1553443200000,
    "LastInstance": {
      "Status": "SUCCESS",
      "Owner": "9527952795****",
      "FinishTime": 1553443200000,
      "InstanceId": 12345,
      "NodeName": "Node name",
      "ProjectId": 1234,
      "NodeId": 1234,
      "EndCast": 1553443200000
    }
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.7. ListBaselineStatuses

Queries baseline instances.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListBaselineStatuses	The operation that you want to perform.
Bizdate	String	Yes	2020-07-07T00:00:00+0800	The data timestamp of the baseline instance. Specify the time in the ISO 8601 standard in the yyyy-MM-ddThh:mm:ssZ format. The time must be in UTC.
PageNumber	Integer	Yes	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
RegionId	String	Yes	cn-shanghai	The ID of the region.
Priority	String	No	1,3,5,7,8	The priority of the baseline. Valid values: 1, 3, 5, 7, and 8. You can specify multiple priorities. Separate them with commas (,).
SearchText	String	No	Keyword	The keyword in the baseline name used to search for the baseline.
Owner	String	No	9527952795****	The ID of the Apsara Stack tenant account used by the baseline owner.
TopicId	Long	No	1234	The ID of the event.

Parameter	Type	Required	Example	Description
FinishStatus	String	No	FINISH,UNFINISH	The status of the baseline instance. Valid values: UNFINISH and FINISH. A value of UNFINISH indicates that the baseline instance is still running. A value of FINISH indicates that the baseline instance is run. You can specify multiple states. Separate them with commas (,).
Status	String	No	SAFE,DANGROUS,OVER	The status of the baseline. Valid values: ERROR, SAFE, DANGROUS, and OVER. A value of ERROR indicates that no nodes are associated with the baseline, or all nodes associated with the baseline are suspended. A value of SAFE indicates that nodes are run before the alert duration begins. A value of DANGROUS indicates that nodes are still running after the alert duration ends but the committed time does not arrive. A value of OVER indicates that nodes are still running after the committed time. You can specify multiple states. Separate them with commas (,).
BaselineTypes	String	No	DAILY,HOURLY	The type of the baseline. Valid values: DAILY and HOURLY. A value of DAILY indicates that the baseline is a day-level baseline. A value of HOURLY indicates that the baseline is an hour-level baseline. You can specify multiple types. Separate them with commas (,).

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned baseline instances.
BaselineStatuses	Array of BaselineStatuses	N/A	The list of baseline instances.
BaselineId	Long	1234	The ID of the baseline.

Parameter	Type	Example	Description
BaselineName	String	Baseline name	The name of the baseline.
Bizdate	Long	1553443200000	The data timestamp of the baseline instance.
Buffer	Long	1800	The margin of the baseline instance. Unit: seconds.
EndCast	Long	1553531400000	The timestamp of the predicted time when the baseline instance stopped running.
ExpTime	Long	1553531400000	The timestamp of the alert time of the baseline instance.
FinishStatus	String	UNFINISH	The status of the baseline instance. Valid values: UNFINISH and FINISH. A value of UNFINISH indicates that the baseline instance is still running. A value of FINISH indicates that the baseline instance is run.
FinishTime	Long	1553531400000	The timestamp of the actual time when the baseline instance stopped running. This parameter is returned if the FinishStatus parameter is set to FINISH.
InGroupld	Integer	1	The ID of the cycle of the baseline instance. Valid values of the ID of an hour-level cycle: 1 to 24. The ID of a day-level cycle is 1.
Owner	String	9527952795****	The ID of the Apsara Stack tenant account used by the baseline owner. Multiple IDs are separated by commas (,).
Priority	Integer	1	The priority of the baseline. Valid values: 1, 2, 5, 7, and 8.
ProjectId	Long	1234	The ID of the workspace to which the baseline belongs.
SlaTime	Long	1553531400000	The timestamp of the actual time when the baseline instance stopped running.

Parameter	Type	Example	Description
Status	String	SAFE	The status of the baseline. Valid values: ERROR, SAFE, DANGROUS, and OVER. A value of ERROR indicates that no nodes are associated with the baseline, or all nodes associated with the baseline are suspended. A value of SAFE indicates that nodes are run before the alert duration begins. A value of DANGROUS indicates that nodes are still running after the alert duration ends but the committed time does not arrive. A value of OVER indicates that nodes are still running after the committed time.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
TotalCount	Integer	100	The total number of baseline instances returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListBaselineStatuses
&Bizdate=2020-07-07T00:00:00+0800
&PageNumber=1
&PageSize=10
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <BaselineStatuses>
    <BaselineId>1234</BaselineId>
    <Buffer>1800</Buffer>
    <Status>SAFE</Status>
    <Owner>9527952795****</Owner>
    <Priority>1</Priority>
    <FinishStatus>UNFINISH</FinishStatus>
    <ProjectId>1234</ProjectId>
    <Bizdate>1553443200000</Bizdate>
    <ExpTime>1553531400000</ExpTime>
    <InGroupId>1</InGroupId>
    <FinishTime>1553531400000</FinishTime>
    <BaselineName>Baseline name</BaselineName>
    <SlaTime>1553531400000</SlaTime>
    <EndCast>1553531400000</EndCast>
  </BaselineStatuses>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "BaselineStatuses": {
      "BaselineId": 1234,
      "Buffer": 1800,
      "Status": "SAFE",
      "Owner": "9527952795****",
      "Priority": 1,
      "FinishStatus": "UNFINISH",
      "ProjectId": 1234,
      "Bizdate": 1553443200000,
      "ExpTime": 1553531400000,
      "InGroupId": 1,
      "FinishTime": 1553531400000,
      "BaselineName": "Baseline name",
      "SlaTime": 1553531400000,
      "EndCast": 1553531400000
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.

HTTP status code	Error code	Error message	Description
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.8. DeleteRemind

Deletes a custom alert rule.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteRemind	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RemindId	Long	Yes	1234	The ID of the custom alert rule.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the custom alert rule is deleted. Valid values: true and false. A value of true indicates that the custom alert rule is deleted. A value of false indicates that the custom alert rule fails to be deleted.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteRemind
&RegionId=cn-shanghai
&RemindId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Serve	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
500	Forbidden.Wkbench.RemindNoPermission	You are not authorized to modify the reminder.	The error message returned because you are not authorized to modify the custom alert rule.
500	Invalid.Wkbench.RemindNotExist	The reminder does not exist.	The error message returned because the custom alert rule does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

HTTP status code	Error code	Error message	Description

2.1.6.4.9. UpdateRemind

Updates a custom alert rule.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateRemind	The operation that you want to perform.
AlertInterval	Integer	Yes	1800	The minimum interval at which alerts are reported. Unit: seconds. Minimum value: 1200. Default value: 1800.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RemindId	Long	Yes	1234	The ID of the custom alert rule.
RemindName	String	No	update_remindname	The name of the custom alert rule. The name can be a maximum of 128 characters in length.
DndEnd	String	No	08:00	The end time of the quiet hours. Specify this parameter in the hh:mm format. Valid values of hh: 0 to 23. Valid values of mm: 0 to 59.

Parameter	Type	Required	Example	Description
RemindUnit	String	No	NODE	The type of the object to which the custom alert rule is applied. Valid values: NODE, BASELINE, PROJECT, and BIZPROCESS. A value of NODE indicates that the monitored object is a node. A value of BASELINE indicates that the monitored object is a baseline. A value of PROJECT indicates that the monitored object is a workspace. A value of BIZPROCESS indicates that the monitored object is a workflow.
Nodelds	String	No	1,2,3	The ID of the node to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to NODE. You can specify multiple IDs. Separate them with commas (.). A maximum of 50 nodes can be specified for a custom alert rule.
Baselineds	String	No	1,2,3	The ID of the baseline to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to BASELINE. You can specify multiple IDs. Separate them with commas (.). A maximum of five baselines can be specified for a custom alert rule.
ProjectId	Long	No	1234	The ID of the workspace to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to PROJECT. Only one workspace can be specified for a custom alert rule.

Parameter	Type	Required	Example	Description
BizProcessIds	String	No	1,2,3	The ID of the workflow to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to BIZPROCESS. You can specify multiple IDs. Separate them with commas (.). A maximum of five workflows can be specified for a custom alert rule.
RemindType	String	No	FINISHED	The conditions that trigger an alert. Valid values: FINISHED, UNFINISHED, ERROR, CYCLE_UNFINISHED, and TIMEOUT. A value of FINISHED indicates that the node is run. A value of UNFINISHED indicates that the node is still running at the specified point in time. A value of ERROR indicates that an error occurs when the node is running. A value of CYCLE_UNFINISHED indicates that the node is still running in the specified cycle. A value of TIMEOUT indicates that the node times out.
MaxAlertTimes	Integer	No	3	The maximum number of alerts. Valid values: 1 to 10. Default value: 3.

Parameter	Type	Required	Example	Description
Detail	String	No	null	The details of the conditions that trigger an alert. - If the RemindType parameter is set to FINISHED, leave this parameter empty. - If the RemindType parameter is set to UNFINISHED, specify this parameter as key-value pairs. Example: {"hour":23,"minu":59}. Valid values of hour: 0 to 47. Valid values of minu: 0 to 59. - If the RemindType parameter is set to ERROR, leave this parameter empty. - If the RemindType parameter is set to CYCLE_UNFINISHED, specify this parameter as key-value pairs in the JSON format. Example: {"1":"05:50","2":"06:50","3":"07:50","4":"08:50","5":"09:50","6":"10:50","7":"11:50","8":"12:50","9":"13:50","10":"14:50","11":"15:50","12":"16:50","13":"17:50","14":"18:50","15":"19:50","16":"20:50","17":"21:50","18":"22:50","19":"23:50","20":"24:50","21":"25:50"}. The key indicates the ID of the cycle. Valid values of the ID: 1 to 288. The value indicates the timeout period of the node that is running in the cycle. Specify the value in the hh:mm format. Valid values of hh: 0 to 47. Valid values of mm: 0 to 59. - If the RemindType parameter is set to TIMEOUT, set this parameter to the timeout period. Unit: seconds. Example: 1800. This indicates that an alert is reported if the node is running for longer than 30 minutes.
AlertUnit	String	No	OWNER	The alert recipient. Valid values: OWNER and OTHER. A value of OWNER indicates that the alert notification is sent to the node owner. A value of OTHER indicates that the alert notification is sent to a specific user.

Parameter	Type	Required	Example	Description
AlertMethods	String	No	SMS,MAIL	The notification method. Valid values: MAIL, SMS, and PHONE. A value of MAIL indicates that the notification is sent by email. A value of SMS indicates that the notification is sent by text message. A value of PHONE indicates that the notification is sent by call, which is supported only by DataWorks Professional Edition and more advanced editions. You can specify multiple notification methods. Separate them with commas (,).
AlertTargets	String	No	9527952795279527	- If the AlertUnit parameter is set to OWNER, leave this parameter empty. - If the AlertUnit parameter is set to OTHER, set this parameter to the ID of the Apsara Stack tenant account of a specific user. You can specify multiple IDs. Separate them with commas (,). You can specify a maximum of 10 IDs.
UseFlag	Boolean	No	true	Specifies whether to enable the custom alert rule. Valid values: true and false.
RobotUrls	String	No	https://oapi.dingtalk.com/robot/send?access_token=*****	The webhook URL of the DingTalk chatbot. You can specify multiple webhook URLs. Separate them with commas (,).

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the custom alert rule is updated. Valid values: true and false.
ErrorCode	String	1031203110005	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateRemind
&AlertInterval=1800
&RegionId=cn-shanghai
&RemindId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Servicer	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	Forbidden.Wkbench.BaselineNoPermission	You are not authorized to access the baseline.	The error message returned because you are not authorized to access the baseline.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HTTP status code	Error code	Error message	Description
500	Forbidden.Wkbench.RemindNoPermission	You are not authorized to modify the reminder.	The error message returned because you are not authorized to modify the custom alert rule.
500	Invalid.Wkbench.RemindNotExist	The reminder does not exist.	The error message returned because the custom alert rule does not exist.

2.1.6.4.10. CreateRemind

Creates a custom alert rule.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateRemind	The operation that you want to perform.
AlertUnit	String	Yes	OWNER	The alert recipient. Valid values: OWNER and OTHER. A value of OWNER indicates that the alert notification is sent to the node owner. A value of OTHER indicates that the alert notification is sent to a specific user.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RemindName	String	Yes	test_error_remind	The name of the custom alert rule. The name can be a maximum of 128 characters in length.

Parameter	Type	Required	Example	Description
RemindType	String	Yes	FINISHED	The conditions that trigger an alert. Valid values: FINISHED, UNFINISHED, ERROR, CYCLE_UNFINISHED, and TIMEOUT. A value of FINISHED indicates that the node is run. A value of UNFINISHED indicates that the node is still running at the specified point in time. A value of ERROR indicates that an error occurs when the node is running. A value of CYCLE_UNFINISHED indicates that the node is still running in the specified cycle. A value of TIMEOUT indicates that the node times out.
RemindUnit	String	Yes	NODE	The type of the object to which the custom alert rule is applied. Valid values: NODE, BASELINE, PROJECT, and BIZPROCESS. A value of NODE indicates that the monitored object is a node. A value of BASELINE indicates that the monitored object is a baseline. A value of PROJECT indicates that the monitored object is a workspace. A value of BIZPROCESS indicates that the monitored object is a workflow.
DndEnd	String	No	08:00	The end time of the quiet hours. Specify this parameter in the hh:mm format. Valid values of hh: 0 to 23. Valid values of mm: 0 to 59.
Nodelds	String	No	1,2,3	The ID of the node to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to NODE. You can specify multiple IDs. Separate them with commas (.). A maximum of 50 nodes can be specified for a custom alert rule.
Baselineds	String	No	1,2,3	The ID of the baseline to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to BASELINE. You can specify multiple IDs. Separate them with commas (.). A maximum of five baselines can be specified for a custom alert rule.

Parameter	Type	Required	Example	Description
ProjectId	Long	No	9527	The ID of the workspace to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to PROJECT. Only one workspace can be specified for a custom alert rule.
BizProcessIds	String	No	1,2,3	The ID of the workflow to which the custom alert rule is applied. This parameter takes effect if the RemindUnit parameter is set to BIZPROCESS. You can specify multiple IDs. Separate them with commas (.). A maximum of five workflows can be specified for a custom alert rule.
MaxAlertTimes	Integer	No	2	The maximum number of alerts. Valid values: 1 to 10. Default value: 3.
AlertInterval	Integer	No	1800	The minimum interval at which alerts are reported. Unit: seconds. Minimum value: 1200. Default value: 1800.

Parameter	Type	Required	Example	Description
Detail	String	No	null	The details of the conditions that trigger an alert. - If the RemindType parameter is set to FINISHED, leave this parameter empty. - If the RemindType parameter is set to UNFINISHED, specify this parameter as key-value pairs. Example: {"hour":23,"minu":59}. Valid values of hour: 0 to 47. Valid values of minu: 0 to 59. - If the RemindType parameter is set to ERROR, leave this parameter empty. - If the RemindType parameter is set to CYCLE_UNFINISHED, specify this parameter as key-value pairs in the JSON format. Example: {"1":"05:50","2":"06:50","3":"07:50","4":"08:50","5":"09:50","6":"10:50","7":"11:50","8":"12:50","9":"13:50","10":"14:50","11":"15:50","12":"16:50","13":"17:50","14":"18:50","15":"19:50","16":"20:50","17":"21:50","18":"22:50","19":"23:50","20":"24:50","21":"25:50"}. The key indicates the ID of the cycle. Valid values of the ID: 1 to 288. The value indicates the timeout period of the node that is running in the cycle. Specify the value in the hh:mm format. Valid values of hh: 0 to 47. Valid values of mm: 0 to 59. - If the RemindType parameter is set to TIMEOUT, set this parameter to the timeout period. Unit: seconds. Example: 1800. This indicates that an alert is reported if the node is running for longer than 30 minutes.

Parameter	Type	Required	Example	Description
AlertMethods	String	No	SMS,MAIL	The notification method. Valid values: MAIL, SMS, and PHONE. A value of MAIL indicates that the notification is sent by email. A value of SMS indicates that the notification is sent by text message. A value of PHONE indicates that the notification is sent by call, which is supported only by DataWorks Professional Edition and more advanced editions. You can specify multiple notification methods. Separate them with commas (,).
AlertTargets	String	No	9527952795279527	- If the AlertUnit parameter is set to OWNER, leave this parameter empty. - If the AlertUnit parameter is set to OTHER, set this parameter to the ID of the Apsara Stack tenant account of a specific user. You can specify multiple IDs. Separate them with commas (,). You can specify a maximum of 10 IDs.
RobotUrls	String	No	https://oapi.dingtalk.com/robot/send?access_token=*****	The webhook URL of the DingTalk chatbot. You can specify multiple webhook URLs. Separate them with commas (,).

Response parameters

Parameter	Type	Example	Description
Data	Long	1234	The ID of the custom alert rule.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateRemind
&AlertUnit=OWNER
&RegionId=cn-shanghai
&RemindName=test_error_remind
&RemindType=FINISHED
&RemindUnit=NODE
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>9527</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": 9527,
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Servicer	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.

HTTP status code	Error code	Error message	Description
500	Forbidden.Wkbench.BaselineNoPermission	You are not authorized to access the baseline.	The error message returned because you are not authorized to access the baseline.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Wkbench.NoProject	You have not joined any project.	The error message returned because you have not joined a workspace.

HTTP status code	Error code	Error message	Description
500	Invalid.Wkbench.NodeNotExist	The node does not exist.	The error message returned because the node does not exist.
500	Invalid.Wkbench.BaselineNotExist	The baseline does not exist.	The error message returned because the baseline does not exist.
500	Forbidden.Wkbench.RemindOverlimit	The maximum number of reminders is exceeded.	The error message returned because the number of custom alert rules has reached the upper limit.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.11. GetRemind

Queries the details of a custom alert rule.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetRemind	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RemindId	Long	Yes	1234	The ID of the custom alert rule.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the custom alert rule.
AlertInterval	Integer	1800	The minimum interval at which alerts are reported. Unit: seconds.

Parameter	Type	Example	Description
AlertMethods	List	SMS,MAIL	The notification method. Valid values: MAIL, SMS, and PHONE. A value of MAIL indicates that the notification is sent by email. A value of SMS indicates that the notification is sent by text message. A value of PHONE indicates that the notification is sent by call, which is supported only by DataWorks Professional Edition and more advanced editions. Multiple notification methods are separated by commas (,).
AlertTargets	List	9527952795279527	- If the AlertUnit parameter is set to OWNER, this parameter is left empty. - If the AlertUnit parameter is set to OTHER, the ID of the Apsara Stack tenant account of a specific user is returned. Multiple IDs are separated by commas (,).
AlertUnit	String	OWNER	The alert recipient. Valid values: OWNER and OTHER. A value of OWNER indicates that the alert notification is sent to the node owner. A value of OTHER indicates that the alert notification is sent to a specific user.
Baselines	Array of Baselines	N/A	The baselines to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to BASELINE.
BaselineId	Long	1234	The ID of the baseline.
BaselineName	String	Baseline name	The name of the baseline.
BizProcesses	Array of BizProcesses	N/A	The workflows to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to BIZPROCESS.
BizId	Long	9527	The ID of the workflow.
BizProcessName	String	Workflow name	Workflow name

Parameter	Type	Example	Description
Detail	String	null	- If the RemindType parameter is set to FINISHED, this parameter is left empty. - If the RemindType parameter is set to UNFINISHED, the trigger conditions are returned as key-value pairs. Example: {"hour":23,"minu":59}. Valid values of hour: 0 to 47. Valid values of minu: 0 to 59. - If the RemindType parameter is set to ERROR, this parameter is left empty. - If the RemindType parameter is set to CYCLE_UNFINISHED, the trigger conditions are returned as key-value pairs in the JSON format. Example: {"1":"05:50","2":"06:50","3":"07:50","4":"08:50","5":"09:50","6":"10:50","7":"11:50","8":"12:50","9":"13:50","10":"14:50","11":"15:50","12":"16:50","13":"17:50","14":"18:50","15":"19:50","16":"20:50","17":"21:50","18":"22:50","19":"23:50","20":"24:50","21":"25:50"}. The key indicates the ID of the cycle. Valid values of the ID: 1 to 288. The value indicates the timeout period of the node that is running in the cycle. The value is presented in the hh:mm format. Valid values of hh: 0 to 47. Valid values of mm: 0 to 59. - If the RemindType parameter is set to TIMEOUT, the timeout period is returned. Unit: seconds. Example: 1800. This indicates that an alert is reported if the node is running for longer than 30 minutes.
DndEnd	String	08:00	The end time of the quiet hours. This parameter is presented in the hh:mm format. Valid values of hh: 0 to 23. Valid values of mm: 0 to 59.
DndStart	String	00:00	The start time of the quiet hours. This parameter is presented in the hh:mm format. Valid values of hh: 0 to 23. Valid values of mm: 0 to 59.
Founder	String	9527951795****	The ID of the Apsara Stack tenant account used by the creator of the custom alert rule.

Parameter	Type	Example	Description
MaxAlertTimes	Integer	3	The maximum number of alerts.
Nodes	Array of Nodes	N/A	The nodes to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to NODE.
NodeId	Long	1234	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	9527951795****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.
Projects	Array of Projects	N/A	The workspaces to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to PROJECT.
ProjectId	Long	1234	The ID of the workspace.
RemindId	Long	1234	The ID of the custom alert rule.
RemindName	String	Rule name	The name of the monitoring rule.
RemindType	String	FINISHED	The conditions that trigger an alert. Valid values: FINISHED, UNFINISHED, ERROR, CYCLE_UNFINISHED, and TIMEOUT. A value of FINISHED indicates that the node is run. A value of UNFINISHED indicates that the node is still running at the specified point in time. A value of ERROR indicates that an error occurs when the node is running. A value of CYCLE_UNFINISHED indicates that the node is still running in the specified cycle. A value of TIMEOUT indicates that the node times out.

Parameter	Type	Example	Description
RemindUnit	String	NODE	The type of the object to which the custom alert rule is applied. Valid values: NODE, BASELINE, PROJECT, and BIZPROCESS. A value of NODE indicates that the monitored object is a node. A value of BASELINE indicates that the monitored object is a baseline. A value of PROJECT indicates that the monitored object is a workspace. A value of BIZPROCESS indicates that the monitored object is a workflow.
Robots	Array of Robots	N/A	The webhook URLs of the DingTalk chatbots.
AtAll	Boolean	true	Indicates whether all group members are notified when the alert notification is sent to a DingTalk group. Valid values: true and false.
WebUrl	String	https://oapi.dingtalk.com/robot/send?access_token=***** ***** **	The webhook URL of the DingTalk chatbot.
Useflag	Boolean	true	Indicates whether the custom alert rule is enabled. Valid values: true and false.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFGH-IJKLMNOPQ	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetRemind
&RegionId=cn-shanghai
&RemindId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <AlertInterval>1800</AlertInterval>
  <DndStart>00:00</DndStart>
  <BizProcesses>
    <AtAll>true</AtAll>
    <WebUrl>https://oapi.dingtalk.com/robot/send?access_token=*****</WebUrl>
  >
  </BizProcesses>
  <BizProcesses>
    <Owner>9527951795****</Owner>
    <NodeName>Node name</NodeName>
    <NodeId>1234</NodeId>
    <ProjectId>1234</ProjectId>
  </BizProcesses>
  <BizProcesses>
    <BaselineId>1234</BaselineId>
    <BaselineName>Baseline name</BaselineName>
  </BizProcesses>
  <BizProcesses>
    <ProjectId>1234</ProjectId>
  </BizProcesses>
  <BizProcesses>
    <BizProcessName>Workflow name</BizProcessName>
    <BizId>1234</BizId>
  </BizProcesses>
  <DndEnd>08:00</DndEnd>
  <AlertUnit>OWNER</AlertUnit>
  <AlertMethods>SMS,MAIL</AlertMethods>
  <Projects>
    <AtAll>true</AtAll>
    <WebUrl>https://oapi.dingtalk.com/robot/send?access_token=*****</WebUrl>
  >
  </Projects>
  <Projects>
    <Owner>9527951795****</Owner>
    <NodeName>Node name</NodeName>
    <NodeId>1234</NodeId>
    <ProjectId>1234</ProjectId>
  </Projects>
  <Projects>
    <BaselineId>1234</BaselineId>
    <BaselineName>Baseline name</BaselineName>
  </Projects>
```

```
<Projects>
  <ProjectId>1234</ProjectId>
</Projects>
<Projects>
  <BizProcessName>Workflow name</BizProcessName>
  <BizId>1234</BizId>
</Projects>
<RemindType>FINISHED</RemindType>
<Founder>9527951795****</Founder>
<Nodes>
  <AtAll>true</AtAll>
  <WebUrl>https://oapi.dingtalk.com/robot/send?access_token=*****</WebUrl>
>
</Nodes>
<Nodes>
  <Owner>9527951795****</Owner>
  <NodeName>Node name</NodeName>
  <NodeId>1234</NodeId>
  <ProjectId>1234</ProjectId>
</Nodes>
<Nodes>
  <BaselineId>1234</BaselineId>
  <BaselineName>Baseline name</BaselineName>
</Nodes>
<Nodes>
  <ProjectId>1234</ProjectId>
</Nodes>
<Nodes>
  <BizProcessName>Workflow name</BizProcessName>
  <BizId>1234</BizId>
</Nodes>
<Baselines>
  <AtAll>true</AtAll>
  <WebUrl>https://oapi.dingtalk.com/robot/send?access_token=*****</WebUrl>
>
</Baselines>
<Baselines>
  <Owner>9527951795****</Owner>
  <NodeName>Node name</NodeName>
  <NodeId>1234</NodeId>
  <ProjectId>1234</ProjectId>
</Baselines>
<Baselines>
  <BaselineId>1234</BaselineId>
  <BaselineName>Baseline name</BaselineName>
</Baselines>
<Baselines>
  <ProjectId>1234</ProjectId>
</Baselines>
<Baselines>
  <BizProcessName>Workflow name</BizProcessName>
  <BizId>1234</BizId>
</Baselines>
<RemindId>1234</RemindId>
```

```

<RemindUnit>NODE</RemindUnit>
<AlertTargets>9527952795279528</AlertTargets>
<Useflag>true</Useflag>
<Detail>{"hour":23,"minu":59}</Detail>
<Robots>
  <AtAll>true</AtAll>
  <WebUrl>https://oapi.dingtalk.com/robot/send?access_token=*****</WebUrl
>
</Robots>
<Robots>
  <Owner>9527951795****</Owner>
  <NodeName>Node name</NodeName>
  <NodeId>1234</NodeId>
  <ProjectId>1234</ProjectId>
</Robots>
<Robots>
  <BaselineId>1234</BaselineId>
  <BaselineName>Baseline name</BaselineName>
</Robots>
<Robots>
  <ProjectId>1234</ProjectId>
</Robots>
<Robots>
  <BizProcessName>Workflow name</BizProcessName>
  <BizId>1234</BizId>
</Robots>
<RemindName>Rule name</RemindName>
<MaxAlertTimes>3</MaxAlertTimes>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "AlertInterval": 1800,
    "DndStart": "00:00",
    "BizProcesses": [
      {
        "AtAll": true,
        "WebUrl": "https://oapi.dingtalk.com/robot/send?access_token=*****"
      },
      {
        "Owner": "9527951795****",
        "NodeName": "Node name",
        "NodeId": 1234,
        "ProjectId": 1234
      },
      {
        "BaselineId": 1234.

```



```
    "BaselineName": "Baseline name"
  },
  {
    "ProjectId": 1234
  },
  {
    "BizProcessName": "Workflow name",
    "BizId": 1234
  }
],
"DndEnd": "08:00",
"AlertUnit": "OWNER",
"AlertMethods": "SMS,MAIL",
"Projects": [
  {
    "AtAll": true,
    "WebUrl": "https://oapi.dingtalk.com/robot/send?access_token=*****"
  },
  {
    "Owner": "9527951795****",
    "NodeName": "Node name",
    "NodeId": 1234,
    "ProjectId": 1234
  },
  {
    "BaselineId": 1234,
    "BaselineName": "Baseline name"
  },
  {
    "ProjectId": 1234
  },
  {
    "BizProcessName": "Workflow name",
    "BizId": 1234
  }
],
"RemindType": "FINISHED",
"Founder": "9527951795****",
"Nodes": [
  {
    "AtAll": true,
    "WebUrl": "https://oapi.dingtalk.com/robot/send?access_token=*****"
  },
  {
    "Owner": "9527951795****",
    "NodeName": "Node name",
    "NodeId": 1234,
    "ProjectId": 1234
  },
  {
    "BaselineId": 1234,
    "BaselineName": "Baseline name"
  },
  {

```

```

    "ProjectId": 1234
  },
  {
    "BizProcessName": "Workflow name",
    "BizId": 1234
  }
],
"Baselines": [
  {
    "AtAll": true,
    "WebUrl": "https://oapi.dingtalk.com/robot/send?access_token=*****"
  },
  {
    "Owner": "9527951795****",
    "NodeName": "Node name",
    "NodeId": 1234,
    "ProjectId": 1234
  },
  {
    "BaselineId": 1234,
    "BaselineName": "Baseline name"
  },
  {
    "ProjectId": 1234
  },
  {
    "BizProcessName": "Workflow name",
    "BizId": 1234
  }
],
"RemindId": 1234,
"RemindUnit": "NODE",
"AlertTargets": 9527952795279528,
"Useflag": true,
"Detail": "{\"hour\":\"23\",\"minu\":\"59\"}",
"Robots": [
  {
    "AtAll": true,
    "WebUrl": "https://oapi.dingtalk.com/robot/send?access_token=*****"
  },
  {
    "Owner": "9527951795****",
    "NodeName": "Node name",
    "NodeId": 1234,
    "ProjectId": 1234
  },
  {
    "BaselineId": 1234,
    "BaselineName": "Baseline name"
  },
  {
    "ProjectId": 1234
  },
  {

```

```

    "BizProcessName": "Workflow name",
    "BizId": 1234
  },
  "RemindName": "Rule name",
  "MaxAlertTimes": 3
},
"ErrorCode": 1031203110005,
"ErrorMessage": "The specified parameters are invalid.",
"Success": true
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.

HTTP status code	Error code	Error message	Description
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Invalid.Wkbench.RemindNotExist	The reminder does not exist.	The error message returned because the custom alert rule does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.12. ListReminds

Queries custom alert rules.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListReminds	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
RegionId	String	Yes	cn-shanghai	The ID of the region.
Founder	String	No	9527952795****	The ID of the Apsara Stack tenant account used by the creator of the custom alert rules.

Parameter	Type	Required	Example	Description
NodeId	Long	No	1234	The ID of the node to which the custom alert rules are applied. The ID can be used to search for the custom alert rules that are applied to the node.
RemindTypes	String	No	FINISHED,ERROR	The conditions that trigger an alert. Valid values: FINISHED, UNFINISHED, ERROR, CYCLE_UNFINISHED, and TIMEOUT. A value of FINISHED indicates that the node is run. A value of UNFINISHED indicates that the node is still running at the specified point in time. A value of ERROR indicates that an error occurs when the node is running. A value of CYCLE_UNFINISHED indicates that the node is still running in the specified cycle. A value of TIMEOUT indicates that the node times out. You can specify multiple conditions. Separate them with commas (,).
AlertTarget	String	No	9527952795****	The ID of the Apsara Stack tenant account used by the alert recipient.
SearchText	String	No	Keyword	The keyword in the rule name used to search for the rule. Fuzzy search is supported.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned custom alert rules.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
Reminds	Array of Reminds	N/A	The list of custom alert rules.

Parameter	Type	Example	Description
AlertMethods	List	SMS	The notification method. Valid values: MAIL, SMS, and PHONE. A value of MAIL indicates that the notification is sent by email. A value of SMS indicates that the notification is sent by text message. A value of PHONE indicates that the notification is sent by call, which is supported only by DataWorks Professional Edition and more advanced editions.
AlertTargets	List	952795****	The IDs of the Apsara Stack tenant accounts used by the alert recipients.
AlertUnit	String	OWNER	The alert recipient. Valid values: OWNER and OTHER. A value of OWNER indicates that the alert notification is sent to the node owner. A value of OTHER indicates that the alert notification is sent to a specific user.
Baselinelds	List	9527	The IDs of the baselines to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to BASELINE.
BizProcessIds	List	9527	The IDs of the workflows to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to BIZPROCESS.
DndEnd	String	08:00	The end time of the quiet hours. This parameter is presented in the hh:mm format. Valid values of hh: 0 to 23. Valid values of mm: 0 to 59.
DndStart	String	00:00	The start time of the quiet hours. This parameter is presented in the hh:mm format. Valid values of hh: 0 to 23. Valid values of mm: 0 to 59.
Founder	String	952795****	The ID of the Apsara Stack tenant account used by the creator of the custom alert rule.

Parameter	Type	Example	Description
Nodelds	List	1234	The IDs of the nodes to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to NODE.
ProjectIds	List	1234	The IDs of the workspaces to which the custom alert rule is applied. This parameter is returned if the RemindUnit parameter is set to PROJECT.
RemindId	Long	1234	The ID of the custom alert rule.
RemindName	String	Rule name	The name of the custom alert rule.
RemindType	String	FINISHED	The conditions that trigger an alert. Valid values: FINISHED, UNFINISHED, ERROR, CYCLE_UNFINISHED, and TIMEOUT. A value of FINISHED indicates that the node is run. A value of UNFINISHED indicates that the node is still running at the specified point in time. A value of ERROR indicates that an error occurs when the node is running. A value of CYCLE_UNFINISHED indicates that the node is still running in the specified cycle. A value of TIMEOUT indicates that the node times out.
RemindUnit	String	NODE	The type of the object to which the custom alert rule is applied. Valid values: NODE, BASELINE, PROJECT, and BIZPROCESS. A value of NODE indicates that the monitored object is a node. A value of BASELINE indicates that the monitored object is a baseline. A value of PROJECT indicates that the monitored object is a workspace. A value of BIZPROCESS indicates that the monitored object is a workflow.
Useflag	Boolean	true	Indicates whether the custom alert rule is enabled. Valid values: true and false.
TotalCount	Integer	100	The total number of custom alert rules returned.
ErrorCode	String	1031203110005	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListReminds
&PageNumber=1
&PageSize=10
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format


```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Reminds>
    <RemindId>1234</RemindId>
    <DndStart>00:00</DndStart>
    <DndEnd>08:00</DndEnd>
    <AlertUnit>OWNER</AlertUnit>
    <RemindUnit>NODE</RemindUnit>
    <RemindType>FINISHED</RemindType>
    <Useflag>true</Useflag>
    <Founder>952795****</Founder>
    <RemindName>Rule name</RemindName>
  </Reminds>
  <Reminds>
    <BizProcessIds>1234</BizProcessIds>
    <NodeIds>1234</NodeIds>
    <AlertMethods>SMS</AlertMethods>
    <ProjectIds>1234</ProjectIds>
    <BaselineIds>1234</BaselineIds>
    <AlertTargets>952795****</AlertTargets>
  </Reminds>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "Reminds": [
      {
        "RemindId": 1234,
        "DndStart": "00:00",
        "DndEnd": "08:00",
        "AlertUnit": "OWNER",
        "RemindUnit": "NODE",
        "RemindType": "FINISHED",
        "Useflag": true,
        "Founder": "952795****",
        "RemindName": "Rule name"
      },
      {
        "BizProcessIds": 1234,
        "NodeIds": 1234,
        "AlertMethods": "SMS",
        "ProjectIds": 1234,
        "BaselineIds": 1234,
        "AlertTargets": "952795****"
      }
    ]
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Serve	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
500	InternalError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.13. ListTopics

Queries events.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListTopics	The operation that you want to perform.

Parameter	Type	Required	Example	Description
BeginTime	String	Yes	2019-03-24T00:00:00+0800	The beginning of the time range to query. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
EndTime	String	Yes	2019-03-25T23:00:00+0800	The end of the time range to query. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
PageNumber	Integer	Yes	1	The number of the page to return. Valid values: 1 to 30. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TopicTypes	String	No	ERROR,SLOW	The type of the event. Valid values: SLOW and ERROR. A value of SLOW indicates that the duration of the task is significantly longer than the average duration of the task in previous cycles. A value of ERROR indicates that the task fails to run. You can specify multiple types. Separate them with commas (,).
TopicStatuses	String	No	IGNORE,NEW,FIXING,RECOVER	The status of the event. Valid values: IGNORE, NEW, FIXING, and RECOVER. A value of IGNORE indicates that the event is ignored. A value of NEW indicates that the event is a new event. A value of FIXING indicates that the event is being processed. A value of RECOVER indicates that the event is processed. You can specify multiple states. Separate them with commas (,).

Parameter	Type	Required	Example	Description
Nodeld	Long	No	1234	The ID of the node that triggers the event. You can specify either this parameter or the InstanceId parameter, but not both.
InstanceId	Long	No	12345	The ID of the node instance that triggers the event. You can specify either this parameter or the Nodeld parameter, but not both.
Owner	String	No	952795****	The ID of the Apsara Stack tenant account used by the event owner.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned events.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
Topics	Array of Topics	N/A	The list of events.
AddTime	Long	1553524393000	The timestamp when the event was found.
FixTime	Long	1553508465000	The timestamp when the event was processed.
HappenTime	Long	1553508465000	The timestamp when the event occurred. A time difference may exist between the time when the event occurred and the time when the event was found.
InstanceId	Long	12345	The ID of the node instance that triggered the event.
Nodeld	Long	1234	The ID of the node that triggered the event.

Parameter	Type	Example	Description
NodeName	String	Node name	The name of the node.
NodeOwner	String	952795****	The ID of the Apsara Stack tenant account used by the node owner.
ProjectId	Long	1234	The ID of the workspace to which the node belongs.
TopicId	Long	1234	The ID of the event.
TopicName	String	1234 error	The name of the event.
TopicStatus	String	NEW	The status of the event. Valid values: IGNORE, NEW, FIXING, and RECOVER. A value of IGNORE indicates that the event is ignored. A value of NEW indicates that the event is a new event. A value of FIXING indicates that the event is being processed. A value of RECOVER indicates that the event is processed.
TopicType	String	ERROR	The type of the event. Valid values: SLOW and ERROR. A value of SLOW indicates that the duration of the task is significantly longer than the average duration of the task in previous cycles. A value of ERROR indicates that the task fails to run.
TotalCount	Integer	100	The total number of events returned.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.

Parameter	Type	Example	Description
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListTopics
&BeginTime=2019-03-24T00:00:00+0800
&EndTime=2019-03-25T23:00:00+0800
&PageNumber=1
&PageSize=10
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <Topics>
    <TopicType>ERROR</TopicType>
    <FixTime>1553508465000</FixTime>
    <NodeOwner>952795****</NodeOwner>
    <InstanceId>12345</InstanceId>
    <NodeName>Node name</NodeName>
    <NodeId>1234</NodeId>
    <ProjectId>1234</ProjectId>
    <TopicId>1234</TopicId>
    <TopicStatus>NEW</TopicStatus>
    <AddTime>1553524393000</AddTime>
    <HappenTime>1553508465000</HappenTime>
    <TopicName>1234 error</TopicName>
  </Topics>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "Topics": {
      "TopicType": "ERROR",
      "FixTime": 1553508465000,
      "NodeOwner": "952795****",
      "InstanceId": 12345,
      "NodeName": "Node name",
      "NodeId": 1234,
      "ProjectId": 1234,
      "TopicId": 1234,
      "TopicStatus": "NEW",
      "AddTime": 1553524393000,
      "HappenTime": 1553508465000,
      "TopicName": "1234 error"
    },
    "PageSize": 10,
    "PageNumber": 1
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Servicer	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.

HTTP status code	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.14. GetTopic

Queries the details of an event.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetTopic	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TopicId	Long	Yes	1234	The ID of the event.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the event.
AddTime	Long	1553524393000	The timestamp when the event was found.
AlertTime	Long	1553524393000	The timestamp when the first alert was reported.
Assigner	String	952795****	The ID of the Apsara Stack tenant account used by the alert recipient.
BaselineBuffer	Long	3600	The margin of the worst baseline instance. Unit: seconds.
BaselineId	Long	1234	The ID of the baseline to which the worst baseline instance belongs.
BaselineInGroupId	Integer	1	The ID of the cycle of the worst baseline instance.
BaselineName	String	Baseline name	The name of the baseline to which the worst baseline instance belongs.
BaselineStatus	String	SAFE	The status of the baseline. Valid values: ERROR, SAFE, DANGROUS, and OVER. A value of ERROR indicates that no nodes are associated with the baseline, or all nodes associated with the baseline are suspended. A value of SAFE indicates that nodes are run before the alert duration begins. A value of DANGROUS indicates that nodes are still running after the alert duration ends but the committed time does not arrive. A value of OVER indicates that nodes are still running after the committed time.

Parameter	Type	Example	Description
Buffer	Long	1200	The margin of the event. Unit: seconds.
DealTime	Long	1553524393000	The timestamp when the event was last processed.
DealUser	String	952795****	The ID of the Apsara Stack tenant account used by the user who last processed the event.
FixTime	Long	1553524393000	The timestamp when the event was processed.
HappenTime	Long	1553524393000	The timestamp when the event occurred. A time difference may exist between the time when the event occurred and the time when the event was found.
InstancelId	Long	12345	The ID of the node instance that triggered the event.
NextAlertTime	Long	1553524393000	The timestamp when the system reports the next alert.
NodeId	Long	1234	The ID of the node that triggered the event.
NodeName	String	Node name	The name of the node that triggered the event.
Owner	String	952795****	The ID of the Apsara Stack tenant account used by the event owner.
ProjectId	Long	1234	The ID of the workspace to which the node that triggered the event belongs.
TopicId	Long	1234	The ID of the event.
TopicName	String	1234 error	The name of the event.

Parameter	Type	Example	Description
TopicStatus	String	FIXING	The status of the event. Valid values: IGNORE, NEW, FIXING, and RECOVER. A value of IGNORE indicates that the event is ignored. A value of NEW indicates that the event is a new event. A value of FIXING indicates that the event is being processed. A value of RECOVER indicates that the event is processed.
TopicType	String	ERROR	The type of the event. Valid values: SLOW and ERROR. A value of SLOW indicates that the duration of the task is significantly longer than the average duration of the task in previous cycles. A value of ERROR indicates that the task fails to run.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFGH-IJKLMNOPQ	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetTopic
&RegionId=cn-shanghai
&TopicId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <BaselineId>1234</BaselineId>
  <Buffer>1200</Buffer>
  <FixTime>1553524393000</FixTime>
  <Owner>952795****</Owner>
  <InstanceId>12345</InstanceId>
  <NodeName>Node name</NodeName>
  <BaselineStatus>SAFE</BaselineStatus>
  <ProjectId>1234</ProjectId>
  <TopicStatus>FIXING</TopicStatus>
  <AddTime>1553524393000</AddTime>
  <HappenTime>1553524393000</HappenTime>
  <AlertTime>1553524393000</AlertTime>
  <BaselineInGroupId>1</BaselineInGroupId>
  <TopicName>1234 error</TopicName>
  <TopicType>ERROR</TopicType>
  <Assigner>9527952795279528</Assigner>
  <DealUser>9527952795279528</DealUser>
  <NextAlertTime>1553524393000</NextAlertTime>
  <BaselineName>Baseline name</BaselineName>
  <NodeId>1234</NodeId>
  <TopicId>1234</TopicId>
  <BaselineBuffer>3600</BaselineBuffer>
  <DealTime>1553524393000</DealTime>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "BaselineId": 1234,
    "Buffer": 1200,
    "FixTime": 1553524393000,
    "Owner": "952795****",
    "InstanceId": 12345,
    "NodeName": "Node name",
    "BaselineStatus": "SAFE",
    "ProjectId": 1234,
    "TopicStatus": "FIXING",
    "AddTime": 1553524393000,
    "HappenTime": 1553524393000,
    "AlertTime": 1553524393000,
    "BaselineInGroupId": 1,
    "TopicName": "1234 error",
    "TopicType": "ERROR",
    "Assigner": 9527952795279528,
    "DealUser": 9527952795279528,
    "NextAlertTime": 1553524393000,
    "BaselineName": "Baseline name",
    "NodeId": 1234,
    "TopicId": 1234,
    "BaselineBuffer": 3600,
    "DealTime": 1553524393000
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.TopicNotExist	The topic does not exist.	The error message returned because the event does not exist.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.15. GetTopicInfluence

Queries baseline instances affected by an event.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetTopicInfluence	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TopicId	Long	Yes	1234	The ID of the event.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of baseline instances affected by the event.
Influences	Array of Influences	N/A	The list of affected baseline instances.
BaselineId	Long	1234	The ID of the baseline.
BaselineName	String	Baseline name	The name of the baseline.
Bizdate	Long	1553356800000	The data timestamp of the baseline instance.
Buffer	Long	360	The margin of the baseline instance. Unit: seconds.
InGroupId	Integer	1	The ID of the cycle of the baseline instance. Valid values of the ID of an hour-level cycle: 1 to 24. The ID of a day-level cycle is 1.
Owner	String	952795****	The ID of the Apsara Stack tenant account used by the baseline owner. Multiple IDs are separated by commas (,).
Priority	Integer	1	The priority of the baseline. Valid values: 1, 2, 5, 7, and 8.
ProjectId	Long	1234	The ID of the workspace to which the baseline belongs.

Parameter	Type	Example	Description
Status	String	SAFE	The status of the baseline. Valid values: ERROR, SAFE, DANGROUS, and OVER. A value of ERROR indicates that no nodes are associated with the baseline, or all nodes associated with the baseline are suspended. A value of SAFE indicates that nodes are run before the alert duration begins. A value of DANGROUS indicates that nodes are still running after the alert duration ends but the committed time does not arrive. A value of OVER indicates that nodes are still running after the committed time.
TopicId	Long	1234	The ID of the event.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetTopicInfluence
&RegionId=cn-shanghai
&TopicId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Influences>
    <BaselineId>1234</BaselineId>
    <Status>SAFE</Status>
    <Buffer>360</Buffer>
    <Owner>952795****</Owner>
    <Priority>1</Priority>
    <BaselineName>Baseline name</BaselineName>
    <ProjectId>1234</ProjectId>
    <Bizdate>1553356800000</Bizdate>
    <InGroupId>1</InGroupId>
  </Influences>
  <TopicId>1234</TopicId>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "Influences": {
      "BaselineId": 1234,
      "Status": "SAFE",
      "Buffer": 360,
      "Owner": "952795****",
      "Priority": 1,
      "BaselineName": "Baseline name",
      "ProjectId": 1234,
      "Bizdate": 1553356800000,
      "InGroupId": 1
    },
    "TopicId": 1234
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Wkbench.ServiceError	An internal error occurred.	The error message returned because a service error has occurred.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
500	Invalid.Wkbench.TopicNotExist	The topic does not exist.	The error message returned because the event does not exist.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.16. GetNode

Queries the details of a node.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetNode	The operation that you want to perform.
NodeId	Long	Yes	1234	The ID of the node.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the node.
BaselineId	Long	123456	The ID of the baseline.
Connection	String	odps_first_dev	The connection string.

Parameter	Type	Example	Description
CronExpress	String	00 00 00 * * ?	The CRON expression.
Description	String	test	The description of the node.
DqcDescription	String	[{"projectName":"test_0923001","tableName":"test_table_001","partition":"ds\u003d\${yyyymmdd}"},"{"projectName":"test_0923001","tableName":"test_table_002","partition":"NOT APARTITIONTABLE"}]	The table and partition filter expression in Data Quality that are associated with the node.
DqcType	Integer	1	Indicates whether the node is associated with Data Quality. Valid values: 0 and 1. A value of 0 indicates that the node is associated with Data Quality. A value of 1 indicates that the node is not associated with Data Quality.
NodeId	Long	1234	The ID of the node.
NodeName	String	SQL node	The name of the node.
OwnerId	String	17366294****	The ID of the node owner.
ParamValues	String	a=b	The other parameters.
Priority	Integer	1	The priority of the node. Valid values: 1, 3, 5, 7, and 8.
ProgramType	String	ODPS_SQL	The type of the node.
ProjectId	Long	1234	The ID of the workspace.
RelatedFlowId	Long	123	The ID of the workflow to which the node belongs.

Parameter	Type	Example	Description
RepeatInterval	Long	60	Indicates the interval at which the node is rerun after the node fails to run.
Repeatability	String	true	Indicates whether the node can be rerun.
ResGroupName	String	Default resource group	The name of the resource group.
SchedulerType	String	NORMAL	The scheduling type of the node. Valid values: NORMAL, MANUAL, PAUSE, and SKIP. A value of NORMAL indicates that the scheduling system normally runs the node based on the specified recurrence. A value of MANUAL indicates that the node is only manually run. A value of PAUSE indicates that the scheduling system runs the node based on the specified recurrence but directly returns a failure response. A value of SKIP indicates that the scheduling system does not actually run the node but directly returns a success response.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetNode
&NodeId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Repeatability>true</Repeatability>
  <Description>Description</Description>
  <ResGroupName>Default resource group</ResGroupName>
  <OwnerId>173662940****</OwnerId>
  <NodeName>SQL node</NodeName>
  <ParamValues>a=b</ParamValues>
  <NodeId>1234</NodeId>
  <ProjectId>1234</ProjectId>
  <ProgramType>ODPS_SQL</ProgramType>
  <SchedulerType>NORMAL</SchedulerType>
  <CronExpress>E6F0DBDD-5AD****</CronExpress>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": {
    "Repeatability": true,
    "Description": "Description",
    "ResGroupName": "Default resource group",
    "OwnerId": "173662940****",
    "NodeName": "SQL node",
    "ParamValues": "a=b",
    "NodeId": 1234,
    "ProjectId": 1234,
    "ProgramType": "ODPS_SQL",
    "SchedulerType": "NORMAL",
    "CronExpress": "E6F0DBDD-5AD****"
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.17. ListNodeInputOrOutput

Queries the ancestor or descendant nodes of a node. Only the ancestor or descendant nodes at the nearest level can be queried each time. Such nodes are also known as parent or child nodes.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListNodeInputOrOutput	The operation that you want to perform.
IoType	String	Yes	output	The type of node that you want to query. Valid values: - input: ancestor nodes - output: descendant nodes
NodeId	Long	Yes	12314567	The ID of the node. You can call the ListNodes operation to query the node ID.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: DEV and PROD. A value of DEV indicates the development environment. A value of PROD indicates the production environment.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Array of Data	N/A	The list of ancestor or descendant nodes.
Data	String	xxxx.123141254_out	The name of the ancestor or descendant node.
NodeId	Long	1234667	The ID of the node.
TableName	String	dwd_xxx_table	The name of the associated table.
ErrorCode	String	Invalid.Tenant.ProjectNotExists	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	The project does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListNodeInputOrOutput
&IoType=output
&NodeId=12314567
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TableName>dwd_xxx_table</TableName>
  <NodeId>1234667</NodeId>
  <Data>xxx.123141254_out</Data>
</Data>
<ErrorCode>Invalid.Tenant.ProjectNotExists</ErrorCode>
<ErrorMessage>The project does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": {
    "TableName": "dwd_xxx_table",
    "NodeId": 1234667,
    "Data": "xxxx.123141254_out"
  },
  "ErrorCode": "Invalid.Tenant.ProjectNotExists",
  "ErrorMessage": "The project does not exist.",
  "Success": true
}
```

2.1.6.4.18. ListNodes

Queries nodes.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListNodes	The operation that you want to perform.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: DEV and PROD.
ProjectId	Long	Yes	1234	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
Owner	String	No	193379****	The ID of the node owner.
BizName	String	No	test_bizName	The name of the workflow.
ProgramType	String	No	ODPS_SQL	The type of the node.
PageNumber	Integer	No	1	The number of the page to return.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
NodeName	String	No	liux_test_n****	The name of the node.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of nodes.
Nodes	Array of Nodes	N/A	The information about nodes.
BaselineId	Long	123456	The ID of the baseline.
Connection	String	odps_first	The connection string.
CronExpress	String	00 00 00 * * ?	The CRON expression.
Description	String	test	The description of the node.
DqcDescription	String	[{"projectName": "ztjy_dim", "tableName": "dim_user_agent_manage_area_a", "partition": "ds\u003d\$[yyyy-mm-dd-1]"}]	The table and partition filter expression in Data Quality that are associated with the node.
DqcType	Integer	1	Indicates whether the node is associated with Data Quality. Valid values: 0 and 1. A value of 0 indicates that the node is associated with Data Quality. A value of 1 indicates that the node is not associated with Data Quality.
NodeId	Long	1234	The ID of the node.

Parameter	Type	Example	Description
NodeName	String	liux_test_n****	The name of the node.
OwnerId	String	19337906836551	The ID of the node owner.
ParamValues	String	a=b	The other parameters.
Priority	Integer	1	The priority of the node. Valid values: 1, 3, 5, 7, and 8.
ProgramType	String	ODPS_SQL	The type of the node.
ProjectId	Long	33671	The ID of the workspace.
RelatedFlowId	Long	1231123	The ID of the workflow to which the node belongs.
RepeatInterval	Long	60	Indicates the interval at which the node is rerun after the node fails to run.
Repeatability	Boolean	true	Indicates whether the node can be rerun.
ResGroupName	String	Default resource group	The name of the resource group.
SchedulerType	String	NORMAL	The scheduling type of the node. Valid values: NORMAL, MANUAL, PAUSE, and SKIP. A value of NORMAL indicates that the scheduling system normally runs the node based on the specified recurrence. A value of MANUAL indicates that the node is only manually run. A value of PAUSE indicates that the scheduling system runs the node based on the specified recurrence but directly returns a failure response. A value of SKIP indicates that the scheduling system does not actually run the node but directly returns a success response.
PageNumber	Integer	1	The page number of the returned page.

Parameter	Type	Example	Description
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	66	The total number of nodes returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListNodes
&ProjectEnv=PROD
&ProjectId=1234
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>66</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Nodes>
    <BaselineId>123456</BaselineId>
    <ResGroupName>Default resource group</ResGroupName>
    <Description>test</Description>
    <NodeName>liux_test_n****</NodeName>
    <DqcType>1</DqcType>
    <Priority>1</Priority>
    <ParamValues>a=b</ParamValues>
    <Connection>odps_first</Connection>
    <ProjectId>33671</ProjectId>
    <RelatedFlowId>1231123</RelatedFlowId>
    <SchedulerType>NORMAL</SchedulerType>
    <CronExpress>00 00 00 * * ?</CronExpress>
    <Repeatability>true</Repeatability>
    <OwnerId>19337906836551</OwnerId>
    <DqcDescription>[{"projectName":"ztjy_dim","tableName":"dim_user_agent_manage_area_a","partition":"ds\u003d${yyyy-mm-dd-1}"}]</DqcDescription>
    <NodeId>1234</NodeId>
    <ProgramType>ODPS_SQL</ProgramType>
    <RepeatInterval>60</RepeatInterval>
  </Nodes>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>

```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 66,
    "PageSize": 10,
    "PageNumber": 1,
    "Nodes": {
      "BaselineId": 123456,
      "ResGroupName": "Default resource group",
      "Description": "test",
      "NodeName": "liux_test_n****",
      "DqcType": 1,
      "Priority": 1,
      "ParamValues": "a=b",
      "Connection": "odps_first",
      "ProjectId": 33671,
      "RelatedFlowId": 1231123,
      "SchedulerType": "NORMAL",
      "CronExpress": "00 00 00 * * ?",
      "Repeatability": true,
      "OwnerId": 19337906836551,
      "DqcDescription": "[{"projectName\\":\\"ztjy_dim\\",\\"tableName\\":\\"dim_user_agent_manage_area_a\\",\\"partition\\":\\"ds\\u003d${yyyy-mm-dd-1}\\"}]",
      "NodeId": 1234,
      "ProgramType": "ODPS_SQL",
      "RepeatInterval": 60
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.19. GetNodeCode

Queries the code of a node.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetNodeCode	The operation that you want to perform.
NodeId	Long	Yes	1234	The ID of the node.

Parameter	Type	Required	Example	Description
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	String	select a;	The code of the node.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetNodeCode
&NodeId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format


```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>select a;</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": "select a;",
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.20. GetInstance

Queries the details of an instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetInstance	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance. You can call the ListInstances operation to query the instance ID.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the instance.
BaselineId	Long	123123	The ID of the baseline.
BeginRunningTime	Long	1590416703313	The time when the instance started to run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
BeginWaitResTime	Long	1590416703313	The time when the instance started to wait for resources. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
BeginWaitTime	Long	1590416703313	The time when the instance started to wait to be scheduled. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Bizdate	Long	1590336000000	The data timestamp of the instance. In most cases, the value indicates one day before the time when the instance is run.
Connection	String	odps_first	The connection string.
CreateTime	Long	1590416703313	The time when the instance was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CycTime	Long	1590422400000	The time when the instance was scheduled to run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
DagId	Long	338450167	The ID of the workflow.

Parameter	Type	Example	Description
DagType	String	DAILY	The type of the workflow. Valid values: - DAILY: The workflow is used to run auto triggered nodes. - MANUAL: The workflow is used to run manually triggered nodes. - SMOKE_TEST: The workflow is used to perform smoke testing. - SUPPLY_DATA: The workflow is used to generate retroactive data.
DqcDescription	String	<pre>[{"projectName": "ztjy_dim", "tableName": "dim_user_agent_manage_area_a", "partition": "ds\u003d\$[yyyy-mm-dd-1]"}]</pre>	The table and partition filter expression in Data Quality that are associated with the instance.
DqcType	Integer	1	Indicates whether the instance is associated with Data Quality. Valid values: 0: The instance is associated with Data Quality. 1: The instance is not associated with Data Quality.
FinishTime	Long	1590416703313	The time when the instance stopped running. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Instanceld	Long	11713307578	The ID of the instance.
ModifyTime	Long	1590416703313	The time when the instance was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
NodeId	Long	33115	The ID of the node.
NodeName	String	kzh	The name of the node.

Parameter	Type	Example	Description
ParamValues	String	bizdate=\$bizdate tbods=\$tbods tbdw=\$tbdw tbpmic=\$tbpmic tbpidx=\$tbpidx tbptcif=\$tbptcif	The parameters related to the node.
Priority	Integer	1	The priority of the instance. Valid values: 1, 3, 5, 7, and 8. A greater value indicates a higher priority. Default value: 1.
RelatedFlowId	Long	123123	The ID of the workflow to which the node belongs.
RepeatInterval	Long	60000	The interval at which the node is rerun after the node fails to run. Unit: milliseconds.
Repeatability	Boolean	true	Indicates whether the node can be rerun.
Status	String	NOT_RUN	The status of the node. Valid values: • NOT_RUN: The node is not run. • WAIT_TIME: The node is waiting for its scheduled time to arrive. • WAIT_RESOURCE: The node is waiting for resources. • RUNNING: The node is running. • CHECKING: Data quality is being checked for the node. • CHECKING_CONDITION: Branch conditions are being checked for the node. • FAILURE: The node fails to run. • SUCCESS: The node is run.

Parameter	Type	Example	Description
TaskRerunTime	Integer	0	The number of times the node that corresponds to the instance can be rerun. This parameter can be left empty, or the value of this parameter can be an integer that is greater than or equal to 0. • If this parameter is left empty, the number of rerun times is not specified for the node. • If the value of this parameter is 0, the node cannot be rerun. • If the value of this parameter is n, which is an integer greater than 0, the node can be rerun n times. For example, if the value of this parameter is 1, the node can be rerun once. If the value of this parameter is 2, the node can be rerun twice.

Parameter	Type	Example	Description
TaskType	String	NORMAL(0)	The scheduling type of the node. Valid values: - NORMAL(0): The node is an auto triggered node. It is run on a regular basis. - MANUAL(1): The node is a manually triggered node. It is not run on a regular basis. - PAUSE(2): The node is a frozen node. The scheduling system still runs the node but sets it to FAILURE when the scheduling system starts to run the node. - SKIP(3): The node is a dry-run node. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node. - SKIP_UNCHOOSE(4): The node is an unselected node in a temporary workflow. This type of node exists only in temporary workflows. The scheduling system sets the node to SUCCESS when the scheduling system starts to run the node. - SKIP_CYCLE(5): The node is a weekly or monthly node whose scheduled time does not arrive. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node. - CONDITION_UNCHOOSE(6): The node is not selected by its ancestor branch node and is run as a dry-run node. - REALTIME_DEPRECATED(7): The node has instances generated in real time but deprecated. The scheduling system directly sets it to SUCCESS.
ErrorCode	String	0	The error code returned.
ErrorMessage	String	test	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetInstance
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <BaselineId>123123</BaselineId>
  <ModifyTime>1590416703313</ModifyTime>
  <NodeName>kzh</NodeName>
  <DagType>DAILY</DagType>
  <ParamValues>bizdate=$bizdate tbods=$tbods tbdw=$tbdw tbpmic=$tbpmic tbpidx=$tbpidx tbptcif=$tbptcif</ParamValues>
  <DagId>338450167</DagId>
  <BeginRunningTime>1590416703313</BeginRunningTime>
  <Bizdate>1590336000000</Bizdate>
  <FinishTime>1590416703313</FinishTime>
  <DqcDescription>[{"projectName":"ztjy_dim","tableName":"dim_user_agent_manage_area_a","partition":"ds\u003d${yyyy-mm-dd-1}"}]</DqcDescription>
  <Status>NOT_RUN</Status>
  <TaskType>NORMAL(0)</TaskType>
  <TaskRetunTime>0</TaskRetunTime>
  <InstanceId>11713307578</InstanceId>
  <DqcType>1</DqcType>
  <Priority>1</Priority>
  <CreateTime>1590416703313</CreateTime>
  <Connection>odps_first</Connection>
  <RelatedFlowId>123123</RelatedFlowId>
  <BeginWaitTimeTime>1590416703313</BeginWaitTimeTime>
  <CycTime>1590422400000</CycTime>
  <Repeatability>true</Repeatability>
  <NodeId>33115</NodeId>
  <BeginWaitResTime>1590416703313</BeginWaitResTime>
  <ErrorMessage>error message</ErrorMessage>
  <RepeatInterval>60000</RepeatInterval>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>test</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": {
    "BaselineId": 123123,
    "ModifyTime": 1590416703313,
    "NodeName": "kzh",
    "DagType": "DAILY",
    "ParamValues": "bizdate=$bizdate tbods=$tbods tbdw=$tbdw tbpmic=$tbpmic tbpidx=$tbpidx tbptcif=$tbptcif",
    "DagId": 338450167,
    "BeginRunningTime": 1590416703313,
    "Bizdate": 1590336000000,
    "FinishTime": 1590416703313,
    "DqcDescription": "[{"projectName\\":\\"ztjy_dim\\",\\"tableName\\":\\"dim_user_agent_manage_area_a\\",\\"partition\\":\\"ds\\u003d${yyyy-mm-dd-1}\\"}]",
    "Status": "NOT_RUN",
    "TaskType": "NORMAL(0)",
    "TaskRetunTime": 0,
    "InstanceId": 11713307578,
    "DqcType": 1,
    "Priority": 1,
    "CreateTime": 1590416703313,
    "Connection": "odps_first",
    "RelatedFlowId": 123123,
    "BeginWaitTime": 1590416703313,
    "CycTime": 1590422400000,
    "Repeatability": true,
    "NodeId": 33115,
    "BeginWaitResTime": 1590416703313,
    "ErrorMessage": "error message",
    "RepeatInterval": 60000
  },
  "ErrorCode": 0,
  "ErrorMessage": "test",
  "Success": true
}
```

2.1.6.4.21. ListInstances

Queries instances.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListInstances	The operation that you want to perform.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
ProjectId	Long	Yes	12345	The ID of the workspace. You can call the ListProjects operation to query the workspace ID.
NodeId	Long	Yes	100000000000	The ID of the node. You can call the ListNodes operation to query the node ID.
NodeName	String	No	openmr_8****	The name of the node. You can call the ListNodes operation to query the node name.
Owner	String	No	193379****	The ID of the Apsara Stack tenant account used by the workspace administrator. You can log on to the Apsara Uni-manager Management Console and view the ID in the Basic Information section of the User Center console.
BizName	String	No	test_bizName	The name of the workflow. You can call the ListBusiness operation to query the workflow name.
ProgramType	String	No	ODPS_SQL	The type of the node. You can call the ListNodes operation to query the node type.
PageNumber	Integer	No	1	The number of the page to return. Minimum value: 1. Maximum value: 100.

Parameter	Type	Required	Example	Description
PageSize	Integer	No	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
DagId	Long	No	11111	The ID of the directed acyclic graph (DAG). You can set this parameter to the value of the DagId parameter returned by the CreateDagComplement, CreateTest, or CreateManualDag operation.
Bizdate	String	No	2020-02-02 00:00:00	The data timestamp of the instances that you want to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.
BeginBizdate	String	No	2020-02-02 00:00:00	The beginning of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.
EndBizdate	String	No	2020-02-03 00:00:00	The end of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of instances.
Instances	Array of Instance	N/A	The information about instances.
BaselineId	Long	123123	The ID of the baseline.
BeginRunningTime	Long	1590416703313	The time when the instance started to run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.

Parameter	Type	Example	Description
BeginWaitResTime	Long	1590416703313	The time when the instance started to wait for resources. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
BeginWaitTime	Long	1590416703313	The time when the instance started to wait to be scheduled. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Bizdate	Long	1590336000000	The data timestamp of the instance. In most cases, the value indicates one day before the time when the instance is run.
Connection	String	odps_first	The connection string.
CreateTime	Long	1590416703313	The time when the instance was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CycTime	Long	1590422400000	The time when the node was scheduled to run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
DagId	Long	33845	The ID of the workflow.
DagType	String	DAILY	The type of the workflow. Valid values: • DAILY: The workflow is used to run auto triggered nodes. • MANUAL: The workflow is used to run manually triggered nodes. • SMOKE_TEST: The workflow is used to perform smoke testing. • SUPPLY_DATA: The workflow is used to generate retroactive data.

Parameter	Type	Example	Description
DqcDescription	String	<code>[{"projectName": "ztjy_dim", "tableName": "dim_user_agent_manage_area_a", "partition": "ds\u003d\$[yyyy-mm-dd-1]"}]</code>	The table and partition filter expression in Data Quality that are associated with the instance.
DqcType	Integer	1	Indicates whether the instance is associated with Data Quality. Valid values: 0: The instance is associated with Data Quality. 1: The instance is not associated with Data Quality.
ErrorMessage	String	error message	The error message that is returned for the instance. This parameter is deprecated. You can call the <code>GetInstanceLog</code> operation to query the error information related to the node.
FinishTime	Long	1590416703313	The time when the node stopped running. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Instanceld	Long	1234	The ID of the instance.
ModifyTime	Long	1590416703313	The time when the node was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Nodeld	Long	33115	The ID of the node.
NodeName	String	kzh	The name of the node.
ParamValues	String	<code>bizdate=\$bizdate tbods=\$tbods</code>	The parameters related to the node.
Priority	Integer	1	The priority of the instance. Valid values: 1, 3, 5, 7, and 8. A greater value indicates a higher priority. Default value: 1.

Parameter	Type	Example	Description
RelatedFlowId	Long	123456	The ID of the workflow to which the node belongs.
RepeatInterval	Long	60000	The interval at which the node is rerun after the node fails to run. Unit: milliseconds.
Repeatability	Boolean	true	Indicates whether the node can be rerun.
Status	String	NOT_RUN	The status of the node. Valid values: • NOT_RUN: The node is not run. • WAIT_TIME: The node is waiting for its scheduled time to arrive. • WAIT_RESOURCE: The node is waiting for resources. • RUNNING: The node is running. • CHECKING: Data quality is being checked for the node. • CHECKING_CONDITION: Branch conditions are being checked for the node. • FAILURE: The node fails to run. • SUCCESS: The node is run.
TaskRerunTime	Integer	0	The number of times the node that corresponds to the instance can be rerun. This parameter can be left empty, or the value of this parameter can be an integer that is greater than or equal to 0. • If this parameter is left empty, the number of rerun times is not specified for the node. • If the value of this parameter is 0, the node cannot be rerun. • If the value of this parameter is n, which is an integer greater than 0, the node can be rerun n times. For example, if the value of this parameter is 1, the node can be rerun once. If the value of this parameter is 2, the node can be rerun twice.

Parameter	Type	Example	Description
TaskType	String	NORMAL(0)	The scheduling type of the node. Valid values: • NORMAL(0): The node is an auto triggered node. It is run on a regular basis. • MANUAL(1): The node is a manually triggered node. It is not run on a regular basis. • PAUSE(2): The node is a frozen node. The scheduling system still runs the node but sets it to FAILURE when the scheduling system starts to run the node. • SKIP(3): The node is a dry-run node. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node. • SKIP_UNCHOOSE(4): The node is an unselected node in a temporary workflow. This type of node exists only in temporary workflows. The scheduling system sets the node to SUCCESS when the scheduling system starts to run the node. • SKIP_CYCLE(5): The node is a weekly or monthly node whose scheduled time does not arrive. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node. • CONDITION_UNCHOOSE(6): The node is not selected by its ancestor branch node and is run as a dry-run node. REALTIME_DEPRECATED(7): The node has instances generated in real time but deprecated. The scheduling system directly sets it to SUCCESS.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	66	The total number of instances returned.
ErrorCode	String	Invalid.Tenant.ProjectNotExists	The error code returned.
ErrorMessage	String	The project does not exist.	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListInstances
&RegionId=cn-shanghai
&ProjectEnv=PROD
&ProjectId=12345
&NodeId=100000000000
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Instances>
    <BaselineId>123123</BaselineId>
    <ModifyTime>1590416703313</ModifyTime>
    <NodeName>kzh</NodeName>
    <DagType>DAILY</DagType>
    <ParamValues>bizdate=$bizdate tbods=$tbods</ParamValues>
    <DagId>33845</DagId>
    <BeginRunningTime>1590416703313</BeginRunningTime>
    <Bizdate>1590336000000</Bizdate>
    <FinishTime>1590416703313</FinishTime>
    <DqcDescription>[{"projectName":"ztjy_dim","tableName":"dim_user_agent_manage_area_a","partition":"ds\u003d${yyyy-mm-dd-1}"}]</DqcDescription>
    <Status>NOT_RUN</Status>
    <TaskType>NORMAL(0)</TaskType>
    <TaskRerunTime>0</TaskRerunTime>
    <InstanceId>1234</InstanceId>
    <DqcType>1</DqcType>
    <Priority>1</Priority>
    <CreateTime>1590416703313</CreateTime>
    <Connection>odps_first</Connection>
    <RelatedFlowId>123456</RelatedFlowId>
    <BeginWaitTimeTime>1590416703313</BeginWaitTimeTime>
    <CycTime>1590422400000</CycTime>
    <Repeatability>true</Repeatability>
    <NodeId>33115</NodeId>
    <BeginWaitResTime>1590416703313</BeginWaitResTime>
    <ErrorMessage>error message</ErrorMessage>
    <RepeatInterval>60000</RepeatInterval>
  </Instances>
  <TotalCount>66</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
</Data>
<ErrorCode>Invalid.Tenant.ProjectNotExists</ErrorCode>
<ErrorMessage>The project does not exist.</ErrorMessage>
<Success>true</Success>

```

JSON format


```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": {
    "Instances": {
      "BaselineId": 123123,
      "ModifyTime": 1590416703313,
      "NodeName": "kzh",
      "DagType": "DAILY",
      "ParamValues": "bizdate=$bizdate tbods=$tbods",
      "DagId": 33845,
      "BeginRunningTime": 1590416703313,
      "Bizdate": 1590336000000,
      "FinishTime": 1590416703313,
      "DqcDescription": "[{"projectName\\":\\"ztjy_dim\\",\\"tableName\\":\\"dim_user_agent_manage_area_a\\",\\"partition\\":\\"ds\\u003d${yyyy-mm-dd-1}\\"}]",
      "Status": "NOT_RUN",
      "TaskType": "NORMAL(0)",
      "TaskRerunTime": 0,
      "InstanceId": 1234,
      "DqcType": 1,
      "Priority": 1,
      "CreateTime": 1590416703313,
      "Connection": "odps_first",
      "RelatedFlowId": 123456,
      "BeginWaitTime": 1590416703313,
      "CycTime": 1590422400000,
      "Repeatability": true,
      "NodeId": 33115,
      "BeginWaitResTime": 1590416703313,
      "ErrorMessage": "error message",
      "RepeatInterval": 60000
    },
    "TotalCount": 66,
    "PageSize": 10,
    "PageNumber": 1
  },
  "ErrorCode": "Invalid.Tenant.ProjectNotExists",
  "ErrorMessage": "The project does not exist.",
  "Success": true
}
```

2.1.6.4.22. GetInstanceLog

Queries the logs of an instance.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	GetInstanceLog	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	String	log_content	The content of the logs.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetInstanceLog
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>log_content</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": "log_content",
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.23. StopInstance

Terminates an instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	StopInstance	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The result returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=StopInstance
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.24. RestartInstance

Restarts an instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RestartInstance	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The result returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	q32ety****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RestartInstance
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>q32ety****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "q32ety****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.25. SetSuccessInstance

Resets the state of a failed instance to successful.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SetSuccessInstance	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The result returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=SetSuccessInstance
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.26. SuspendInstance

Suspends an instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	SuspendInstance	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The result returned.
ErrorCode	String	Invalid.Tenant.ProjectNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0baf87f0159376****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=SuspendInstance
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0baf87f0159376****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ProjectNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0baf87f0159376****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ProjectNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.27. ResumeInstance

Resumes a suspended instance.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ResumeInstance	The operation that you want to perform.
InstanceId	Long	Yes	1234	The ID of the instance.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The result returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ResumeInstance
&InstanceId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.28. RunCycleDagNodes

Creates a workflow to generate retroactive data.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RunCycleDagNodes	The operation that you want to perform.
EndBizDate	String	Yes	2020-05-21 00:00:00	The data timestamp at which retroactive data generation ends. Specify the time in the yyyy-MM-dd 00:00:00 format.
IncludeNodeIds	String	Yes	74324,74325	The ID of the node for which you want to generate retroactive data. If you want to generate retroactive data for multiple nodes, separate the IDs of the nodes with commas (.). You can call the ListNodes operation to query the node IDs.
Name	String	Yes	xm_test	The name of the workflow.
Parallelism	Boolean	Yes	false	Specifies whether retroactive data can be generated for multiple nodes at the same time.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Parameter	Type	Required	Example	Description
RootNodeid	Long	Yes	1234	The ID of the node for which retroactive data is first generated. You can call the ListNodes operation to query the node ID.
StartBizDate	String	Yes	2020-05-20 00:00:00	The data timestamp at which retroactive data generation starts. Specify the time in the yyyy-MM-dd 00:00:00 format.
ExcludeNodeids	String	No	1234,123465	The IDs of the nodes for which no retroactive data needs to be generated. The system generates dry-run instances for all these nodes. After these dry-run instances are scheduled, the states of these instances are directly set to successful, but the scripts are not run.
BizBeginTime	String	No	00:00:00	The time when the node started to run. This parameter is required only for auto triggered nodes that are scheduled by the hour. Specify the time in the HH:mm:ss format. Valid values: 00:00:00 to 23:59:59.
BizEndTime	String	No	01:00:00	The time when the node stopped running. This parameter is required only for auto triggered nodes that are scheduled by the hour. Specify the time in the HH:mm:ss format. Valid values: 00:00:00 to 23:59:59.
NodeParams	String	No	null	The parameters that need to be configured for the node. Set this parameter to a JSON string. The key indicates the ID of the node, and the value indicates the actual values of the parameters.

Response parameters

Parameter	Type	Example	Description
Data	List	111	The IDs of the nodes in the workflow.

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	>E6F0DBDD-5AD****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RunCycleDagNodes
&EndBizDate=2020-05-21 00:00:00
&IncludeNodeIds=74324,74325
&Name=xm_test
&Parallelism=false
&ProjectEnv=PROD
&RegionId=cn-shanghai
&RootNodeId=1234
&StartBizDate=2020-05-20 00:00:00
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>111</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": 111,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.29. RunSmokeTest

Creates a workflow to perform smoke testing.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RunSmokeTest	The operation that you want to perform.
Bizdate	String	Yes	2020-05-26 00:00:00	The data timestamp.
Name	String	Yes	xm_create_test	The name of the workflow.
NodeId	Long	Yes	1234	The ID of the node. You can call the ListNodes operation to query the ID.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. The value PROD indicates the production environment. The value DEV indicates the development environment.
RegionId	String	Yes	cn-shanghai	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Parameter	Type	Required	Example	Description
NodeParams	String	No	bizdate=\$bizdate param1=demo1 param2=demo2	The parameters related to the node.

Response parameters

Parameter	Type	Example	Description
Data	Long	3333333	The ID of the workflow.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	E6F0DBDD-5AD****	The ID of the request. You can use the ID to obtain logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RunSmokeTest
&Bizdate=2020-05-26 00:00:00
&Name=xm_create_test
&NodeId=1234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>E6F0DBDD-5AD****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>3333333</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```


JSON format

```
{
  "RequestId": "E6F0DBDD-5AD****",
  "HttpStatusCode": 200,
  "Data": 3333333,
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.30. ListSuccessInstanceAmount

Queries auto triggered node instances that are generated on the current day and collects statistics about the trends in the number of instances that are run every hour on the hour.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListSuccessInstanceAmount	The operation that you want to perform.
ProjectId	Long	Yes	9527	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
InstanceStatusTrend	Struct	N/A	The trends in the number of instances that are run every hour on the hour on the current day.
AvgTrend	Array of AvgTrend	N/A	The historical average trends.

Parameter	Type	Example	Description
Count	Integer	10	The number of instances that were run.
TimePoint	String	01:00	The time when the data was collected. The value indicates the beginning of an hour in the range of 00:00 to 23:00, such as 00:00, 01:00, or 02:00.
TodayTrend	Array of TodayTrend	N/A	The trends on the current day.
Count	Integer	10	The number of instances that were run.
TimePoint	String	01:00	The time when the data was collected. The value indicates the beginning of an hour in the range of 00:00 to 23:00, such as 00:00, 01:00, or 02:00.
YesterdayTrend	Array of YesterdayTrend	N/A	The trends on the day before the current day.
Count	Integer	10	The number of instances that were run.
TimePoint	String	01:00	The time when the data was collected. The value indicates the beginning of an hour in the range of 00:00 to 23:00, such as 00:00, 01:00, or 02:00.
RequestId	String	952795279527ab****	The ID of the request. You can use the ID to troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListSuccessInstanceAmount
&ProjectId=9527
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>952795279527ab****</RequestId>
<InstanceStatusTrend>
  <YesterdayTrend>
    <TimePoint>01:00</TimePoint>
    <Count>10</Count>
  </YesterdayTrend>
  <AvgTrend>
    <TimePoint>01:00</TimePoint>
    <Count>10</Count>
  </AvgTrend>
  <TodayTrend>
    <TimePoint>01:00</TimePoint>
    <Count>10</Count>
  </TodayTrend>
</InstanceStatusTrend>
```

JSON format

```
{
  "RequestId": "952795279527ab****",
  "InstanceStatusTrend": {
    "YesterdayTrend": {
      "TimePoint": "01:00",
      "Count": 10
    },
    "AvgTrend": {
      "TimePoint": "01:00",
      "Count": 10
    },
    "TodayTrend": {
      "TimePoint": "01:00",
      "Count": 10
    }
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Serve r	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.31. ListInstanceAmount

Queries the trends in the number of auto triggered node instances within a specified period of time.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListInstanceAmount	The operation that you want to perform.
BeginDate	String	Yes	2020-04-02T00:00:00+0800	The beginning of the time range to query, accurate to the day. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
EndDate	String	Yes	2020-09-10T00:00:00+0800	The end of the time range to query, accurate to the day. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
ProjectId	Long	Yes	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
InstanceCounts	Array of InstanceCounts	N/A	The trends in the number of auto triggered node instances within the specified period of time.
Count	Integer	9527	The number of auto triggered node instances.
Date	Long	1600963200000	The data timestamp at which the number of auto triggered node instances was obtained. This value is a UNIX timestamp.

Parameter	Type	Example	Description
RequestId	String	95279527adhj****	The ID of the request. You can use the ID to troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListInstanceAmount
&BeginDate=2020-04-02T00:00:00+0800
&EndDate=2020-09-10T00:00:00+0800
&ProjectId=10000
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>95279527adhj****</RequestId>
<InstanceCounts>
  <Count>9527</Count>
  <Date>1600963200000</Date>
</InstanceCounts>
```

JSON format

```
{
  "RequestId": "95279527adhj****",
  "InstanceCounts": {
    "Count": 9527,
    "Date": 1600963200000
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Servicer	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Wkbench.NoProject	You have not joined any project.	The error message returned because you have not joined a workspace.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.32. TopTenElapsedTimeInstance

Queries the rankings of the durations of instances.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	TopTenElapsedTimeInstance	The operation that you want to perform.
BusinessDate	String	Yes	2020-09-21T00:00:00+0800	The data timestamp, accurate to the day. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
ProjectId	Long	Yes	100000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
InstanceConsumeTimeRank	Struct	N/A	The rankings of the durations of instances.
ConsumeTimeRank	Array of ConsumeTimeRank	N/A	The ranking data of the durations of instances.
BusinessDate	Long	1600963200000	The data timestamp of the instance.
Consumed	Long	1000	The duration of the instance. Unit: seconds.
InstanceId	Long	95279527	The ID of the instance.

Parameter	Type	Example	Description
NodeId	Long	9527	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	952795279527	The ID of the Apsara Stack tenant account used by the node owner.
ProgramType	Integer	10	The type of the node. Valid values: 6: a Shell node. 10: an ODPS SQL node. 11: an ODPS MR node. 23: a Data Integration node. 24: an ODPS Script node. 99: a zero load node. 221: a PyODPS 2 node. 225: an ODPS Spark node. 227: an EMR Hive node. 228: an EMR Spark node. 229: an EMR Spark SQL node. 230: an EMR MR node. 239: an OSS object inspection node. 257: an EMR Shell node. 258: an EMR Spark Shell node. 259: an EMR Presto node. 260: an EMR Impala node. 900: a real-time sync node. 1089: a cross-tenant collaboration node. 1091: a Hologres development node. 1093: a Hologres SQL node. 1100: an assignment node. 1221: a PyODPS 3 node.
UpdateTime	Long	1600963200000	The timestamp at which the rankings of the durations of instances were updated.
RequestId	String	6347364dadsadf*** *	The ID of the request. You can use the ID to troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=TopTenElapsedTimeInstance
&BusinessDate=2020-09-21T00:00:00+0800
&ProjectId=100000
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>6347364dadsfadf****</RequestId>
<InstanceConsumeTimeRank>
  <UpdateTime>1600963200000</UpdateTime>
  <ConsumeTimeRank>
    <Owner>952795279527</Owner>
    <BusinessDate>1600963200000</BusinessDate>
    <Instanceld>95279527</Instanceld>
    <NodeName>Node name</NodeName>
    <NodeId>9527</NodeId>
    <Consumed>1000</Consumed>
    <ProgramType>10</ProgramType>
  </ConsumeTimeRank>
</InstanceConsumeTimeRank>

```

JSON format

```

{
  "RequestId": "6347364dadsfadf****",
  "InstanceConsumeTimeRank": {
    "UpdateTime": 1600963200000,
    "ConsumeTimeRank": {
      "Owner": 952795279527,
      "BusinessDate": 1600963200000,
      "Instanceld": 95279527,
      "NodeName": "Node name",
      "NodeId": 9527,
      "Consumed": 1000,
      "ProgramType": 10
    }
  }
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Served	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.33. TopTenErrorTimesInstance

Queries the rankings of nodes on which errors occurred within the last month.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	TopTenErrorTimeInstance	The operation that you want to perform.
ProjectId	Long	Yes	9527	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
InstanceErrorRank	Struct	N/A	The rankings of nodes on which errors occurred.
ErrorRank	Array of ErrorRank	N/A	The ranking data of nodes on which errors occurred within the last month.
Count	Integer	5	The number of errors that occurred on the node.
NodeId	Long	9527	The ID of the node.
NodeName	String	Node name	The name of the node.
Owner	String	952795279527	The ID of the Apsara Stack tenant account used by the node owner.

Parameter	Type	Example	Description
ProgramType	Integer	10	The type of the node. Valid values: 6: a Shell node. 10: an ODPS SQL node. 11: an ODPS MR node. 23: a Data Integration node. 24: an ODPS Script node. 99: a zero load node. 221: a PyODPS 2 node. 225: an ODPS Spark node. 227: an EMR Hive node. 228: an EMR Spark node. 229: an EMR Spark SQL node. 230: an EMR MR node. 239: an OSS object inspection node. 257: an EMR Shell node. 258: an EMR Spark Shell node. 259: an EMR Presto node. 260: an EMR Impala node. 900: a real-time sync node. 1089: a cross-tenant collaboration node. 1091: a Hologres development node. 1093: a Hologres SQL node. 1100: an assignment node. 1221: a PyODPS 3 node.
ProjectId	Long	9527	The ID of the DataWorks workspace.
UpdateTime	Long	1600963200000	The timestamp at which the rankings were updated.
RequestId	String	952795279527****	The ID of the request. You can use the ID to troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=TopTenErrorTimesInstance
&ProjectId=9527
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```

<RequestId>952795279527****</RequestId>
<InstanceErrorRank>
  <UpdateTime>1600963200000</UpdateTime>
  <ErrorRank>
    <Owner>952795279527</Owner>
    <NodeName>Node name</NodeName>
    <ProjectId>9527</ProjectId>
    <NodeId>9527</NodeId>
    <ProgramType>10</ProgramType>
    <Count>5</Count>
  </ErrorRank>
</InstanceErrorRank>

```

JSON format

```

{
  "RequestId": "952795279527****",
  "InstanceErrorRank": {
    "UpdateTime": 1600963200000,
    "ErrorRank": {
      "Owner": 952795279527,
      "NodeName": "Node name",
      "ProjectId": 9527,
      "NodeId": 9527,
      "ProgramType": 10,
      "Count": 5
    }
  }
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	Forbidden.Not.Servicer	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.

HTTP status code	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Wkbench.ProjectNoPermission	You are not authorized to access the workspace.	The error message returned because you are not authorized to access the workspace.
500	Invalid.Wkbench.Parameter	The specified parameters are invalid.	The error message returned because the values of one or more parameters are invalid.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.34. GetFileTypeStatistic

Queries the distribution of node types.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetFileTypeStatistic	The operation that you want to perform.

Parameter	Type	Required	Example	Description
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: DEV and PROD. A value of DEV indicates the development environment. A value of PROD indicates the production environment.
ProjectId	Long	Yes	123465	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
ProgramTypeAndCounts	Array of ProgramTypeAndCount	N/A	The list of node types and quantity.
Count	Integer	1	The number of nodes.

Parameter	Type	Example	Description
ProgramType	String	ODPS_SQL	The type of node. Valid values: 6: a Shell node. 10: an ODPS SQL node. 11: an ODPS MR node. 23: a Data Integration node. 24: an ODPS Script node. 99: a zero load node. 221: a PyODPS 2 node. 225: an ODPS Spark node. 227: an EMR Hive node. 228: an EMR Spark node. 229: an EMR Spark SQL node. 230: an EMR MR node. 239: an OSS object inspection node. 257: an EMR Shell node. 258: an EMR Spark Shell node. 259: an EMR Presto node. 260: an EMR Impala node. 900: a real-time sync node. 1089: a cross-tenant collaboration node. 1091: a Hologres development node. 1093: a Hologres SQL node. 1100: an assignment node. 1221: a PyODPS 3 node.
RequestId	String	SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF	The ID of the request. You can use the ID to query logs and troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetFileTypeStatistic
&ProjectEnv=PROD
&ProjectId=123465
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF</RequestId>
<ProgramTypeAndCounts>
  <Count>1</Count>
  <ProgramType>ODPS_SQL</ProgramType>
</ProgramTypeAndCounts>
```

JSON format

```
{
  "RequestId": "SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF",
  "ProgramTypeAndCounts": {
    "Count": 1,
    "ProgramType": "ODPS_SQL"
  }
}
```

2.1.6.4.35. GetInstanceStatusStatistic

Queries the number of instances in each state.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetInstanceStatusStatistic	The operation that you want to perform.
BizDate	String	Yes	2020-01-01	The data timestamp at which the numbers of instances in different states are obtained.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
ProjectId	Long	Yes	1234	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to query logs and troubleshoot issues.
StatusCount	Struct	N/A	The numbers of instances in different states.
FailureCount	Integer	1	The number of instances that failed to run.
NotRunCount	Integer	1	The number of instances that were not run.
RunningCount	Integer	1	The number of running instances.
SuccessCount	Integer	1	The number of instances that were run.
TotalCount	Integer	1	The total number of instances returned.
WaitResCount	Integer	1	The number of instances that were waiting for resources.
WaitTimeCount	Integer	1	The number of instances that were waiting to run.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetInstanceStatusStatistic
&BizDate=2020-01-01
&ProjectEnv=PROD
&ProjectId=1234
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```

<StatusCount>
  <TotalCount>1</TotalCount>
  <FailureCount>1</FailureCount>
  <RunningCount>1</RunningCount>
  <WaitResCount>1</WaitResCount>
  <SuccessCount>1</SuccessCount>
  <NotRunCount>1</NotRunCount>
  <WaitTimeCount>1</WaitTimeCount>
</StatusCount>
<RequestId>0000-ABCD-EFG****</RequestId>

```

JSON format

```

{
  "StatusCount": {
    "TotalCount": 1,
    "FailureCount": 1,
    "RunningCount": 1,
    "WaitResCount": 1,
    "SuccessCount": 1,
    "NotRunCount": 1,
    "WaitTimeCount": 1
  },
  "RequestId": "0000-ABCD-EFG****"
}

```

2.1.6.4.36. ListFileType

Queries the information about one or more node types, such as the code and name of a node type.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListFileType	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return.
PageSize	Integer	Yes	50	The number of entries to return on each page. Maximum value: 100.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-zhangjiakou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
ProjectId	Long	No	10000	The ID of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace ID. You must specify either this parameter or the ProjectIdentifier parameter to determine the DataWorks workspace to which the operation is applied.
ProjectIdentifier	String	No	dw_project	The name of the DataWorks workspace. You can log on to the DataWorks console and go to the Workspace Management page to obtain the workspace name. You must specify either this parameter or the ProjectId parameter to determine the DataWorks workspace to which the operation is applied.
Keyword	String	No	ODPS SQL	The name of the node type. You can log on to the DataWorks console, go to the DataStudio page, and then view the name of a specific node type in the left-side navigation pane of the page. Take note of the following items when you use this parameter: - You can view the English or Chinese name of a specific node type, but the language specified by this parameter to present the name must be the same as the language specified by the Locale parameter. - Fuzzy search is supported. - If this parameter is not specified, the names of all node types are returned.

Parameter	Type	Required	Example	Description
Locale	String	No	zh-CN	The language that you use for the query. Valid values: zh-CN and en-US.

Response parameters

Parameter	Type	Example	Description
NodeTypeInfoList	Struct	N/A	The information about node types.
NodeTypeInfo	Array of NodeTypeInfo	N/A	The information about each node type.
NodeType	Integer	10	The code of the node type. The codes and names of node types have the following mappings: 6: a Shell node. 10: an ODPS SQL node. 11: an ODPS MR node. 23: a Data Integration node. 24: an ODPS Script node. 99: a zero load node. 221: a PyODPS 2 node. 225: an ODPS Spark node. 227: an EMR Hive node. 228: an EMR Spark node. 229: an EMR Spark SQL node. 230: an EMR MR node. 239: an OSS object inspection node. 257: an EMR Shell node. 258: an EMR Spark Shell node. 259: an EMR Presto node. 260: an EMR Impala node. 900: a real-time sync node. 1089: a cross-tenant collaboration node. 1091: a Hologres development node. 1093: a Hologres SQL node. 1100: an assignment node. 1221: a PyODPS 3 node.

Parameter	Type	Example	Description
NodeTypeNames	String	ODPS SQL	The name of the node type. The codes and names of node types have the following mappings: 6: a Shell node. 10: an ODPS SQL node. 11: an ODPS MR node. 23: a Data Integration node. 24: an ODPS Script node. 99: a zero load node. 221: a PyODPS 2 node. 225: an ODPS Spark node. 227: an EMR Hive node. 228: an EMR Spark node. 229: an EMR Spark SQL node. 230: an EMR MR node. 239: an OSS object inspection node. 257: an EMR Shell node. 258: an EMR Spark Shell node. 259: an EMR Presto node. 260: an EMR Impala node. 900: a real-time sync node. 1089: a cross-tenant collaboration node. 1091: a Hologres development node. 1093: a Hologres SQL node. 1100: an assignment node. 1221: a PyODPS 3 node.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	50	The number of entries returned per page.
TotalCount	Integer	127	The total number of entries returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request. You can use the ID to troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListFileType
&PageNumber=1
&PageSize=50
&RegionId=cn-zhangjiakou
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<NodeTypeInfoList>
  <TotalCount>127</TotalCount>
  <PageSize>50</PageSize>
  <PageNumber>1</PageNumber>
  <NodeTypeInfo>
    <NodeTypeName>ODPS SQL</NodeTypeName>
    <NodeType>10</NodeType>
  </NodeTypeInfo>
</NodeTypeInfoList>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "NodeTypeInfoList": {
    "TotalCount": 127,
    "PageSize": 50,
    "PageNumber": 1,
    "NodeTypeInfo": {
      "NodeTypeName": "ODPS SQL",
      "NodeType": 10
    }
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.4.37. RunManualDagNodes

Runs nodes in a manually triggered workflow. The manually triggered workflow must be committed and deployed. Then, you can find the manually triggered workflow on the Operation Center page and call this operation.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RunManualDagNodes	The operation that you want to perform. Set the value to RunManualDagNodes.
BizDate	String	Yes	2020-11-11 00:00:00	The data timestamp. The value must be one or more days before the current date. For example, if the current date is November 11, 2020, set the value to 2020-11-10 00:00:00 or earlier. Specify this parameter in the YYYY-MM-DD 00:00:00 format.
FlowName	String	Yes	test_workflow	The name of the manually triggered workflow.

Parameter	Type	Required	Example	Description
NodeParameters	String	Yes	<pre>{"20000123121": "key1=val2 key2=val2", "20000123124": "kkkk=vvvv aaaa=bbbb"}</pre>	The parameters transmitted between nodes in the manually triggered workflow. The parameters are in the following JSON format: { "<ID of a node in the manually triggered workflow>": "Scheduling parameter settings of the node, which are in the same format as the Parameters parameter on the Properties tab of the DataStudio page", "<ID of a node in the manually triggered workflow>": "Scheduling parameter settings of the node, which are in the same format as the Parameters parameter on the Properties tab of the DataStudio page" }.
ProjectEnv	String	Yes	PROD or DEV	The environment type. Valid values: PROD and DEV. The value PROD indicates the production environment. The value DEV indicates the development environment.
ProjectName	String	Yes	test_workspace	The name of the workspace to which the manually triggered workflow belongs.
RegionId	String	Yes	cn-shanghai	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
DagParameters	String	No	<pre>{"kaaaa": "vaaaaa", "kbbbb": "vbbbbbb"}</pre>	The parameters of the manually triggered workflow, which are synchronized to all the instances in the directed acyclic graph (DAG) of the workflow. If a workflow parameter specified in DagParameters is referenced as a scheduling parameter of a node, the value of the scheduling parameter is replaced with the value of the workflow parameter.

Response parameters

Parameter	Type	Example	Description
DagId	Long	700000123123141	The ID of the DAG for the manually triggered workflow. You can call an operation with this parameter as a request parameter to query the details and statuses of the nodes in this manually triggered workflow.
RequestId	String	AASFDFSDFG-DFSDF-DFSDFD-SDFSDF	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RunManualDagNodes
&BizDate=2020-11-11 00:00:00
&FlowName=test_workflow
&NodeParameters={"20000123121": "key1=val2 key2=val2", "20000123124": "kkkk=vvvvvv aaaa=bbbb"}
&ProjectEnv=PROD or DEV
&ProjectName=test_workspace
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>AASFDFSDFG-DFSDF-DFSDFD-SDFSDF</RequestId>
<DagId>700000123123141</DagId>
```

JSON format

```
{
  "RequestId": "AASFDFSDFG-DFSDF-DFSDFD-SDFSDF",
  "DagId": 700000123123141
}
```

2.1.6.4.38. GetManualDagInstances

Queries the information about instances in a manually triggered workflow.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	GetManualDagInstances	The operation that you want to perform.
DagId	String	Yes	7000001231241	The ID of the directed acyclic graph (DAG) for the manually triggered workflow. You can call the CreateManualDag operation to obtain the ID.
ProjectEnv	String	Yes	RPOD	The environment type. Valid values: PROD and DEV. The value PROD indicates the production environment. The value DEV indicates the development environment.
ProjectName	String	Yes	test_workspace	The name of the workspace to which the manually triggered workflow belongs. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the workspace name.
RegionId	String	Yes	cn-shanghai	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
RequestId	String	SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF	The ID of the request.
Instances	Array of Instances		The instances in the manually triggered workflow.
BeginRunningTime	Long	1605178414676	The time when the instance started to run.

Parameter	Type	Example	Description
BeginWaitResTime	Long	1605178414676	The time when the instance started to wait for resources.
BeginWaitTime	Long	1605178414676	The time when the instance started to wait to be run.
BizDate	Long	1605178414676	The data timestamp. In most cases, the value is one day before the time when the instance is run.
CreateTime	Long	1605178414676	The time when the instance was created.
CycTime	Long	1605178414676	The time when the instance was scheduled to run.
DagId	Long	350850491	The ID of the DAG for the manually triggered workflow.
DagType	String	5	The type of the manually triggered workflow.
FinishTime	Long	1605178414676	The time when the running of the instance was complete.
InstancelId	Long	11726873619	The ID of the instance in the manually triggered workflow.
ModifyTime	Long	1605178414676	The time when the instance was last modified.
NodeId	Long	37851	The ID of the node in the manually triggered workflow.
NodeName	String	test2	The name of the node.
ParamValues	String	xxx	The parameter settings of the instance.

Parameter	Type	Example	Description
Status	String	WAIT_TIME	The status of the instance. Valid values: • NOT_RUN: The instance is not run. • WAIT_TIME: The instance is waiting for its scheduled time to arrive. • WAIT_RESOURCE: The instance is waiting for resources. • RUNNING: The instance is running. • CHECKING: Data quality is being checked for the instance. • CHECKING_CONDITION: Branch conditions are being checked for the instance. • FAILURE: The instance fails to run. • SUCCESS: The running of the instance is successful.

Parameter	Type	Example	Description
TaskType	String	NORMAL(0)	The scheduling type of the instance. Valid values: • NORMAL(0): The instance is an auto triggered node instance. It is run on a regular basis. • MANUAL(1): The instance is a manually triggered node instance. It is not run on a regular basis. • PAUSE(2): The instance is paused. The scheduling system still runs the instance but sets it to FAILURE when the scheduling system starts to run the instance. • SKIP(3): The instance is a dry-run instance. The scheduling system still runs the instance but sets it to SUCCESS when the scheduling system starts to run the instance. • SKIP_UNCHOOSE(4): The instance is an unselected instance in a temporary workflow. This type of instance exists only in temporary workflows. The scheduling system sets the instance to SUCCESS when the scheduling system starts to run the instance. • SKIP_CYCLE(5): The node is a weekly or monthly instance whose running cycle does not arrive. The scheduling system still runs the instance but sets it to SUCCESS when the scheduling system starts to run the instance. • CONDITION_UNCHOOSE(6): The instance is not selected by its ancestor branch node instance and is set to a dry-run instance. • REALTIME_DEPRECATED(7): The instance is generated in real time but is deprecated. The scheduling system directly sets it to SUCCESS.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetManualDagInstances
&DagId=7000001231241
&ProjectEnv=RPOD
&ProjectName=test_workspace
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```

<success>true</success>
<statusCode>200</statusCode>
<data>
  <nodeId>37851</nodeId>
  <instanceId>11726873619</instanceId>
  <dagId>350850491</dagId>
  <dagType>BUSINESS_PROCESS_DAG</dagType>
  <status>WAIT_TIME</status>
  <TaskType>NORMAL(0)</TaskType>
  <bizdate>1589846400000</bizdate>
  <paramValues>bizdate=20200519</paramValues>
  <cycTime>1589932800000</cycTime>
  <beginWaitTime>1605178414676</beginWaitTime>
  <createTime>1605178401323</createTime>
  <modifyTime>1605178414676</modifyTime>
  <nodeName>test2</nodeName>
</data>
<data>
  <nodeId>37178</nodeId>
  <instanceId>11726873618</instanceId>
  <dagId>350850491</dagId>
  <dagType>BUSINESS_PROCESS_DAG</dagType>
  <status>WAIT_TIME</status>
  <bizdate>1589846400000</bizdate>
  <paramValues>bizdate=20200519</paramValues>
  <cycTime>1589932800000</cycTime>
  <beginWaitTime>1605178414531</beginWaitTime>
  <createTime>1605178401311</createTime>
  <modifyTime>1605178414531</modifyTime>
  <nodeName>ide_file_full</nodeName>
</data>
<data>
  <nodeId>37829</nodeId>
  <instanceId>11726873617</instanceId>
  <dagId>350850491</dagId>
  <dagType>BUSINESS_PROCESS_DAG</dagType>
  <status>WAIT_TIME</status>
  <bizdate>1589846400000</bizdate>
  <paramValues>bizdate=20200519</paramValues>
  <cycTime>1589932800000</cycTime>
  <beginWaitTime>1605178414616</beginWaitTime>
  <createTime>1605178401299</createTime>
  <modifyTime>1605178414616</modifyTime>
  <nodeName>test</nodeName>
</data>

```

JSON format


```
{
  "success": true,
  "statusCode": 200,
  "data": [
    {
      "nodeId": 37851,
      "instanceId": 11726873619,
      "dagId": 350850491,
      "dagType": "BUSINESS_PROCESS_DAG",
      "status": "WAIT_TIME",
      "TaskType": "NORMAL(0)",
      "bizdate": 1589846400000,
      "paramValues": "bizdate\u003d20200519",
      "cycTime": 1589932800000,
      "beginWaitTime": 1605178414676,
      "createTime": 1605178401323,
      "modifyTime": 1605178414676,
      "nodeName": "test2"
    },
    {
      "nodeId": 37178,
      "instanceId": 11726873618,
      "dagId": 350850491,
      "dagType": "BUSINESS_PROCESS_DAG",
      "status": "WAIT_TIME",
      "bizdate": 1589846400000,
      "paramValues": "bizdate\u003d20200519",
      "cycTime": 1589932800000,
      "beginWaitTime": 1605178414531,
      "createTime": 1605178401311,
      "modifyTime": 1605178414531,
      "nodeName": "ide_file_full"
    },
    {
      "nodeId": 37829,
      "instanceId": 11726873617,
      "dagId": 350850491,
      "dagType": "BUSINESS_PROCESS_DAG",
      "status": "WAIT_TIME",
      "bizdate": 1589846400000,
      "paramValues": "bizdate\u003d20200519",
      "cycTime": 1589932800000,
      "beginWaitTime": 1605178414616,
      "createTime": 1605178401299,
      "modifyTime": 1605178414616,
      "nodeName": "test"
    }
  ]
}
```

2.1.6.4.39. ListManualDagInstances

Queries instances in a manually triggered workflow.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListManualDagInstances	The operation that you want to perform.
DagId	String	Yes	7000001231241	The ID of the directed acyclic graph (DAG) for the manually triggered workflow. You can call the CreateManualDag operation to obtain the workflow ID.
ProjectEnv	String	Yes	RPOD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
ProjectName	String	Yes	test_workspace	The name of the workspace to which the manually triggered workflow belongs. You can log on to the DataWorks console and go to the Workspace Settings panel to obtain the workspace name.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Instances	Array of Instances	N/A	The list of instances in the manually triggered workflow.
BeginRunningTime	Long	1605178414676	The time when the instance started to run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.

Parameter	Type	Example	Description
BeginWaitResTime	Long	1605178414676	The time when the instance started to wait for resources. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
BeginWaitTime	Long	1605178414676	The time when the instance started to wait to be scheduled. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
BizDate	Long	1605178414676	The data timestamp of the instance. In most cases, the value indicates one day before the time when the instance is run.
CreateTime	Long	1605178414676	The time when the instance was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CycTime	Long	1605178414676	The time when the instance was scheduled to run. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
DagId	Long	350850491	The ID of the DAG for the manually triggered workflow.
DagType	String	5	The type of the manually triggered workflow.
FinishTime	Long	1605178414676	The time when the instance stopped running. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
InstancelId	Long	11726873619	The ID of the instance in the manually triggered workflow.

Parameter	Type	Example	Description
ModifyTime	Long	1605178414676	The time when the instance was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
NodeId	Long	37851	The ID of the node in the manually triggered workflow.
NodeName	String	test2	The name of the node.
ParamValues	String	xxx=yyy	The parameters related to the instance.
Status	String	SUCCESS	The status of the node. Valid values: • NOT_RUN: The node is not run. • WAIT_TIME: The node is waiting for its scheduled time to arrive. • WAIT_RESOURCE: The node is waiting for resources. • RUNNING: The node is running. • CHECKING: Data quality is being checked for the node. • CHECKING_CONDITION: Branch conditions are being checked for the node. • FAILURE: The node fails to run. • SUCCESS: The node is run.

Parameter	Type	Example	Description
TaskType	String	MANUAL	The scheduling type of the node. Valid values: • NORMAL(0): The node is an auto triggered node. It is run on a regular basis. • MANUAL(1): The node is a manually triggered node. It is not run on a regular basis. • PAUSE(2): The node is a frozen node. The scheduling system still runs the node but sets it to FAILURE when the scheduling system starts to run the node. • SKIP(3): The node is a dry-run node. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node. • SKIP_UNCHOOSE(4): The node is an unselected node in a temporary workflow. This type of node exists only in temporary workflows. The scheduling system sets the node to SUCCESS when the scheduling system starts to run the node. • SKIP_CYCLE(5): The node is a weekly or monthly node whose scheduled time does not arrive. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node. • CONDITION_UNCHOOSE(6): The node is not selected by its ancestor branch node and is run as a dry-run node. • REALTIME_DEPRECATED(7): The node has instances generated in real time but deprecated. The scheduling system directly sets it to SUCCESS.
RequestId	String	SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF	The ID of the request. You can use the ID to query logs and troubleshoot issues.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListManualDagInstances
&DagId=7000001231241
&ProjectEnv=RPOD
&ProjectName=test_workspace
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<Instances>
  <Status>SUCCESS</Status>
  <ModifyTime>1605178414676</ModifyTime>
  <NodeName>test2</NodeName>
  <DagType>5</DagType>
  <Instanceld>11726873619</Instanceld>
  <ParamValues>xxx=yyy</ParamValues>
  <CreateTime>1605178414676</CreateTime>
  <BeginWaitTimeTime>1605178414676</BeginWaitTimeTime>
  <CycTime>1605178414676</CycTime>
  <DagId>350850491</DagId>
  <BeginRunningTime>1605178414676</BeginRunningTime>
  <FinishTime>1605178414676</FinishTime>
  <TaskType>MANUAL</TaskType>
  <NodeId>37851</NodeId>
  <BeginWaitResTime>1605178414676</BeginWaitResTime>
  <BizDate>1605178414676</BizDate>
</Instances>
<RequestId>SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF</RequestId>
```

JSON format

```
{
  "Instances": {
    "Status": "SUCCESS",
    "ModifyTime": 1605178414676,
    "NodeName": "test2",
    "DagType": 5,
    "Instanceld": 11726873619,
    "ParamValues": "xxx=yyy",
    "CreateTime": 1605178414676,
    "BeginWaitTimeTime": 1605178414676,
    "CycTime": 1605178414676,
    "DagId": 350850491,
    "BeginRunningTime": 1605178414676,
    "FinishTime": 1605178414676,
    "TaskType": "MANUAL",
    "NodeId": 37851,
    "BeginWaitResTime": 1605178414676,
    "BizDate": 1605178414676
  },
  "RequestId": "SDFSDFSDF-SDFSDF-SDFSDF-SDFSDF"
}
```

2.1.6.4.40. GetDag

Queries the details of a directed acyclic graph (DAG) based on the DAG ID. You can call the GetDag operation to query the details of the DAG for a manually triggered workflow, a manually triggered node, or a retroactive instance. You cannot query the details of the DAG for an auto triggered workflow or node.

Supported DAG types:

MANUAL: the DAG for a manually triggered workflow

SMOKE_TEST: the DAG for a smoke testing workflow

SUPPLY_DATA: the DAG for a retroactive instance

BUSINESS_PROCESS_DAG: the DAG for a one-time workflow

Supported DAG states:

CREATED: The DAG is created.

RUNNING: The object is running.

FAILURE: The object fails to run.

SUCCESS: The object is run.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDag	The operation that you want to perform.
DagId	Long	Yes	123141452344525	The ID of the DAG. You can set this parameter to the value of the DagId parameter returned by the CreateDagComplement, CreateTest, or CreateManualDag operation.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
RegionId	String	Yes	cn-shanghai	The ID of the region where the workspace resides.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the DAG.

Parameter	Type	Example	Description
Bizdate	Long	1605052800000	The data timestamp at which the data was queried.
CreateTime	Long	1605052800000	The time when the DAG was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
CreateUser	String	1736629400048545	The user who created the DAG.
DagId	Long	351249682	The ID of the DAG.
FinishTime	Long	1605052800000	The end of the time range in which the data was queried. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Gmtdate	Long	1605052800000	The timestamp on the current day.
ModifyTime	Long	1605052800000	The time when the DAG was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Name	String	test_dag	The name of the DAG.
ProjectId	Long	112345	The ID of the workspace.
StartTime	Long	1605052800000	The beginning of the time range in which the data was queried. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
Status	String	FAILURE	The status of the DAG. Valid values: CREATED, RUNNING, FAILURE, and SUCCESS.
Type	String	MANUAL_FLOW	The type of the DAG. Valid values: MANUAL, SMOKE_TEST, SUPPLY_DATA, and BUSINESS_PROCESS_DAG.

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	7782a299-b291-4fee-8424-cf8058efa8e8	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDag
&DagId=123141452344525
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<success>true</success>
<statusCode>200</statusCode>
<data>
  <projectId>33679</projectId>
  <dagId>351249682</dagId>
  <name>aadasdfasdfasdf</name>
  <type>MANUAL_FLOW</type>
  <status>FAILURE</status>
  <bizdate>1605052800000</bizdate>
  <gmtdate>1605139200000</gmtdate>
  <startTime>1605211837166</startTime>
  <finishTime>1605211837282</finishTime>
  <createTime>1605211377024</createTime>
  <createUser>1736629400048545</createUser>
  <modifyTime>1605211837282</modifyTime>
</data>
<requestId>7782a299-b291-4fee-8424-cf8058efa8e8</requestId>
```

JSON format

```
{
  "success": true,
  "statusCode": 200,
  "data": {
    "projectId": 33679,
    "dagId": 351249682,
    "name": "aadasdfasdfsdf",
    "type": "MANUAL_FLOW",
    "status": "FAILURE",
    "bizdate": 1605052800000,
    "gmtdate": 1605139200000,
    "startTime": 1605211837166,
    "finishTime": 1605211837282,
    "createTime": 1605211377024,
    "createUser": "1736629400048545",
    "modifyTime": 1605211837282
  },
  "requestId": "7782a299-b291-4fee-8424-cf8058efa8e8"
}
```

2.1.6.4.41. ListNodesByOutput

Queries nodes based on the output of the nodes.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListNodesByOutput	The operation that you want to perform.
Outputs	String	Yes	aaaaa.1231412_output,bbbb.12313123_out	The output name of the node. You can specify multiple output names. Separate them with commas (,).
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Array of OutputNodePair	N/A	The list of returned nodes.
NodeList	Array of Node	N/A	The information about nodes.
BaselineId	Long	1235667	The ID of the baseline.
Connection	String	odps_first	The name of the connection.
CronExpress	String	00 00 00 * * ?	The CRON expression.
Description	String	test	The description of the node.
DqcDescription	String	[{"projectName": "ztjy_dim", "tableName": "dim_user_agent_manage_area_a", "partition": "ds\u003d\$[yyyy-mm-dd-1]"}]	The table and partition filter expression in Data Quality that are associated with the node.
DqcType	Integer	1	Indicates whether the node is associated with Data Quality. Valid values: 0 and 1. A value of 0 indicates that the node is associated with Data Quality. A value of 1 indicates that the node is not associated with Data Quality.

Parameter	Type	Example	Description
FileType	String	ODPS_SQL	The type of the node. Valid values: 6: a Shell node. 10: an ODPS SQL node. 11: an ODPS MR node. 23: a Data Integration node. 24: an ODPS Script node. 99: a zero load node. 221: a PyODPS 2 node. 225: an ODPS Spark node. 227: an EMR Hive node. 228: an EMR Spark node. 229: an EMR Spark SQL node. 230: an EMR MR node. 239: an OSS object inspection node. 257: an EMR Shell node. 258: an EMR Spark Shell node. 259: an EMR Presto node. 260: an EMR Impala node. 900: a real-time sync node. 1089: a cross-tenant collaboration node. 1091: a Hologres development node. 1093: a Hologres SQL node. 1100: an assignment node. 1221: a PyODPS 3 node.
NodeId	Long	125677	The ID of the node.
NodeName	String	liux_test_n****	The name of the node.
OwnerId	String	19337906836551	The ID of the node owner.
ParamValues	String	a=b	The scheduling parameters configured for the node.
Priority	Integer	1	The priority of the node. Valid values: 1, 3, 5, 7, and 8. A greater value indicates a higher priority. Default value: 1.
ProgramType	String	ODPS_SQL	The type of the node. This parameter is deprecated. For more information about node types, see the valid values of the FileType parameter.
ProjectId	Long	33671	The ID of the workspace.
RelatedFlowId	Long	1235655464	The ID of the workflow to which the node belongs.

Parameter	Type	Example	Description
RepeatInterval	Integer	60	The interval at which the node is rerun after the node fails to run.
Repeatability	Boolean	true	Indicates whether the node can be rerun.
ResGroupName	String	Default resource group	The name of the resource group.
SchedulerType	String	NORMAL	The scheduling type of the node. Valid values: NORMAL: The node is an auto triggered node. It is run on a regular basis. MANUAL: The node is a manually triggered node. It is not run on a regular basis. PAUSE: The node is a frozen node. The scheduling system still runs the node but sets it to FAILURE when the scheduling system starts to run the node. SKIP: The node is a dry-run node. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node.
Output	String	test_0709_1.630003556_out	The output name of the current node.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	SDFSDFSDF-asdfDFSDF-SDFSDF-SDFSDF	The ID of the request. You can use the ID to query logs and troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListNodesByOutput
&Outputs=aaaaa.1231412_out,bbbb.12313123_out
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>SDFSDFSDF-asdfDFSDF-SDFSDF-SDFSDF</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Output>test_0709_1.630003556_out</Output>
  <NodeList>
    <BaselineId>1235667</BaselineId>
    <ResGroupName>Default resource group</ResGroupName>
    <Description>test</Description>
    <NodeName>liux_test_n****</NodeName>
    <DqcType>1</DqcType>
    <Priority>1</Priority>
    <ParamValues>a=b</ParamValues>
    <Connection>odps_first</Connection>
    <ProjectId>33671</ProjectId>
    <RelatedFlowId>1235655464</RelatedFlowId>
    <SchedulerType>NORMAL</SchedulerType>
    <CronExpress>00 00 00 * * ?</CronExpress>
    <Repeatability>true</Repeatability>
    <OwnerId>19337906836551</OwnerId>
    <FileType>ODPS_SQL</FileType>
    <DqcDescription>[{"projectName":"ztjy_dim","tableName":"dim_user_agent_manage_area_a","partition":"ds\u003d${yyyy-mm-dd-1}"}]</DqcDescription>
    <NodeId>125677</NodeId>
    <ProgramType>ODPS_SQL</ProgramType>
    <RepeatInterval>60</RepeatInterval>
  </NodeList>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>The connection does not exist.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "SDFSDFSDF-asdfDFSDF-SDFSDF-SDFSDF",
  "HttpStatusCode": 200,
  "Data": {
    "Output": "test_0709_1.630003556_out",
    "NodeList": {
      "BaselineId": 1235667,
      "ResGroupName": "Default resource group",
      "Description": "test",
      "NodeName": "liux_test_n****",
      "DqcType": 1,
      "Priority": 1,
      "ParamValues": "a=b",
      "Connection": "odps_first",
      "ProjectId": 33671,
      "RelatedFlowId": 1235655464,
      "SchedulerType": "NORMAL",
      "CronExpress": "00 00 00 * * ?",
      "Repeatability": true,
      "OwnerId": 19337906836551,
      "FileType": "ODPS_SQL",
      "DqcDescription": "[{"projectName\\":\\"ztjy_dim\\",\\"tableName\\":\\"dim_user_agent_manage_area_a\\",\\"partition\\":\\"ds\\u003d${yyyy-mm-dd-1}\\"}]",
      "NodeId": 125677,
      "ProgramType": "ODPS_SQL",
      "RepeatInterval": 60
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "Success": true
}
```

2.1.6.4.42. RunTriggerNode

Runs a manually triggered node.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	RunTriggerNode	The operation that you want to perform.
RegionId	String	Yes	cn-shanghai	The ID of the region where the node resides.

Parameter	Type	Required	Example	Description
NodeId	Long	Yes	10000011	The ID of the manually triggered node. You can call the ListNodes operation to query the node ID.
CycleTime	Long	Yes	1606200230105	The timestamp for running the manually triggered node.
BizDate	Long	Yes	1606200230105	The data timestamp of the manually triggered node instance.
AppId	Long	Yes	10001	The ID of the DataWorks workspace to which the manually triggered node belongs. You can call the ListProjects operation to query the workspace ID.

Response parameters

Parameter	Type	Example	Description
RequestId	String	9BA675F1-F848-4752-A6E3-92ABA0616005	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful. - true: The call is successful. - false: The call fails.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=RunTriggerNode
&AppId=10001
&BizDate=1606200230105
&CycleTime=1606200230105
&NodeId=10000011
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>9BA675F1-F848-4752-A6E3-92ABA0616005</RequestId>
<Success>true</Success>
```

JSON format


```
{
  "RequestId": "9BA675F1-F848-4752-A6E3-92ABA0616005",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.Api.Risk	The request for this resource has exceeded your daily available limit.	The error message returned because the number of requests per day for this resource has exceeded the upper limit.
429	Throttling.Resource	You have reached the limit for the resource. Please contact us by submitting a ticket.	The error message returned because your quota for this resource has reached the upper limit. Submit a ticket for help.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
500	Forbidden.Not.Server	The dataworksAPI service is not available in this area. Please try it in another area.	The error message returned because the DataWorks API is unavailable in this region. Try another region.
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

2.1.6.4.43. GetNodeParents

Queries the ancestor nodes of a node.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetNodeParents	The operation that you want to perform.
NodeId	Long	Yes	12345456211234	The ID of the node. You can go to the Operation Center page in the DataWorks console to view the node ID.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned ancestor nodes.
Nodes	Array of Nodes	N/A	The list of ancestor nodes.
BaselineId	Long	1244564565	The ID of the baseline.
CronExpress	String	00 00 * * * *	The CRON expression. The CRON expression is used to run an auto triggered node.

Parameter	Type	Example	Description
NodeId	Long	1244564565	The ID of the node.
NodeName	String	test_Node	The name of the node.
OwnerId	String	123124561341251321	The ID of the Apsara Stack tenant account used by the node owner.
Priority	Integer	3	The priority of the node. Valid values: 1 to 8. A greater value indicates a higher priority.
ProgramType	String	ODPS_SQL	The type of the node.
ProjectId	Long	12315412412	The ID of the workspace to which the node belongs.
Repeatability	Boolean	true	Indicates whether the node can be rerun if the node fails to run. Valid values: - true: The node can be rerun. - false: The node cannot be rerun.
SchedulerType	String	NORMAL	The scheduling type of the node. Valid values: - NORMAL: The node is an auto triggered node. - MANUAL: The node is a manually triggered node. It is not run on a regular basis. - PAUSE: The node is a paused node. The scheduling system still runs the node but sets it to FAILURE when the scheduling system starts to run the node. - SKIP: The node is a dry-run node. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node.
ErrorCode	String	1060010000000	The error code returned.
ErrorMessage	String	err	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	asdfasdfas- adfasdf-asfas- dfasdf-asdf	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetNodeParents
&NodeId=12345456211234
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>asdfasdfas-adfasdf-asfas-dfasdf-asdf</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Nodes>
    <BaselineId>1244564565</BaselineId>
    <Repeatability>true</Repeatability>
    <OwnerId>123124561341251321</OwnerId>
    <NodeName>test_Node</NodeName>
    <Priority>3</Priority>
    <ProjectId>12315412412</ProjectId>
    <NodeId>1244564565</NodeId>
    <ProgramType>ODPS_SQL</ProgramType>
    <SchedulerType>NORMAL</SchedulerType>
    <CronExpress>00 00 * * * * </CronExpress>
  </Nodes>
</Data>
<ErrorCode>1060010000000</ErrorCode>
<ErrorMessage>err</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "asdfasdfas-adfasdf-asfas-dfasdf-asdf",
  "HttpStatusCode": 200,
  "Data": {
    "Nodes": {
      "BaselineId": 1244564565,
      "Repeatability": true,
      "OwnerId": 123124561341251330,
      "NodeName": "test_Node",
      "Priority": 3,
      "ProjectId": 12315412412,
      "NodeId": 1244564565,
      "ProgramType": "ODPS_SQL",
      "SchedulerType": "NORMAL",
      "CronExpress": "00 00 * * * *"
    }
  },
  "ErrorCode": 1060010000000,
  "ErrorMessage": "err",
  "Success": true
}
```

2.1.6.4.44. GetNodeChildren

Queries the descendant nodes of a node.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetNodeChildren	The operation that you want to perform.
NodeId	Long	Yes	123456878	The ID of the node. You can go to the Operation Center page in the DataWorks console to view the node ID.
ProjectEnv	String	Yes	PROD	The environment of the workspace. Valid values: PROD and DEV. A value of PROD indicates the production environment. A value of DEV indicates the development environment.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of returned descendant nodes.
Nodes	Array of Nodes	N/A	The list of descendant nodes.
BaselineId	Long	12345656	The ID of the baseline.
CronExpress	String	00 00 * * * *	The CRON expression. The CRON expression is used to run an auto triggered node.
NodeId	Long	1244564565	The ID of the node.
NodeName	String	test_Node	The name of the node.
OwnerId	String	123124561341251321	The ID of the Apsara Stack tenant account used by the node owner.
Priority	Integer	3	The priority of the node. Valid values: 1 to 8. A greater value indicates a higher priority.
ProgramType	String	ODPS_SQL	The type of the node.
ProjectId	Long	12315412412	The ID of the workspace to which the node belongs.

Parameter	Type	Example	Description
Repeatability	Boolean	true	Indicates whether the node can be rerun if the node fails to run. Valid values: • true: The node can be rerun. • false: The node cannot be rerun.
SchedulerType	String	NORMAL	The scheduling type of the node. Valid values: • NORMAL: The node is an auto triggered node. • MANUAL: The node is a manually triggered node. It is not run on a regular basis. ◦ PAUSE: The node is a paused node. The scheduling system still runs the node but sets it to FAILURE when the scheduling system starts to run the node. • SKIP: The node is a dry-run node. The scheduling system still runs the node but sets it to SUCCESS when the scheduling system starts to run the node.
ErrorCode	String	1060010000000	The error code returned.
ErrorMessage	String	err	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	adsfasdf-adf-asdf-asdf-asdfadfasdd	The ID of the request. You can use the ID to troubleshoot issues.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetNodeChildren
&NodeId=123456878
&ProjectEnv=PROD
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>adsfasdf-adf-asdf-asdf-asdfadfasdd</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Nodes>
    <BaselineId>12345656</BaselineId>
    <Repeatability>true</Repeatability>
    <OwnerId>123124561341251321</OwnerId>
    <NodeName>test_Node</NodeName>
    <Priority>3</Priority>
    <ProjectId>12315412412</ProjectId>
    <NodeId>1244564565</NodeId>
    <ProgramType>ODPS_SQL</ProgramType>
    <SchedulerType>NORMAL</SchedulerType>
    <CronExpress>00 00 * * * * </CronExpress>
  </Nodes>
</Data>
<ErrorCode>10600100000000</ErrorCode>
<ErrorMessage>err</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "adsfasdf-adf-asdf-asdf-asdfadfasdd",
  "HttpStatusCode": 200,
  "Data": {
    "Nodes": {
      "BaselineId": 12345656,
      "Repeatability": true,
      "OwnerId": 123124561341251330,
      "NodeName": "test_Node",
      "Priority": 3,
      "ProjectId": 12315412412,
      "NodeId": 1244564565,
      "ProgramType": "ODPS_SQL",
      "SchedulerType": "NORMAL",
      "CronExpress": "00 00 * * * *"
    }
  },
  "ErrorCode": 10600100000000,
  "ErrorMessage": "err",
  "Success": true
}
```

2.1.6.5. Data Quality

2.1.6.5.1. UpdateQualityFollower

Updates a subscriber.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateQualityFollower	The operation that you want to perform.
AlarmMode	Integer	Yes	2	The notification method. Valid values: 1, 2, 4, and 5. A value of 1 indicates that the notification is sent by email. A value of 2 indicates that the notification is sent by email and text message. A value of 4 indicates that the notification is sent by a DingTalk chatbot. A value of 5 indicates that the notification is sent by a DingTalk chatbot to all members in a DingTalk group.
Follower	String	Yes	1912232****	The ID of the subscriber.
FollowerId	Long	Yes	1234	The ID of the subscription relationship.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the subscriber is updated.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	576b9457-2cf5-4****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateQualityFollower
&AlarmMode=2
&Follower=1912232****
&FollowerId=1234
&ProjectName=autotest
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>576b9457-2cf5-4****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>>true</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "576b9457-2cf5-4****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission",
  "Success": true
}
```

2.1.6.5.2. UpdateQualityRule

Updates a monitoring rule.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateQualityRule	The operation that you want to perform.
Id	Long	Yes	1234	The ID of the monitoring rule.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
BlockType	Integer	No	0	The strength of the monitoring rule. Valid values: 0 and 1. A value of 0 indicates that the monitoring rule is a weak rule. A value of 1 indicates that the monitoring rule is a strong rule.
EntityId	Long	No	123	The ID of the partition filter expression.
Comment	String	No	Table row check	The description of the monitoring rule.
Checker	Integer	No	9	The ID of the checker.
ExpectValue	String	No	300	The expected value of the monitoring rule.
Trend	String	No	up	The trend of the monitoring results. Valid values: up, down, and abs.
MethodName	String	No	table_count	The method used to collect sample data.
Operator	String	No	>	The comparison operator of the monitoring rule.
Property	String	No	id	The field that is monitored.
PropertyType	String	No	bigint	The data type of the field.

Parameter	Type	Required	Example	Description
RuleType	Integer	No	0	The type of the monitoring rule. Valid values: 0, 1, and 2. A value of 0 indicates that the monitoring rule is created by the system. A value of 1 indicates that the monitoring rule is created by a user. A value of 2 indicates that the monitoring rule is a workspace-level rule.
WhereCondition	String	No	dt=[yyyymmdd]	The filter condition or custom SQL statement.
CriticalThreshold	String	No	10	The error threshold of the monitoring rule.
WarningThreshold	String	No	5	The warning threshold of the monitoring rule.
TemplateId	Integer	No	7	The ID of the monitoring template.
RegionId	String	No	cn-shanghai	The ID of the region.
RuleName	String	No	Table row check	The name of the monitoring rule.
PredictType	Integer	No	0	Specifies whether the threshold is a dynamic threshold. Valid values: 0 and 2. A value of 0 indicates that the monitoring rule is a statistical analysis rule. A value of 2 indicates that the threshold is a dynamic threshold.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the monitoring rule is updated.
ErrorCode	String	0	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	576b9457-2cf5-4****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateQualityRule
&Id=1234
&ProjectName=autotest
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>576b9457-2cf5-4****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "576b9457-2cf5-4****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 0,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.3. GetQualityRule

Queries the information about a monitoring rule.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	GetQualityRule	The operation that you want to perform.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
RuleId	Long	Yes	1234	The ID of the monitoring rule.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The information about the monitoring rule.
BlockType	Integer	1	The strength of the monitoring rule. The strength of monitoring rules indicates the importance of the rules. Valid values: • 1: The monitoring rule is a strong rule. • 0: The monitoring rule is a weak rule. You can specify whether a monitoring rule is a strong rule based on your business requirements. If a strong rule is used and an error alert is reported, nodes are blocked.
Checker	Integer	9	The ID of the checker. The value of this parameter corresponds to the ID at the frontend and is converted from the ID of the primary key.
CheckerName	String	compared with a fixed value	The name of the checker.
Comment	String	Unique primary key verification	The description of the monitoring rule.

Parameter	Type	Example	Description
CriticalThreshold	String	20	The error threshold of the monitoring rule. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements. If a strong rule is used and an error alert is reported, nodes are blocked.
EntityId	Long	165523	The ID of the partition filter expression.
ExpectValue	String	30	The expected value of the monitoring rule.
FixCheck	Boolean	true	Indicates whether the monitoring is performed based on a fixed value.
Id	Long	123232	The ID of the monitoring rule.
MethodId	Integer	8	The ID of the method used to collect sample data.
MethodName	String	table_count	The method used to collect sample data, such as avg, count, sum, min, max, count_distinct, user_defined, table_count, table_size, table_dt_load_count, table_dt_refuseload_count, null_value, null_value/table_count, (table_count - count_distinct)/table_count, or table_count - count_distinct.
OnDuty	String	1822931****	The ID of the Apsara Stack tenant account used to configure the monitoring rule.
OnDutyAccount Name	String	test	The name of the Apsara Stack tenant account used to configure the monitoring rule.

Parameter	Type	Example	Description
Operator	String	>=	The comparison operator of the monitoring rule.
PredictType	Integer	0	Indicates whether the threshold is a dynamic threshold. Valid values: 0: The threshold is not a dynamic threshold. 1: The threshold is a dynamic threshold.
Property	String	id	The field that is monitored. This field is a column name of the monitored table in the data source.
RuleName	String	View fluctuations in a table	The name of the monitoring rule.
RuleType	Integer	0	The type of the monitoring rule.
TemplateId	Integer	7	The ID of the monitoring template.
TemplateName	String	SQL task table rows, 1,7, 30 days fluctuation test	The name of the monitoring template.
Trend	String	abs	The trend of the monitoring results.
WarningThreshold	String	10	The warning threshold of the monitoring rule. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements.
WhereCondition	String	id>10	The filter condition or custom SQL statement that is used for monitoring.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.

Parameter	Type	Example	Description
RequestId	String	576b9457-2cf5-4****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetQualityRule
&ProjectName=autotest
&RegionId=cn-shanghai
&RuleId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>576b9457-2cf5-4****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Operator>&gt;&gt;</Operator>
  <Comment>Unique primary key verification</Comment>
  <Checker>9</Checker>
  <CriticalThreshold>20</CriticalThreshold>
  <CheckerName>compared with a fixed value</CheckerName>
  <ExpectValue>30</ExpectValue>
  <WarningThreshold>10</WarningThreshold>
  <OnDuty>1822931****</OnDuty>
  <OnDutyAccountName>test</OnDutyAccountName>
  <Trend>abs</Trend>
  <WhereCondition>id&gt;10</WhereCondition>
  <RuleType>0</RuleType>
  <EntityId>165523</EntityId>
  <MethodId>8</MethodId>
  <FixCheck>true</FixCheck>
  <BlockType>1</BlockType>
  <TemplateName>SQL task table rows, 1,7, 30 days fluctuation test</TemplateName>
  <PredictType>0</PredictType>
  <Id>123232</Id>
  <MethodName>table_count</MethodName>
  <Property>id</Property>
  <TemplateId>7</TemplateId>
  <RuleName>View fluctuations in a table</RuleName>
</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "576b9457-2cf5-4***",
  "HttpStatusCode": 200,
  "Data": {
    "Operator": ">=",
    "Comment": "Unique primary key verification",
    "Checker": 9,
    "CriticalThreshold": 20,
    "CheckerName": "compared with a fixed value",
    "ExpectValue": 30,
    "WarningThreshold": 10,
    "OnDuty": "1822931****",
    "OnDutyAccountName": "test",
    "Trend": "abs",
    "WhereCondition": "id>10",
    "RuleType": 0,
    "EntityId": 165523,
    "MethodId": 8,
    "FixCheck": true,
    "BlockType": 1,
    "TemplateName": "SQL task table rows, 1,7, 30 days fluctuation test",
    "PredictType": 0,
    "Id": 123232,
    "MethodName": "table_count",
    "Property": "id",
    "TemplateId": 7,
    "RuleName": "View fluctuations in a table"
  },
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.4. ListQualityRules

Queries monitoring rules based on a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListQualityRules	The operation that you want to perform.
EntityId	Long	Yes	1234	The ID of the partition filter expression. You can call the GetQualityEntity operation to query the ID of the partition filter expression.

Parameter	Type	Required	Example	Description
PageNumber	Integer	Yes	1	The number of the page to return.
PageSize	Integer	Yes	20	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection. You can obtain the name from connection configurations.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The list of monitoring rules.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	20	The number of entries returned per page. Default value: 10. Maximum value: 100.
Rules	Array of Rules	N/A	The details of monitoring rules.

Parameter	Type	Example	Description
BlockType	Integer	0	The strength of the monitoring rule. The strength of monitoring rules indicates the importance of the rules. Valid values: 1: The monitoring rule is a strong rule. 0: The monitoring rule is a weak rule. You can specify whether a monitoring rule is a strong rule based on your business requirements. If a strong rule is used and an error alert is reported, nodes are blocked.
CheckerId	Integer	7	The ID of the checker.
Comment	String	Monitoring rule for a table	The description of the monitoring rule.
CriticalThreshold	String	40	The error threshold of the monitoring rule. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements. If a strong rule is used and an error alert is reported, nodes are blocked.
EntityId	Integer	1234	The ID of the partition filter expression.
ExpectValue	String	1000	The expected value of the monitoring rule.
FixCheck	Boolean	true	Indicates whether the monitoring rule is a strong rule.
HistoryCriticalThreshold	String	history max:80%,history min:50%	The historical error thresholds of the monitoring rule.
HistoryWarningThreshold	String	history max:40%,history min:10%	The historical warning thresholds of the monitoring rule.
Id	Integer	1234	The ID of the monitoring rule.
MatchExpression	String	dt=\${yyyymmdd}	The partition filter expression.

Parameter	Type	Example	Description
MethodId	Integer	21	The ID of the method used to collect sample data.
MethodName	String	count/table_count	The method used to collect sample data, such as avg, count, sum, min, max, count_distinct, user_defined, table_count, table_size, table_dt_load_count, table_dt_refuseload_count, null_value, null_value/table_count, (table_count - count_distinct)/table_count, or table_count - count_distinct.
OnDuty	String	1912****	The ID of the Apsara Stack tenant account used to configure the monitoring rule.
OnDutyAccount Name	String	test	The name of the Apsara Stack tenant account used to configure the monitoring rule.
ProjectName	String	autotest	The name of the compute engine instance or connection.
Property	String	id	The field that is monitored.
PropertyKey	String	table_count	The field that is used to associate with monitoring rules at the frontend. This parameter can be ignored.
RuleCheckerRelationId	Integer	132323	The ID of the node that is associated with the partition filter expression.

Parameter	Type	Example	Description
RuleType	Integer	0	The type of the monitoring rule. Valid values: • 0: The monitoring rule is created by the system. • 1: The monitoring rule is created by a user. • 2: The monitoring rule is a workspace-level rule.
TableName	String	dual	The name of the table.
TemplateId	Integer	7	The ID of the monitoring template.
TemplateName	String	Check for the fluctuations of table rows on an SQL node on the first, seventh, and thirtieth days	The name of the monitoring template.
Trend	String	abs	The trend of the monitoring results.
WarningThreshold	String	10	The warning threshold of the monitoring rule. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements.
TotalCount	Long	400	The total number of entries returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	38cbdef0-f6cf-49****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListQualityRules
&EntityId=1234
&PageNumber=1
&PageSize=20
&ProjectName=autotest
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>38cbdef0-f6cf-49****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>400</TotalCount>
  <PageSize>20</PageSize>
  <PageNumber>1</PageNumber>
  <Rules>
    <HistoryCriticalThreshold>history max:80%,history min:50%</HistoryCriticalThreshold>
    <Comment>Monitoring rule for a table</Comment>
    <HistoryWarningThreshold>history max:40%,history min:10%</HistoryWarningThreshold>
    <RuleCheckerRelationId>132323</RuleCheckerRelationId>
    <CriticalThreshold>40</CriticalThreshold>
    <ProjectName>autotest</ProjectName>
    <ExpectValue>1000</ExpectValue>
    <WarningThreshold>10</WarningThreshold>
    <OnDuty>1912****</OnDuty>
    <OnDutyAccountName>test</OnDutyAccountName>
    <Trend>abs</Trend>
    <RuleType>0</RuleType>
    <CheckerId>7</CheckerId>
    <MatchExpression>dt=${yyyymmdd}</MatchExpression>
    <EntityId>1234</EntityId>
    <TableName>dual</TableName>
    <MethodId>21</MethodId>
    <FixCheck>true</FixCheck>
    <BlockType>0</BlockType>
    <TemplateName>Check for the fluctuations of table rows on an SQL node on the first, seventh, and thirtieth days</TemplateName>
    <Id>1234</Id>
    <MethodName>count/table_count</MethodName>
    <Property>id</Property>
    <TemplateId>7</TemplateId>
    <PropertyKey>table_count</PropertyKey>
  </Rules>
</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "38cbdef0-f6cf-49****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 400,
    "PageSize": 20,
    "PageNumber": 1,
    "Rules": {
      "HistoryCriticalThreshold": "history max:80%,history min:50%",
      "Comment": "Monitoring rule for a table",
      "HistoryWarningThreshold": "history max:40%,history min:10%",
      "RuleCheckerRelationId": 132323,
      "CriticalThreshold": 40,
      "ProjectName": "autotest",
      "ExpectValue": 1000,
      "WarningThreshold": 10,
      "OnDuty": "1912****",
      "OnDutyAccountName": "test",
      "Trend": "abs",
      "RuleType": 0,
      "CheckerId": 7,
      "MatchExpression": "dt=${yyyymmdd}",
      "EntityId": 1234,
      "TableName": "dual",
      "MethodId": 21,
      "FixCheck": true,
      "BlockType": 0,
      "TemplateName": "Check for the fluctuations of table rows on an SQL node on the first, seventh, and thirtieth days",
      "Id": 1234,
      "MethodName": "count/table_count",
      "Property": "id",
      "TemplateId": 7,
      "PropertyKey": "table_count"
    }
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.5. CreateQualityRule

Creates a monitoring rule.

Request parameters

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateQualityRule	The operation that you want to perform.
BlockType	Integer	Yes	0	The strength of the monitoring rule. Valid values: 0 and 1. A value of 0 indicates that the monitoring rule is a weak rule. A value of 1 indicates that the monitoring rule is a strong rule.
Checker	Integer	Yes	9	The ID of the checker.
EntityId	Long	Yes	15224	The ID of the partition filter expression.
MethodName	String	Yes	count/table_count	The method used to collect sample data. If you want to use a custom SQL statement as a sampling method, set this parameter to user_define.
PredictType	Integer	Yes	0	Specifies whether the threshold is a dynamic threshold. Valid values: 0 and 1. A value of 0 indicates that the threshold is not a dynamic threshold. A value of 1 indicates that the threshold is a dynamic threshold.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RuleName	String	Yes	1234	The name of the monitoring rule.
RuleType	Integer	Yes	0	The type of the monitoring rule. Valid values: 0, 1, and 2. A value of 0 indicates that the monitoring rule is created by the system. A value of 1 indicates that the monitoring rule is created by a user. A value of 2 indicates that the monitoring rule is a workspace-level rule.
Comment	String	No	Verification	The description of the monitoring rule.

Parameter	Type	Required	Example	Description
ExpectValue	String	No	0	The expected value of the monitoring rule.
Trend	String	No	abs	The trend of the monitoring results.
Operator	String	No	>	The comparison operator of the monitoring rule.
Property	String	No	table_id	The field that you want to monitor. If you want to monitor all fields in a table and check the table rows, set this parameter to table_count. If you want to monitor all fields in a table and check the table size, set this parameter to table_size.
PropertyType	String	No	bigint	The data type of the field that you want to monitor. If you want to monitor all fields in a table, set this parameter to table. If you want to monitor only a specific field, set this parameter to bigint.
WhereCondition	String	No	table_id>1	The filter condition or custom SQL statement.
CriticalThreshold	String	No	20	The warning threshold of the monitoring rule.
WarningThreshold	String	No	10	The error threshold of the monitoring rule.
TemplateId	Integer	No	7	The ID of the monitoring template.

Response parameters

Parameter	Type	Example	Description
Data	String	12345	The ID of the created monitoring rule.
ErrorCode	String	401	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	ecb967ec-c137-48****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateQualityRule
&BlockType=0
&Checker=9
&EntityId=15224
&MethodName=count/table_count
&PredictType=0
&ProjectName=autotest
&RegionId=cn-shanghai
&RuleName=1234
&RuleType=0
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>ecb967ec-c137-48****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>1234</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "ecb967ec-c137-48****",
  "HttpStatusCode": 200,
  "Data": 1234,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.6. DeleteQualityFollower

Deletes a subscriber for a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteQualityFollower	The operation that you want to perform.
FollowerId	Long	Yes	1234	The ID of the subscriber.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the subscriber is deleted.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	6d739ef6-098a-47****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteQualityFollower
&FollowerId=1234
&ProjectName=autotest
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>6d739ef6-098a-47****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "6d739ef6-098a-47****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.7. DeleteQualityRule

Deletes a monitoring rule.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteQualityRule	The operation that you want to perform.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.
RuleId	Long	Yes	1234	The ID of the monitoring rule.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the monitoring rule is deleted.

Parameter	Type	Example	Description
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	6d739ef6-098a-47****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteQualityRule
&ProjectName=autotest
&RegionId=cn-shanghai
&RuleId=1234
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>6d739ef6-098a-47****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "6d739ef6-098a-47****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.8. GetQualityFollower

Queries the subscribers for a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetQualityFollower	The operation that you want to perform.
EntityId	Long	Yes	1234	The ID of the partition filter expression.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Array of Follower	N/A	The list of subscribers.
AlarmMode	Integer	1	The notification method. Valid values: 1: The notification is sent by email. 2: The notification is sent by email and text message. 4: The notification is sent by using a DingTalk chatbot or a webhook URL. 5: The notification is sent by a DingTalk chatbot to all members in a DingTalk group.
EntityId	String	1234	The ID of the partition filter expression.
Follower	String	1234	The subscriber for receiving alert notifications.
FollowerAccountName	String	test	The name of the Apsara Stack tenant account used by the subscriber.

Parameter	Type	Example	Description
Id	Long	123	The ID of the subscription relationship.
ProjectName	String	autotest	The name of the compute engine instance or connection.
TableName	String	dual	The name of the partitioned table.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	38cbdef0-f6cf-49	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetQualityFollower
&EntityId=1234
&ProjectName=autotest
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format


```
<RequestId>38cbdef0-f6cf-49****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <EntityId>1234</EntityId>
  <TableName>dual</TableName>
  <ProjectName>autotest</ProjectName>
  <Follower>1822931104031845</Follower>
  <FollowerAccountName>test</FollowerAccountName>
  <Id>123</Id>
  <AlarmMode>1</AlarmMode>
</Data>
<ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "38cbdef0-f6cf-49",
  "HttpStatusCode": 200,
  "Data": {
    "EntityId": 1234,
    "TableName": "dual",
    "ProjectName": "autotest",
    "Follower": 1822931104031845,
    "FollowerAccountName": "test",
    "Id": 123,
    "AlarmMode": 1
  },
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.9. CreateQualityFollower

Creates a subscriber for a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateQualityFollower	The operation that you want to perform.

Parameter	Type	Required	Example	Description
AlarmMode	Integer	Yes	1	The notification method. Valid values: 1, 2, 4, and 5. A value of 1 indicates that the notification is sent by email. A value of 2 indicates that the notification is sent by email and text message. A value of 4 indicates that the notification is sent by a DingTalk chatbot. A value of 5 indicates that the notification is sent by a DingTalk chatbot to all members in a DingTalk group.
EntityId	Long	Yes	12345	The ID of the partition filter expression.
Follower	String	Yes	123	The ID of the Apsara Stack tenant account used by the subscriber.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Response parameters

Parameter	Type	Example	Description
Data	Integer	12345	The ID of the created subscription relationship.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	ecb967ec-c137-48****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateQualityFollower
&AlarmMode=1
&EntityId=12345
&Follower=123
&ProjectName=autotest
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>ecb967ec-c137-48****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>12345</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "ecb967ec-c137-48****",
  "HttpStatusCode": 200,
  "Data": 12345,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.10. DeleteQualityEntity

Deletes a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteQualityEntity	The operation that you want to perform.
EntityId	Long	Yes	1234	The ID of the partition filter expression.
EnvType	String	Yes	ODPS	The type of the compute engine or connection.

Parameter	Type	Required	Example	Description
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	The result returned.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	ecb967ec-c137-48****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteQualityEntity
&EntityId=1234
&EnvType=ODPS
&ProjectName=autotest
&RegionId=cn-shanghai
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>ecb967ec-c137-48****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "ecb967ec-c137-48****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.11. GetQualityEntity

Queries the details of a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetQualityEntity	The operation that you want to perform.
EnvType	String	Yes	odps	The type of the compute engine or connection.
MatchExpression	String	Yes	dt=\${yyyymmdd}	The partition filter expression.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection. You can obtain the name from connection configurations.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.
TableName	String	Yes	dual	The name of the partitioned table. You can call the GetMetaTablePartition operation to query the table name.

Response parameters

Parameter	Type	Example	Description
Data	Array of EntityDto	N/A	The details of the partition filter expression.
CreateTime	Long	1593964800000	The time when the partition filter expression was created. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
EntityLevel	Integer	0	The level of the partition filter expression. Valid values: • 0: The partition filter expression is at the SQL level. This indicates that the system checks data quality after each SQL statement is executed. • 1: The partition filter expression is at the node level. This indicates that the system checks data quality after all the SQL statements for a node are executed.
EnvType	String	odps	The type of the compute engine or connection.
Followers	String	1822931****	The ID of the Apsara Stack tenant account used by the subscriber who receives alert notifications.
HasRelativeNode	Boolean	true	Indicates whether the partition filter expression is associated with a node. Valid values: • true: The partition filter expression is associated with a node. • false: The partition filter expression is not associated with a node.
Id	Long	4003918	The ID of the partition filter expression.
MatchExpression	String	dt=\${yyyymmdd-1}	The partition filter expression.

Parameter	Type	Example	Description
ModifyTime	Long	1593964800000	The time when the partition filter expression was last modified. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
ModifyUser	String	1822931****	The ID of the Apsara Stack tenant account used by the user who last modified the partition filter expression.
OnDuty	String	1822931****	The ID of the Apsara Stack tenant account used by the owner who configured the partition filter expression.
OnDutyAccount Name	String	test	The name of the Apsara Stack tenant account used by the owner.
ProjectName	String	autotest	The name of the compute engine instance or connection.
RelativeNode	String	<pre>{["projectName": "xc_DP****", "nodeId": 7000026****]}</pre>	The information about the node that is associated with the partition filter expression. The information includes the following items: - projectName: the name of the workspace to which the node belongs. - nodeId: the ID of the node.
Sql	Integer	0	Indicates whether the partition filter expression is at the SQL level.
TableName	String	test_dqc_de****	The name of the partitioned table.
Task	Integer	0	Indicates whether the partition filter expression is at the node level.
ErrorCode	String	401	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	6d739ef6-098a-47****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetQualityEntity
&EnvType=ODPS
&MatchExpression=dt=${yyyymmdd}
&ProjectName=autotest
&RegionId=cn-shanghai
&TableName=dual
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>6d739ef6-098a-47****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <ModifyTime>1593964800000</ModifyTime>
  <Task>0</Task>
  <ModifyUser>1822931104031845</ModifyUser>
  <ProjectName>autotest</ProjectName>
  <HasRelativeNode>true</HasRelativeNode>
  <CreateTime>1593964800000</CreateTime>
  <OnDuty>1822931****</OnDuty>
  <OnDutyAccountName>test</OnDutyAccountName>
  <Followers>1822931****</Followers>
  <MatchExpression>dt=${yyyymmdd-1}</MatchExpression>
  <EnvType>odps</EnvType>
  <Sql>0</Sql>
  <TableName>test_dqc_de****</TableName>
  <Id>4003918</Id>
  <RelativeNode>[{"projectName":"xc_DP****","nodeId":7000026****}]</RelativeNode>
  <EntityLevel>
</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```


JSON format

```
{
  "RequestId": "6d739ef6-098a-47****",
  "HttpStatusCode": 200,
  "Data": {
    "ModifyTime": 1593964800000,
    "Task": 0,
    "ModifyUser": 1822931104031845,
    "ProjectName": "autotest",
    "HasRelativeNode": true,
    "CreateTime": 1593964800000,
    "OnDuty": "1822931****",
    "OnDutyAccountName": "test",
    "Followers": "1822931****",
    "MatchExpression": "dt=${yyyymmdd-1}",
    "EnvType": "odps",
    "Sql": 0,
    "TableName": "test_dqc_de****",
    "Id": 4003918,
    "RelativeNode": "[{\"projectId\":\"xc_DP****\",\"nodeId\":\"7000026****\"}]",
    "EntityLevel": ""
  },
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.12. CreateQualityEntity

Creates a partition filter expression in Data Quality.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateQualityEntity	The operation that you want to perform.
EntityLevel	Integer	Yes	0	Specifies whether the partition filter expression is at the SQL level or node level. Valid values: 0 and 1. A value of 0 indicates that the partition filter expression is at the SQL level, and the system checks data quality after each SQL statement is executed. A value of 1 indicates that the partition filter expression is at the node level, and the system checks data quality after a node is run.

Parameter	Type	Required	Example	Description
EnvType	String	Yes	ODPS	The type of the compute engine or connection.
MatchExpression	String	Yes	dt=\${yyyymmdd}	The partition filter expression.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region where the service is activated.
TableName	String	Yes	dual	The name of the table.

Response parameters

Parameter	Type	Example	Description
Data	Integer	12345	The ID of the created partition filter expression.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	Your project is not relative with your account.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	011e1231u3****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateQualityEntity
&EntityLevel=0
&EnvType=ODPS
&MatchExpression=dt=${yyyymmdd]
&ProjectName=autotest
&RegionId=cn-shanghai
&TableName=dual
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>011e1231u3****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>12345</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>Your project is not relative with your account.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "011e1231u3****",
  "HttpStatusCode": 200,
  "Data": 12345,
  "ErrorCode": 401,
  "ErrorMessage": "Your project is not relative with your account.",
  "Success": true
}
```

2.1.6.5.13. CreateQualityRelativeNode

Associates a node with a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateQualityRelativeNode	The operation that you want to perform.
EnvType	String	Yes	ODPS	The type of the engine or data source.
MatchExpression	String	Yes	dt=\${yyyymmdd]	The partition filter expression.
NodeId	Long	Yes	12321	The ID of the node.

Parameter	Type	Required	Example	Description
ProjectId	Long	Yes	123	The ID of the workspace.
ProjectName	String	Yes	autotest	The name of the compute engine or data source.
RegionId	String	Yes	cn-shanghai	The region ID.
TableName	String	Yes	dual	The name of the table.
TargetNodeProjectId	Long	Yes	1234	The ID of the node that is associated with the partition filter expression.
TargetNodeProjectName	String	Yes	autotest	The name of the project that corresponds to the workspace to which the associated node belongs.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the node is associated with the partition filter expression.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	ecb967ec-c137-48****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateQualityRelativeNode
&EnvType=ODPS
&MatchExpression=dt=${yyyyymmdd}
&NodeId=12321
&ProjectId=123
&ProjectName=autotest
&RegionId=cn-shanghai
&TableName=dual
&TargetNodeProjectId=1234
&TargetNodeProjectName=autotest
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>ecb967ec-c137-48****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "ecb967ec-c137-48****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.14. DeleteQualityRelativeNode

Disassociates a node from a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteQualityRelativeNode	The operation that you want to perform.
EnvType	String	Yes	ODPS	The type of the compute engine or connection.

Parameter	Type	Required	Example	Description
MatchExpression	String	Yes	dt=\${yyyymmdd}	The partition filter expression.
NodeId	Long	Yes	156234	The ID of the node.
ProjectId	Long	Yes	12345	The ID of the workspace.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TableName	String	Yes	dual	The name of the table.
TargetNodeProjectId	Long	Yes	12345	The ID of the workspace to which the node associated with the partition filter expression belongs.
TargetNodeProjectName	String	Yes	autotest	The name of the workspace to which the node associated with the partition filter expression belongs.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the node is disassociated.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	6d739ef6-098a-47****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteQualityRelativeNode
&EnvType=ODPS
&MatchExpression=dt=${yyyymmdd}
&NodeId=156234
&ProjectId=12345
&ProjectName=autotest
&RegionId=cn-shanghai
&TableName=dual
&TargetNodeProjectId=12345
&TargetNodeProjectName=autotest
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>6d739ef6-098a-47****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<ErrorCode>401</ErrorCode>
<ErrorMessage>You have no permission.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "6d739ef6-098a-47****",
  "HttpStatusCode": 200,
  "Data": true,
  "ErrorCode": 401,
  "ErrorMessage": "You have no permission.",
  "Success": true
}
```

2.1.6.5.15. ListQualityResultsByRule

Queries monitoring results after the quality of a data source or an engine is monitored based on monitoring rules.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListQualityResultsByRule	The operation that you want to perform.

Parameter	Type	Required	Example	Description
EndDate	String	Yes	2020-09-22 00:00:00	The end of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.
PageNumber	Integer	Yes	1	The number of the page to return.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectName	String	Yes	autotest	The name of the engine or data source.
RuleId	Integer	Yes	152322134	The ID of the monitoring rule. You can use the ID and information such as a partition filter expression to perform a joint query.
StartDate	String	Yes	2020-09-20 00:00:00	The beginning of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.
RegionId	String	No	cn-hangzhou	The region ID. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The top structure of the monitoring results.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.

Parameter	Type	Example	Description
RuleChecks	Array of RuleChecks		The monitoring results returned.
ActualExpression	String	ds=20200925	The partition of the monitored table in the data source.
BeginTime	Long	1600704000000	The time when the monitoring started.
BizDate	Long	1600704000000	The data timestamp. If the monitored business entity is offline data, the value is usually one day before the monitoring is performed.
BlockType	Integer	1	The type of the monitoring rule. The strength of monitoring rules indicates the importance of the rules. Valid values: • 1: The monitoring rule is a strong rule. • 0: The monitoring rule is a weak rule. You can specify whether a monitoring rule is a strong rule based on your business requirements. If a strong rule is used and error alerts are reported, nodes are blocked.
CheckResult	Integer	2	The monitoring result. Valid values: 0, 1, and 2. The value 0 indicates that the monitoring result is normal. The value 1 indicates that a warning alert is reported. The value 2 indicates that an error alert is reported.
CheckResultStatus	Integer	2	The status of the monitoring result. Valid values: 0, 1, and 2. The value 0 indicates that the monitoring result is in a normal state. The value 1 indicates that an alert is reported. The value 2 indicates that nodes are blocked.
CheckerId	Integer	7	The ID of the checker.
CheckerName	String	fulx	The name of the checker.

Parameter	Type	Example	Description
CheckerType	Integer	0	The type of the checker.
Comment	String	Description of the monitoring rule	The description of the monitoring rule.
CriticalThreshold	Float	0.6	The error threshold. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements. If a strong rule is used and error alerts are reported, nodes are blocked.
DateType	String	YMD	The type of the scheduling cycle. Valid values: YMD.
DiscreteCheck	Boolean	true	Indicates whether the monitoring is discrete monitoring. Valid values: - true: indicates that the monitoring is discrete monitoring. - false: indicates that the monitoring is not discrete monitoring.
EndTime	Long	1600704000000	The end time of the query.
EntityId	Integer	14534343	The ID of the partition filter expression.
ExpectValue	Float	90	The expected value.
ExternalId	String	123112232	The ID of the node.
ExternalType	String	CWF2	The type of the scheduling system. Only CWF scheduling systems are supported.
FixedCheck	Boolean	false	Indicates whether the monitoring for a fixed value is performed. Valid values: - true: indicates that the monitoring for a fixed value is performed. - false: indicates whether the monitoring for a fixed value is not performed.

Parameter	Type	Example	Description
Id	Integer	2231123	The ID of the primary key.
IsPrediction	Boolean	false	Indicates whether the monitoring result is consistent with the predicted result. Valid values: - true: indicates that the monitoring result is consistent with the predicted result. - false: indicates that the monitoring result is inconsistent with the predicted result.
LowerValue	Float	2344	The lower limit of the predicted result. The value of this parameter is automatically generated based on the threshold that you specified.
MatchExpression	String	ds=\${yyyymmdd}	The partition filter expression.
MethodName	String	max	The sampling method. The value of this parameter includes avg, count, sum, min, max, count_distinct, user_defined, table_count, table_size, table_dt_load_count, table_dt_refuseload_count, null_value, null_value/table_count, (table_count-count_distinct)/table_count, table_count-count_distinct.
Op	String	>	The comparison operator.
ProjectName	String	autotest	The name of the engine or data source.
Property	String	type	The field that is monitored. This field is a column name in the data source table that is monitored.
ReferenceValue	Array of ReferenceValue		The historical sample value.
BizDate	String	1600704000000	The data timestamp. If the monitored business entity is offline data, the value is usually one day before the monitoring is performed.

Parameter	Type	Example	Description
DiscreteProperty	String	type1	The values of the sample field that are grouped by using the GROUP BY statement. For example, the values of the Gender field are grouped by using the GROUP BY statement. In this case, the values of DiscreteProperty are Male, Female, and null.
SingleCheckResult	Integer	2	The string of the monitoring result.
Threshold	Float	0.2	The threshold.
Value	Float	20	The sample value.
ResultString	String	ResultString	The string of the monitoring result.
RuleId	Integer	123421	The ID of the monitoring rule.
RuleName	String	Name of the monitoring rule	The name of the monitoring rule.
SampleValue	Array of SampleValue		The sample value.
BizDate	String	1600704000000	The data timestamp. If the monitored business entity is offline data, the value is usually one day before the monitoring is performed.
DiscreteProperty	String	type2	The values of the sample field that are grouped by using the GROUP BY statement. For example, the values of the Gender field are grouped by using the GROUP BY statement. In this case, the values of DiscreteProperty are Male, Female, and null.
Value	Float	23	The sample value.
TableName	String	dual	The name of the table that is monitored.
TaskId	String	16008552981681a0d6****	The ID of the monitoring task.

Parameter	Type	Example	Description
TemplateId	Integer	5	The ID of the monitoring template.
TemplateName	String	Compare the Number of Unique Field Values Against Expectation	The name of the monitoring template.
TimeCost	String	10	The time that was taken to run the monitoring task.
Trend	String	up	The trend of the monitoring result.
UpperValue	Float	22200	The upper limit of the predicted result. The upper limit is automatically generated based on the threshold that you specified.
WarningThreshold	Float	0.1	The warning threshold. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements.
WhereCondition	String	type!='type2'	The filter condition of the monitoring rule.
TotalCount	Long	200	The total number of entries returned.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	CBA58543-00D4-41****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```

http(s)://[Endpoint]/?Action=ListQualityResultsByRule
&EndDate=2020-09-22 00:00:00
&PageNumber=1
&PageSize=10
&ProjectName=autotest
&RuleId=152322134
&StartDate=2020-09-20 00:00:00
&<Common request parameters>

```

Sample success responses

XML format

```

<RequestId>CBA58543-00D4-41****</RequestId>
<QualityResults>
  <TotalCount>200</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <RuleChecks>
    <Op>&gt;</Op>
    <TaskId>16008552981681a0d6****</TaskId>
    <EndTime>1600704000000</EndTime>
    <DiscreteCheck>true</DiscreteCheck>
    <CheckerName>fulx</CheckerName>
    <ExpectValue>90</ExpectValue>
    <ProjectName>autotest</ProjectName>
    <RuleId>123421</RuleId>
    <WarningThreshold>0.1</WarningThreshold>
    <WhereCondition>type!='type2'</WhereCondition>
    <BeginTime>1600704000000</BeginTime>
    <CheckResultStatus>2</CheckResultStatus>
    <MatchExpression>ds=${yyyymmdd}</MatchExpression>
    <UpperValue>22200</UpperValue>
    <IsPrediction>false</IsPrediction>
    <CheckResult>2</CheckResult>
    <TemplateName>Compare the Number of Unique Field Values Against Expectation</TemplateName>
    <TimeCost>10</TimeCost>
    <FixedCheck>false</FixedCheck>
    <MethodName>max</MethodName>
    <ResultString>ResultString</ResultString>
    <Property>type</Property>
    <DateType>YMD</DateType>
    <BizDate>1600704000000</BizDate>
    <TemplateId>5</TemplateId>
    <RuleName>Name of the monitoring rule</RuleName>
    <Comment>Description of the monitoring rule</Comment>
    <CriticalThreshold>0.6</CriticalThreshold>
    <ExternalId>123112232</ExternalId>
    <Trend>up</Trend>
    <CheckerType>0</CheckerType>
    <ExternalType>CWF2</ExternalType>
    <CheckerId>7</CheckerId>
    <EntityId>14534343</EntityId>
    <TableName>dual</TableName>

```

```

    <BlockType>1</BlockType>
    <Id>2231123</Id>
    <ActualExpression>ds=20200925</ActualExpression>
    <LowerValue>2344</LowerValue>
  </RuleChecks>
  <RuleChecks>
    <SampleValue>
      <SingleCheckResult>2</SingleCheckResult>
      <Value>20</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>type1</DiscreteProperty>
      <Threshold>0.2</Threshold>
    </SampleValue>
    <SampleValue>
      <Value>23</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>type2</DiscreteProperty>
    </SampleValue>
    <ReferenceValue>
      <SingleCheckResult>2</SingleCheckResult>
      <Value>20</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>type1</DiscreteProperty>
      <Threshold>0.2</Threshold>
    </ReferenceValue>
    <ReferenceValue>
      <Value>23</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>type2</DiscreteProperty>
    </ReferenceValue>
  </RuleChecks>
</QualityResults>

```

JSON format

```

{
  "RequestId": "CBA58543-00D4-41****",
  "QualityResults": {
    "TotalCount": 200,
    "PageSize": 10,
    "PageNumber": 1,
    "RuleChecks": [
      {
        "Op": "&gt;",
        "TaskId": "16008552981681a0d6****",
        "EndTime": 1600704000000,
        "DiscreteCheck": true,
        "CheckerName": "fulx",
        "ExpectValue": 90,
        "ProjectName": "autotest",
        "RuleId": 123421,
        "WarningThreshold": 0.1,
        "WhereCondition": "type!='type2'",
        "BeginTime": 1600704000000
      }
    ]
  }
}

```

```

    beginTime : 1600704000000,
    "CheckResultStatus": 2,
    "MatchExpression": "ds=[yyyymmdd]",
    "UpperValue": 22200,
    "IsPrediction": false,
    "CheckResult": 2,
    "TemplateName": "Compare the Number of Unique Field Values Against Expectation",
    "TimeCost": 10,
    "FixedCheck": false,
    "MethodName": "max",
    "ResultString": "ResultString",
    "Property": "type",
    "DateType": "YMD",
    "BizDate": 1600704000000,
    "TemplateId": 5,
    "RuleName": "Name of the monitoring rule",
    "Comment": "Description of the monitoring rule",
    "CriticalThreshold": 0.6,
    "ExternalId": 123112232,
    "Trend": "up",
    "CheckerType": 0,
    "ExternalType": "CWF2",
    "CheckerId": 7,
    "EntityId": 14534343,
    "TableName": "dual",
    "BlockType": 1,
    "Id": 2231123,
    "ActualExpression": "ds=20200925",
    "LowerValue": 2344
  },
  {
    "SampleValue": [
      {
        "SingleCheckResult": 2,
        "Value": 20,
        "BizDate": 1600704000000,
        "DiscreteProperty": "type1",
        "Threshold": 0.2
      },
      {
        "Value": 23,
        "BizDate": 1600704000000,
        "DiscreteProperty": "type2"
      }
    ],
    "ReferenceValue": [
      {
        "SingleCheckResult": 2,
        "Value": 20,
        "BizDate": 1600704000000,
        "DiscreteProperty": "type1",
        "Threshold": 0.2
      },
      {
        "Value": 23,

```



```
    "BizDate": 1600704000000,  
    "DiscreteProperty": "type2"  
  }  
]  
}  
]  
}  
}
```

2.1.6.5.16. ListQualityResultsByEntity

Queries the historical monitoring results based on a partition filter expression.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListQualityResultsByEntity	The operation that you want to perform.
EndDate	String	Yes	2020-09-21 00:00:00	The end of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.
EntityId	Integer	Yes	152322134	The ID of the partition filter expression. You can call the GetQualityEntity operation to query the ID of the partition filter expression.
PageNumber	Integer	Yes	1	The number of the page to return.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectName	String	Yes	autotest	The name of the compute engine instance or connection. You can obtain the name from connection configurations.
StartDate	String	Yes	2020-09-20 00:00:00	The beginning of the time range to query. Specify the time in the yyyy-MM-dd HH:mm:ss format.

Parameter	Type	Required	Example	Description
RegionId	String	No	cn-hangzhou	The ID of the region where the service is activated. For example, the ID of the China (Shanghai) region is cn-shanghai, and that of the China (Zhangjiakou) region is cn-zhangjiakou. The system automatically determines the value of this parameter based on the endpoint used to call the operation.

Response parameters

Parameter	Type	Example	Description
RequestId	String	CBA58543-00D4-41****	The ID of the request.
Data	Struct	N/A	The top structure of returned monitoring results.
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
RuleChecks	Array of RuleChecks	N/A	The list of monitoring results.
ActualExpression	String	ds=20200912	The partition of the monitored table in the data source.
BeginTime	Long	1600704000000	The time when the monitoring started. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
BizDate	Long	1600704000000	The data timestamp. If the monitored business entity is offline data, the value indicates one day before the time when the monitoring is performed.

Parameter	Type	Example	Description
BlockType	Integer	0	The strength of the monitoring rule. The strength of monitoring rules indicates the importance of the rules. Valid values: • 1: The monitoring rule is a strong rule. • 0: The monitoring rule is a weak rule. You can specify whether a monitoring rule is a strong rule based on your business requirements. If a strong rule is used and an error alert is reported, nodes are blocked.
CheckResult	Integer	0	The monitoring result. Valid values: • 0: The data source table is normal. • 1: A warning alert is reported. • 2: An error alert is reported.
CheckResultStatus	Integer	0	The status of the monitoring result. Valid values: • 0: The data source table is normal. • 1: A warning alert is reported. • 2: An error alert is reported.
CheckerId	Integer	7	The ID of the checker.
CheckerName	String	fulx	The name of the checker.
CheckerType	Integer	0	The monitoring type. Valid values: • 1: VOLATILITY_CHECK • 2: FIXEDVALUE_CHECK
Comment	String	Rule description	The description of the monitoring rule.
CriticalThreshold	Float	0.5	The error threshold of the monitoring rule. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements. If a strong rule is used and an error alert is reported, nodes are blocked.

Parameter	Type	Example	Description
DateType	String	YMD	The scheduling type. In most cases, the value of this parameter is YMD. This value indicates year, month, and day.
DiscreteCheck	Boolean	true	Indicates whether the monitoring is discrete monitoring. Valid values: • true: The monitoring is discrete monitoring. • false: The monitoring is not discrete monitoring.
EndTime	Long	1600704000000	The end of the time range in which the data was queried. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 UTC.
EntityId	Integer	15432322	The ID of the partition filter expression.
ExpectValue	Float	200	The expected value of the monitoring rule.
ExternalId	String	1112323123	The ID of the node.
ExternalType	String	CWF2	The type of the scheduling system. Only CWF scheduling systems are supported.
FixedCheck	Boolean	false	Indicates whether the monitoring is performed based on a fixed value. Valid values: • true: The monitoring is performed based on a fixed value. • false: The monitoring is not performed based on a fixed value.
Id	Integer	121212121	The ID of the primary key.

Parameter	Type	Example	Description
IsPrediction	Boolean	true	Indicates whether the monitoring result is consistent with the predicted result. Valid values: - true: The monitoring result is consistent with the predicted result. - false: The monitoring result is inconsistent with the predicted result.
LowerValue	Float	2344	The lower limit of the predicted result. The value of this parameter is automatically generated based on the threshold that you specified.
MatchExpression	String	ds=\${yyyymmdd}	The partition filter expression.
MethodName	String	count_distinct	The method used to collect sample data, such as avg, count, sum, min, max, count_distinct, user_defined, table_count, table_size, table_dt_load_count, table_dt_refuseload_count, null_value, null_value/table_count, (table_count - count_distinct)/table_count, or table_count - count_distinct.
Op	String	>	The comparison operator of the monitoring rule.
ProjectName	String	autotest	The name of the compute engine instance or connection.
Property	String	type	The field that is monitored. This field is a column name of the monitored table in the data source.
ReferenceValue	Array of ReferenceValue	N/A	The historical sample values.
BizDate	String	2020-12-03	The data timestamp. If the monitored business entity is offline data, the value indicates one day before the time when the monitoring is performed.

Parameter	Type	Example	Description
DiscreteProperty	String	0	The values of the sample field that are grouped by using the GROUP BY clause. For example, the values of the Gender field are grouped by using the GROUP BY clause. In this case, the values of the DiscreteProperty parameter are Male, Female, and null.
SingleCheckResult	Integer	0	The monitoring result.
Threshold	Float	0.5	The threshold.
Value	Float	19	The check value.
ResultString	String	test	The string of the monitoring result.
RuleId	Integer	123123232	The ID of the monitoring rule.
RuleName	String	Rule name	The name of the monitoring rule.
SampleValue	Array of SampleValue	N/A	The sample values.
BizDate	String	2020-12-03	The data timestamp. If the monitored business entity is offline data, the value indicates one day before the time when the monitoring is performed.
DiscreteProperty	String	0	The values of the sample field that are grouped by using the GROUP BY clause. For example, the values of the Gender field are grouped by using the GROUP BY clause. In this case, the values of the DiscreteProperty parameter are Male, Female, and null.
Value	Float	19	The current sample value.
TableName	String	dual	The name of the table that is monitored.
TaskId	String	16008552981681a0d6****	The ID of the monitoring task.

Parameter	Type	Example	Description
TemplateId	Integer	5	The ID of the monitoring template.
TemplateName	String	Compare the Number of Unique Field Values Against Expectation	The name of the monitoring template.
TimeCost	String	202	The time that was taken to run the monitoring task. Unit: seconds.
Trend	String	abs	The trend of the monitoring result.
UpperValue	Float	25555	The upper limit of the predicted result. The value of this parameter is automatically generated based on the threshold that you specified.
WarningThreshold	Float	0.1	The warning threshold of the monitoring rule. The threshold indicates the deviation of monitoring results from expected values. You can customize this threshold based on your business requirements.
WhereCondition	String	id>0	The filter condition of the monitoring rule.
TotalCount	Long	50	The total number of entries returned.
ErrorCode	String	401	The error code returned.
ErrorMessage	String	You have no permission.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```

http(s)://[Endpoint]/?Action=ListQualityResultsByEntity
&EndDate=2020-09-21 00:00:00
&EntityId=152322134
&PageNumber=1
&PageSize=10
&ProjectName=autotest
&StartDate=2020-09-20 00:00:00
&<Common request parameters>

```

Sample success responses

XML format

```

<RequestId>CBA58543-00D4-41****</RequestId>
<Data>
  <TotalCount>50</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <RuleChecks>
    <Op>&gt;</Op>
    <TaskId>16008552981681a0d6****</TaskId>
    <EndTime>1600704000000</EndTime>
    <DiscreteCheck>true</DiscreteCheck>
    <CheckerName>fulx</CheckerName>
    <ExpectValue>200</ExpectValue>
    <ProjectName>autotest</ProjectName>
    <RuleId>123123232</RuleId>
    <WarningThreshold>0.1</WarningThreshold>
    <WhereCondition>id&gt;0</WhereCondition>
    <BeginTime>1600704000000</BeginTime>
    <CheckResultStatus>0</CheckResultStatus>
    <MatchExpression>ds=${yyyymmdd}</MatchExpression>
    <UpperValue>25555</UpperValue>
    <IsPrediction>true</IsPrediction>
    <CheckResult>0</CheckResult>
    <TemplateName>Compare the Number of Unique Field Values Against Expectation</TemplateName>
    <TimeCost>202s</TimeCost>
    <FixedCheck>false</FixedCheck>
    <MethodName>count_distinct</MethodName>
    <ResultString>test</ResultString>
    <Property>type</Property>
    <DateType>YMD</DateType>
    <BizDate>1600704000000</BizDate>
    <TemplateId>5</TemplateId>
    <RuleName>Rule name</RuleName>
    <Comment>Rule description</Comment>
    <CriticalThreshold>0.5</CriticalThreshold>
    <ExternalId>1112323123</ExternalId>
    <Trend>abs</Trend>
    <CheckerType>0</CheckerType>
    <ExternalType>CWF2</ExternalType>
    <CheckerId>7</CheckerId>
    <EntityId>15432322</EntityId>
    <TableName>dual</TableName>

```



```

    <BlockType>0</BlockType>
    <Id>121212121</Id>
    <ActualExpression>ds=20200912</ActualExpression>
    <LowerValue>232222</LowerValue>
  </RuleChecks>
  <RuleChecks>
    <SampleValue>
      <SingleCheckResult>0</SingleCheckResult>
      <Value>19</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>0</DiscreteProperty>
      <Threshold>0.5</Threshold>
    </SampleValue>
    <SampleValue>
      <Value>19</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>0</DiscreteProperty>
    </SampleValue>
    <ReferenceValue>
      <SingleCheckResult>0</SingleCheckResult>
      <Value>19</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>0</DiscreteProperty>
      <Threshold>0.5</Threshold>
    </ReferenceValue>
    <ReferenceValue>
      <Value>19</Value>
      <BizDate>1600704000000</BizDate>
      <DiscreteProperty>0</DiscreteProperty>
    </ReferenceValue>
  </RuleChecks>
</QualityResults>

```

JSON format

```

{
  "RequestId": "CBA58543-00D4-41****",
  "Data": {
    "TotalCount": 50,
    "PageSize": 10,
    "PageNumber": 1,
    "RuleChecks": [
      {
        "Op": "&gt;",
        "TaskId": "16008552981681a0d6****",
        "EndTime": 1600704000000,
        "DiscreteCheck": true,
        "CheckerName": "fulx",
        "ExpectValue": 200,
        "ProjectName": "autotest",
        "RuleId": 123123232,
        "WarningThreshold": 0.1,
        "WhereCondition": "id&gt;0",
        "BeginTime": 1600704000000
      }
    ]
  }
}

```

```

    beginTime : 1600704000000,
    "CheckResultStatus": 0,
    "MatchExpression": "ds=[yyyymmdd]",
    "UpperValue": 25555,
    "IsPrediction": true,
    "CheckResult": 0,
    "TemplateName": "Compare the Number of Unique Field Values Against Expectation",
    "TimeCost": "202s",
    "FixedCheck": false,
    "MethodName": "count_distinct",
    "ResultString": "test",
    "Property": "type",
    "DateType": "YMD",
    "BizDate": 1600704000000,
    "TemplateId": 5,
    "RuleName": "Rule name",
    "Comment": "Rule description",
    "CriticalThreshold": 0.5,
    "ExternalId": 1112323123,
    "Trend": "abs",
    "CheckerType": 0,
    "ExternalType": "CWF2",
    "CheckerId": 7,
    "EntityId": 15432322,
    "TableName": "dual",
    "BlockType": 0,
    "Id": 121212121,
    "ActualExpression": "ds=20200912",
    "LowerValue": 232222
  },
  {
    "SampleValue": [
      {
        "SingleCheckResult": 0,
        "Value": 19,
        "BizDate": 1600704000000,
        "DiscreteProperty": 0,
        "Threshold": 0.5
      },
      {
        "Value": 19,
        "BizDate": 1600704000000,
        "DiscreteProperty": 0
      }
    ],
    "ReferenceValue": [
      {
        "SingleCheckResult": 0,
        "Value": 19,
        "BizDate": 1600704000000,
        "DiscreteProperty": 0,
        "Threshold": 0.5
      },
      {
        "Value": 19

```

```
    "BizDate": 1600704000000,
    "DiscreteProperty": 0
  }
]
}
}
```

2.1.6.6. DataService Studio

2.1.6.6.1. CreateDataServiceApi

Creates an API in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDataServiceApi	The operation that you want to perform.
ApiDescription	String	Yes	Description of the test API	The description of the API.
ApiMode	Integer	Yes	0	The type of the API. Valid values: 0, 1, and 2. The value 0 indicates that the API is generated in wizard mode. The value 1 indicates that the API is generated in script mode. The value 2 indicates that the API is generated by registration.
ApiName	String	Yes	Name of the test API	The name of the API. The name must be 4 to 50 characters in length. It can contain letters, digits, and underscores (_) and must start with a letter.
ApiPath	String	Yes	/test/1	The path of the API. The path must be a maximum of 200 characters in length. It can contain letters, digits, underscores (_), and hyphens (-) and must start with a forward slash (/).
GroupId	String	Yes	ID of the business process	The ID of the business process.

Parameter	Type	Required	Example	Description
ProjectId	Long	Yes	10000	The ID of the workspace.
Protocols	String	Yes	0,1	The protocols used by the API. Valid values: 0 and 1. The value 0 indicates that the API uses HTTP. The value 1 indicates that the API uses HTTPS. Separate multiple protocols with commas (,).
RegionId	String	Yes	cn-shanghai	The region ID.
RequestMethod	Integer	Yes	0	The request method of the API. Valid values: 0, 1, 2, and 3. The value 0 indicates the GET method. The value 1 indicates the POST method. The value 2 indicates the PUT method. The value 3 indicates the DELETE method. APIs generated in wizard or script mode support GET and POST. APIs generated by registration support GET, POST, PUT, and DELETE.
ResponseContentType	Integer	Yes	0	The format in which the result of the API request is returned. Valid values: 0 and 1. The value 0 indicates the JSON format. The value 1 indicates the XML format. APIs generated in wizard or script mode support JSON. APIs generated by registration support JSON and XML.
TenantId	Long	Yes	10001	The tenant ID.
Timeout	Integer	Yes	10000	The timeout of the API request. Unit: milliseconds. Valid values: (0,30000].

Parameter	Type	Required	Example	Description
VisibleRange	Integer	Yes	0	The scope in which the API is visible. Valid values: 0 and 1. The value 0 indicates that the API is visible within the workspace. The value 1 indicates that the API is visible only to the API creator.
WizardDetails	String	No	<pre>{ "isPagedResponse": true, "wizardConnection": { "connectionId": 122786, "tableName": "person", "wizardRequestParameters": [{ "columnName": "id", "defaultValue": "", "exampleValue": "1", "isRequiredParameter": true, "parameterDataType": 2, "parameterDescription": "", "parameterName": "id", "parameterOperator": 0, "parameterPosition": 0 }], "wizardResponseParameters": [{ "columnName": "id", "exampleValue": "11", "parameterDataType": 2, "parameterDescription": "", "parameterName": "id" }] } }</pre>	The details of the API generated in wizard mode. You can query the details in the WizardDetails fields of the JSON sample success response for the GetDataServiceApi operation.

Parameter	Type	Required	Example	Description
ScriptDetails	String	No	<pre> {"isPagedResponse":false,"script": select id as id1, name as name1 from person where id=\${inputid}","scriptConnection": {"connectionId":1 22786,"tableName": "person"},"scriptRequestParameters": [{"defaultValue": ","exampleValue": ":"","isRequiredParameter":true,"parameterDataType": 0,"parameterDescription":"","parameterName":"inputid","parameterOperator":0,"parameterPosition": 1}], "scriptResponseParameters": [{"exampleValue": ":"","parameterDataType":0,"parameterDescription": ":"","parameterName":"id1"}, {"exampleValue": ":"","parameterDataType":0,"parameterDescription": ":"","parameterName":"name1"}]} </pre>	The details of the API generated in script mode. You can query the details in the ScriptDetails fields of the JSON sample success response for the GetDataServiceApi operation.

Parameter	Type	Required	Example	Description
RegistrationDetails	String	No	<pre>{ "failedResultSample": "fail", "registrationErrorCodes": [{ "errorCode": "e1", "errorMessage": "msg1", "errorSolution": "sol1" }, { "errorCode": "e2", "errorMessage": "msg2", "errorSolution": "sol2" }], "registrationRequestParameters": [{ "columnName": "", "defaultValue": "", "exampleValue": "", "isRequiredParameter": true, "parameterDataType": 0, "parameterDescription": "", "parameterName": "id", "parameterOperator": 0, "parameterPosition": 1 }, { "columnName": "", "defaultValue": "xx", "exampleValue": "", "isRequiredParameter": true, "parameterDataType": 0, "parameterDescription": "", "parameterName": "cc", "parameterOperator": 2, "parameterPosition": 2 }], "serviceContentType": 1, "serviceHost": "http://www.baidu.com", "servicePath": "/[uid]", "serviceRequestBodyDescription": "body", "successfulResultSample": "success" }</pre>	The details of the API generated by registration. You can query the details in the RegistrationDetails fields of the JSON sample success response for the GetDataServiceApi operation.

Parameter	Type	Required	Example	Description
FolderId	Long	No	0	The ID of the folder in which the API is stored. The ID of the root folder in a workflow is 0. The ID of the folder created by a user must be greater than 0.

Response parameters

Parameter	Type	Example	Description
Data	Long	100003	The ID of the created API.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```

http(s)://[Endpoint]/?Action=CreateDataServiceApi
&ApiDescription=Description of the test API
&ApiMode=0
&ApiName=Name of the test API
&ApiPath=/test/1
&GroupId=Group ID
&ProjectId=10000
&Protocols=0,1
&RegionId=cn-shanghai
&RequestMethod=0
&ResponseContentType=0
&TenantId=10001
&Timeout=10000
&VisibleRange=0
&<Common request parameters>

```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>100003</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": 100003,
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Activate DataWorks first.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

2.1.6.6.2. PublishDataServiceApi

Publishes an API in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	PublishDataServiceApi	The operation that you want to perform.
ApId	Long	Yes	10000	The ID of the API.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10002	The tenant ID.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the API is published.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=PublishDataServiceApi
&ApId=10000
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

2.1.6.6.3. DeleteDataServiceApi

Deletes an API from DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteDataServiceApi	The operation that you want to perform.
ApId	Long	Yes	10000	The ID of the API.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10002	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the API is deleted.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteDataServiceApi
&ApiId=10000
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6.4. ListDataServiceApis

Queries APIs in the development state in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServiceApis	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.

Parameter	Type	Required	Example	Description
ProjectId	Long	Yes	10000	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10001	The ID of the tenant.
ApiNameKeyword	String	No	My API name	The keyword in the API name. The keyword can be used to search for the API whose name contains the keyword.
ApiPathKeyword	String	No	/test/	The keyword in the path of the API. The keyword can be used to search for the API whose path contains the keyword.
CreatorId	String	No	12345	The ID of the Apsara Stack tenant account used by the creator of the API. The ID can be used to search for the API created by the creator.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The information about APIs in the development state.
Apis	Array of Api	N/A	The list of APIs in the development state.
ApiId	Long	10002	The ID of the API.
ApiMode	Integer	0	The type of the API. Valid values: 0, 1, and 2. A value of 0 indicates that the API is generated in wizard mode. A value of 1 indicates that the API is generated in script mode. A value of 2 indicates that the API is generated by registration.
ApiName	String	My API name	The name of the API.

Parameter	Type	Example	Description
ApiPath	String	/test/1	The path of the API.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the API was created.
CreatorId	String	1234567	The ID of the Apsara Stack tenant account used by the creator of the API.
Description	String	API description	The description of the API.
FolderId	Long	0	The ID of the folder in which the API is stored.
GroupId	String	abcde123456789	The ID of the API group.
ModifiedTime	String	2020-06-23T00:21:01+0800	The time when the API was last modified.
OperatorId	String	2345678	The ID of the Apsara Stack tenant account used by the user who last modified the API.
ProjectId	Long	10000	The ID of the workspace.
Protocols	List	0	The protocols used by the API. Valid values: 0 and 1. A value of 0 indicates that the API uses HTTP. A value of 1 indicates that the API uses HTTPS. Multiple protocols are presented in a list.
RegistrationDetails	Struct	N/A	The details of the API generated by registration. This parameter is returned only if the API is generated by registration.
FailedResultSample	String	null	The sample error response of the API.
RegistrationErrorCodes	Array of RegistrationErrorcode	N/A	The error codes returned for the API generated by registration.
ErrorCode	String	1001	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	fail to call	The error message returned.
ErrorSolution	String	retry	The solution used to resolve the issue.
RegistrationRequestParameters	Array of RegistrationRequestParameter	N/A	The request parameters of the API generated by registration.
ColumnName	String	column1	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	name1	The name of the parameter.

Parameter	Type	Example	Description
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the operator is Equal. A value of 1 indicates that the operator is Like. A value of 2 indicates that the operator is Const. A value of 3 indicates that the operator is In. APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the parameter is in the URL path of the request. A value of 1 indicates that the parameter is in the Query parameter of the request URL. A value of 2 indicates that the parameter is in the request header. A value of 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
ServiceContentType	Integer	0	The format in which the response of the API request is returned. Valid values: 0 and 1. A value of 0 indicates the JSON format. A value of 1 indicates the XML format. APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
ServiceHost	String	http://www.abc.com	The URL of the backend service.
ServicePath	String	/index	The path of the backend service.
ServiceRequestBodyDescription	String	null	The body of the request initiated to call the backend service.

Parameter	Type	Example	Description
SuccessfulResultSample	String	null	The sample success response of the API.
RequestMethod	Integer	0	The request method of the API. Valid values: 0, 1, 2, and 3. A value of 0 indicates the GET method. A value of 1 indicates the POST method. A value of 2 indicates the PUT method. A value of 3 indicates the DELETE method. APIs generated in wizard or script mode support the GET and POST methods. APIs generated by registration support the GET, POST, PUT, and DELETE methods.
ResponseContentType	Integer	0	The format in which the response of the API request is returned. Valid values: 0 and 1. A value of 0 indicates the JSON format. A value of 1 indicates the XML format. APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
ScriptDetails	Struct	N/A	The details of the API generated in script mode. This parameter is returned only if the API is generated in script mode.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
Script	String	select a from t	The SQL script.
ScriptConnection	Struct	N/A	The connection information of the API generated in script mode.
ConnectionId	Long	123	The ID of the connection.
TableName	String	t	The name of the table in the data source.
ScriptRequestParameters	Array of ScriptRequestParameter	N/A	The request parameters of the API generated in script mode.

Parameter	Type	Example	Description
ColumnName	String	column1	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: • 0: STRING • 1: INT • 2: LONG • 3: FLOAT • 4: DOUBLE • 5: BOOLEAN
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: • 0: Equal • 1: Like • 2: Const • 3: In APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: • 0: Path • 1: Query • 2: Head • 3: Body APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
ScriptResponseParameters	Array of ScriptResponseParameter	N/A	The response parameters of the API generated in script mode.
ColumnName	String	column2	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
ExampleValue	String	example2	The sample value of the parameter.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: • 0: STRING • 1: INT • 2: LONG • 3: FLOAT • 4: DOUBLE • 5: BOOLEAN
ParameterDescription	String	description2	The description of the parameter.
ParameterName	String	param2	The name of the parameter.
Status	Integer	0	The status of the API. Valid values: 0 and 1. A value of 0 indicates that the API is not published. A value of 1 indicates that the API is published.

Parameter	Type	Example	Description
TenantId	Long	10000	The ID of the tenant.
Timeout	Integer	10000	The timeout period of the API request. Unit: milliseconds.
VisibleRange	Integer	0	The scope in which the API is visible. Valid values: 0 and 1. A value of 0 indicates that the API is visible within the workspace. A value of 1 indicates that the API is visible only to the API creator.
WizardDetails	Struct	N/A	The details of the API generated in wizard mode. This parameter is returned only if the API is generated in wizard mode.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
WizardConnection	Struct	N/A	The connection information of the API generated in wizard mode.
ConnectionId	Long	123	The ID of the connection.
TableName	String	t	The name of the table in the data source.
WizardRequestParameters	Array of WizardRequestParameter	N/A	The request parameters of the API generated in wizard mode.
ColumnName	String	column1	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.

Parameter	Type	Example	Description
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the operator is Equal. A value of 1 indicates that the operator is Like. A value of 2 indicates that the operator is Const. A value of 3 indicates that the operator is In. APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the parameter is in the URL path of the request. A value of 1 indicates that the parameter is in the Query parameter of the request URL. A value of 2 indicates that the parameter is in the request header. A value of 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
WizardResponseParameters	Array of WizardResponseParameter	N/A	The response parameters of the API generated in wizard mode.

Parameter	Type	Example	Description
ColumnName	String	column2	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
ExampleValue	String	example2	The sample value of the parameter.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description2	The description of the parameter.
ParameterName	String	param2	The name of the parameter.
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	100	The total number of entries returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

The details about the APIs of different types are returned in different parameters. The details about the API generated in wizard mode are returned in the WizardDetails parameter. The details about the API generated in script mode are returned in the ScriptDetails parameter. The details about the API generated by registration are returned in the RegistrationDetails parameter.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServiceApis
&PageNumber=1
&PageSize=10
&ProjectId=10000
&RegionId=cn-shanghai
&TenantId=10001
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Apis>
    <Status>0</Status>
    <VisibleRange>0</VisibleRange>
    <CreatedTime>2020-06-23T00:21:01+0800</CreatedTime>
    <Description>API description</Description>
    <ApiName>My API name</ApiName>
    <CreatorId>1234567</CreatorId>
    <RequestMethod>0</RequestMethod>
    <Timeout>10000</Timeout>
    <ProjectId>10000</ProjectId>
    <ResponseContentType>0</ResponseContentType>
    <OperatorId>2345678</OperatorId>
    <GroupId>abcde123456789</GroupId>
    <TenantId>10000</TenantId>
    <ModifiedTime>2020-06-23T00:21:01+0800</ModifiedTime>
    <ApiId>10002</ApiId>
    <ApiMode>0</ApiMode>
    <ApiPath>/test/1</ApiPath>
  </Apis>
  <Apis>
    <Protocols>0</Protocols>
  </Apis>
  <Apis>
    <WizardDetails>
      <Script>select a from t</Script>
      <ServiceContentType>0</ServiceContentType>
      <ScriptRequestParameters>
        <ColumnName>column1</ColumnName>
```

```

    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptRequestParameters>
  <ServicePath>/index</ServicePath>
  <RegistrationErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationErrorCodes>
  <RegistrationErrorCodes>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </RegistrationErrorCodes>
  <WizardResponseParameters>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </WizardResponseParameters>
  <WizardResponseParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </WizardResponseParameters>
  <ServiceHost>http://www.abc.com</ServiceHost>
  <RegistrationRequestParameters>
    <ErrorCode>1001</ErrorCode>

```

```
</ErrorMessage>fail to call</ErrorMessage>
<ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
```

```

    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
    <TableName>t</TableName>
    <ConnectionId>123</ConnectionId>
</ScriptConnection>
</WizardDetails>
<ScriptDetails>
    <Script>select a from t</Script>
    <ServiceContentType>0</ServiceContentType>
    <ScriptRequestParameters>
        <ColumnName>column1</ColumnName>
        <DefaultValue>default1</DefaultValue>
        <ExampleValue>example1</ExampleValue>
        <ParameterDataType>0</ParameterDataType>
        <ParameterPosition>0</ParameterPosition>
        <ParameterName>param1</ParameterName>
        <ParameterOperator>0</ParameterOperator>
        <ParameterDescription>description1</ParameterDescription>
        <IsRequiredParameter>true</IsRequiredParameter>
    </ScriptRequestParameters>
    <ScriptRequestParameters>
        <ColumnName>column2</ColumnName>
        <ExampleValue>example2</ExampleValue>
        <ParameterDataType>0</ParameterDataType>
        <ParameterName>param2</ParameterName>
        <ParameterDescription>description2</ParameterDescription>
    </ScriptRequestParameters>
    <ServicePath>/index</ServicePath>
    <RegistrationErrorCodes>
        <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
        <ErrorMessage>The connection does not exist.</ErrorMessage>
        <ErrorSolution>retry</ErrorSolution>
    </RegistrationErrorCodes>
    <RegistrationErrorCodes>
        <ColumnName>column1</ColumnName>
        <DefaultValue>default1</DefaultValue>
        <ExampleValue>example1</ExampleValue>
        <ParameterDataType>0</ParameterDataType>
        <ParameterPosition>0</ParameterPosition>
        <ParameterName>name1</ParameterName>
        <ParameterOperator>0</ParameterOperator>
        <ParameterDescription>description1</ParameterDescription>
        <IsRequiredParameter>true</IsRequiredParameter>
    </RegistrationErrorCodes>
    <WizardResponseParameters>
        <ColumnName>column1</ColumnName>
        <DefaultValue>default1</DefaultValue>
        <ExampleValue>example1</ExampleValue>
        <ParameterDataType>0</ParameterDataType>
        <ParameterPosition>0</ParameterPosition>
        <ParameterName>param1</ParameterName>

```

```
<ParameterOperator>0</ParameterOperator>
<ParameterDescription>description1</ParameterDescription>
<IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
```

```

<ScriptResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</ScriptDetails>
<RegistrationDetails>
  <Script>select a from t</Script>
  <ServiceContentType>0</ServiceContentType>
  <ScriptRequestParameters>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptRequestParameters>
  <ServicePath>/index</ServicePath>
  <RegistrationErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationErrorCodes>
  <RegistrationErrorCodes>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
  </RegistrationErrorCodes>

```

```
<ParameterDataType>0</ParameterDataType>
<ParameterPosition>0</ParameterPosition>
<ParameterName>name1</ParameterName>
<ParameterOperator>0</ParameterOperator>
<ParameterDescription>description1</ParameterDescription>
<IsRequiredParameter>true</IsRequiredParameter>
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
```

```

</WizardRequestParameters>
<WizardRequestParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</RegistrationDetails>
</Apis>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>Normal</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "Apis": [
      {

```



```
"Status": 0,
"VisibleRange": 0,
"CreatedTime": "2020-06-23T00:21:01+0800",
"Description": "API description",
"ApiName": "My API name",
"CreatorId": 1234567,
"RequestMethod": 0,
"Timeout": 10000,
"ProjectId": 10000,
"ResponseContentType": 0,
"OperatorId": 2345678,
"GroupId": "abcde123456789",
"TenantId": 10000,
"ModifiedTime": "2020-06-23T00:21:01+0800",
"ApId": 10002,
"ApiMode": 0,
"ApiPath": "/test/1"
},
{
  "Protocols": 0
},
{
  "WizardDetails": {
    "Script": "select a from t",
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
      {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
      },
      {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
      }
    ],
    "ServicePath": "/index",
    "RegistrationErrorCodes": [
      {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
      },
      {
        "ColumnName": "column1",
        "DefaultValue": "default1"
```

```

        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
],
"WizardResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
    {
        "ErrorCode": 1001,
        "ErrorMessage": "fail to call",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":\"1\"}",
"WizardRequestParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1".

```

```
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"ScriptDetails": {
    "Script": "select a from t",
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
        {
            "ColumnName": "column1",
            "DefaultValue": "default1",
```

```

        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServicePath": "/index",
"RegistrationErrorCodes": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"WizardResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
]

```

```
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "ColumnName": "column1",
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":\"1\"}",
"WizardRequestParameters": [
  {
    "ColumnName": "column1",
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ColumnName": "column2",
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
  "TableName": "t",
  "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
  {
    "ColumnName": "column1",
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
```

```

        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
},
"RegistrationDetails": {
    "Script": "select a from t",
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
        {
            "ColumnName": "column1",
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0,
            "ParameterName": "param1",
            "ParameterOperator": 0,
            "ParameterDescription": "description1",
            "IsRequiredParameter": true
        },
        {
            "ColumnName": "column2",
            "ExampleValue": "example2",
            "ParameterDataType": 0,
            "ParameterName": "param2",
            "ParameterDescription": "description2"
        }
    ],
    "ServicePath": "/index",
    "RegistrationErrorCodes": [
        {
            "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
            "ErrorMessage": "The connection does not exist.",
            "ErrorSolution": "retry"
        },
        {
            "ColumnName": "column1",
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0
        }
    ]
}

```

```
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"WizardResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":\"1\"}",
"WizardRequestParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
```

```

        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
}
]
},
"ErrorCode": 0,
"ErrorMessage": "Normal",
"Success": true
}

```


Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6.5. GetDataServiceApi

Queries the details of an API in the development state in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDataServiceApi	The operation that you want to perform.
ApiId	Long	Yes	10000	The ID of the API.
ProjectId	Long	Yes	10001	The ID of the workspace.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10002	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The returned data.
ApiId	Long	10000	The ID of the API.
ApiMode	Integer	0	The type of the API. Valid values: 0, 1, and 2. A value of 0 indicates that the API is generated in wizard mode. A value of 1 indicates that the API is generated in script mode. A value of 2 indicates that the API is generated by registration.
ApiName	String	API name	The name of the API.
ApiPath	String	/test/1	The path of the API.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the API was created.
CreatorId	String	1234567	The ID of the Apsara Stack tenant account used by the creator of the API.
Description	String	API description	The description of the API.
FolderId	Long	0	The ID of the folder in which the API is stored.
GroupId	String	ab123****	The ID of the API group.
ModifiedTime	String	2020-06-23T00:21:01+0800	The time when the API was last modified.

Parameter	Type	Example	Description
OperatorId	String	2345678	The ID of the Apsara Stack tenant account used by the user who last modified the API.
ProjectId	Long	10001	The ID of the workspace.
Protocols	List	0	The protocols used by the API. Valid values: 0 and 1. A value of 0 indicates that the API uses HTTP. A value of 1 indicates that the API uses HTTPS. Multiple protocols are presented in a list.
RegistrationDetails	Struct	N/A	The details of the API generated by registration. This parameter is returned only if the API is generated by registration.
FailedResultSample	String	null	The sample error response of the API.
RegistrationErrorCodes	Array of RegistrationErrorCode	N/A	The error codes returned for the API generated by registration.
ErrorCode	String	1001	The error code returned.
ErrorMessage	String	fail to call	The error message returned.
ErrorSolution	String	retry	The solution used to resolve the issue.
RegistrationRequestParameters	Array of RegistrationRequestParameter	N/A	The request parameters of the API generated by registration.
ColumnName	String	column1	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
DefaultValue	String	default 1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.

Parameter	Type	Example	Description
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	name1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: • 0: Equal • 1: Like • 2: Const • 3: In APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: • 0: Path • 1: Query • 2: Head • 3: Body APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
ServiceContentType	Integer	0	The format in which the response of the API request is returned. Valid values: • 0: JSON • 1: XML APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
ServiceHost	String	http://www.abc.com	The URL of the backend service.
ServicePath	String	/index	The path of the backend service.
ServiceRequestBodyDescription	String	null	The body of the request initiated to call the backend service.
SuccessfulResultSample	String	null	The sample success response of the API.

Parameter	Type	Example	Description
RequestMethod	Integer	0	The request method of the API. Valid values: 0, 1, 2, and 3. A value of 0 indicates the GET method. A value of 1 indicates the POST method. A value of 2 indicates the PUT method. A value of 3 indicates the DELETE method. APIs generated in wizard or script mode support the GET and POST methods. APIs generated by registration support the GET, POST, PUT, and DELETE methods.
ResponseContentType	Integer	0	The format in which the response of the API request is returned. Valid values: 0 and 1. A value of 0 indicates the JSON format. A value of 1 indicates the XML format. APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
ScriptDetails	Struct	N/A	The details of the API generated in script mode. This parameter is returned only if the API is generated in script mode.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
Script	String	select a from t	The SQL script.
ScriptConnection	Struct	N/A	The connection information of the API generated in script mode.
ConnectionId	Long	123	The ID of the connection.
TableName	String	t	The name of the table in the data source.
ScriptRequestParameters	Array of ScriptRequestParameter	N/A	The request parameters of the API generated in script mode.
ColumnName	String	column1	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.

Parameter	Type	Example	Description
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the operator is Equal. A value of 1 indicates that the operator is Like. A value of 2 indicates that the operator is Const. A value of 3 indicates that the operator is In. APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the parameter is in the URL path of the request. A value of 1 indicates that the parameter is in the Query parameter of the request URL. A value of 2 indicates that the parameter is in the request header. A value of 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
ScriptResponseParameters	Array of ScriptResponseParameter	N/A	The response parameters of the API generated in script mode.
ColumnName	String	column2	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
ExampleValue	String	example2	The sample value of the parameter.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description2	The description of the parameter.

Parameter	Type	Example	Description
ParameterName	String	param2	The name of the parameter.
Status	Integer	0	The status of the API. Valid values: 0 and 1. A value of 0 indicates that the API is not published. A value of 1 indicates that the API is published.
TenantId	Long	10002	The ID of the tenant.
Timeout	Integer	10000	The timeout period of the API request. Unit: milliseconds.
VisibleRange	Integer	0	The scope in which the API is visible. Valid values: 0 and 1. A value of 0 indicates that the API is visible within the workspace. A value of 1 indicates that the API is visible only to the API creator.
WizardDetails	Struct	N/A	The details of the API generated in wizard mode. This parameter is returned only if the API is generated in wizard mode.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
WizardConnection	Struct	N/A	The connection information of the API generated in wizard mode.
ConnectionId	Long	123	The ID of the connection.
TableName	String	t	The name of the table in the data source.
WizardRequestParameters	Array of WizardRequestParameter	N/A	The request parameters of the API generated in wizard mode.

Parameter	Type	Example	Description
ColumnName	String	column1	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the operator is Equal. A value of 1 indicates that the operator is Like. A value of 2 indicates that the operator is Const. A value of 3 indicates that the operator is In. APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the parameter is in the URL path of the request. A value of 1 indicates that the parameter is in the Query parameter of the request URL. A value of 2 indicates that the parameter is in the request header. A value of 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
WizardResponseParameters	Array of WizardResponseParameter	N/A	The response parameters of the API generated in wizard mode.
ColumnName	String	column2	The name of the associated field. This parameter is supported only if the API is generated in wizard mode.
ExampleValue	String	example2	The sample value of the parameter.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description2	The description of the parameter.

Parameter	Type	Example	Description
ParameterName	String	param2	The name of the parameter.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

The details about the APIs of different types are returned in different parameters. The details about the API generated in wizard mode are returned in the WizardDetails parameter. The details about the API generated in script mode are returned in the ScriptDetails parameter. The details about the API generated by registration are returned in the RegistrationDetails parameter.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDataServiceApi
&ApId=10000
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Status>0</Status>
  <VisibleRange>0</VisibleRange>
  <Protocols>0</Protocols>
  <WizardDetails>
    <Script>select a from t</Script>
    <ServiceContentType>0</ServiceContentType>
    <ScriptRequestParameters>
      <ColumnName>column1</ColumnName>
      <DefaultValue>default1</DefaultValue>
      <ExampleValue>example1</ExampleValue>
```

```
<ParameterDataType>0</ParameterDataType>
<ParameterPosition>0</ParameterPosition>
<ParameterName>param1</ParameterName>
<ParameterOperator>0</ParameterOperator>
<ParameterDescription>description1</ParameterDescription>
<IsRequiredParameter>true</IsRequiredParameter>
</ScriptRequestParameters>
<ScriptRequestParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptRequestParameters>
<ServicePath>/index</ServicePath>
<RegistrationErrorCodes>
  <ErrorCode>1001</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationErrorCodes>
<RegistrationErrorCodes>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>1001</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
```

```

</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>

```

```
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</WizardDetails>
<CreatedTime>2020-06-23T00:21:01+0800</CreatedTime>
<Description>API description</Description>
<ApiName>API name</ApiName>
<CreatorId>1234567</CreatorId>
<RequestMethod>0</RequestMethod>
<Timeout>10000</Timeout>
<ProjectId>10001</ProjectId>
<ResponseContentType>0</ResponseContentType>
<OperatorId>2345678</OperatorId>
<RegistrationDetails>
  <Script>select a from t</Script>
  <ServiceContentType>0</ServiceContentType>
  <ScriptRequestParameters>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptRequestParameters>
  <ServicePath>/index</ServicePath>
  <RegistrationErrorCodes>
    <ErrorCode>1001</ErrorCode>
    <ErrorMessage>fail to call</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationErrorCodes>
  <RegistrationErrorCodes>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </RegistrationErrorCodes>
</WizardResponseParameters>
```

```

    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </WizardResponseParameters>
  <WizardResponseParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </WizardResponseParameters>
  <ServiceHost>http://www.abc.com</ServiceHost>
  <RegistrationRequestParameters>
    <ErrorCode>1001</ErrorCode>
    <ErrorMessage>fail to call</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationRequestParameters>
  <RegistrationRequestParameters>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </RegistrationRequestParameters>
  <IsPagedResponse>true</IsPagedResponse>
  <ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
  <WizardRequestParameters>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </WizardRequestParameters>
  <WizardRequestParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </WizardRequestParameters>

```



```
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</RegistrationDetails>
<GroupId>abcde123456789</GroupId>
<TenantId>10002</TenantId>
<ModifiedTime>2020-06-23T00:21:01+0800</ModifiedTime>
<ScriptDetails>
  <Script>select a from t</Script>
  <ServiceContentType>0</ServiceContentType>
  <ScriptRequestParameters>
    <ColumnName>column1</ColumnName>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <ColumnName>column2</ColumnName>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptRequestParameters>
  <ServicePath>/index</ServicePath>
```

```

<RegistrationErrorCodes>
  <ErrorCode>1001</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationErrorCodes>
<RegistrationErrorCodes>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>1001</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ColumnName>column1</ColumnName>

```

```
<ColumnName>column1</ColumnName>
<DefaultValue>default1</DefaultValue>
<ExampleValue>example1</ExampleValue>
<ParameterDataType>0</ParameterDataType>
<ParameterPosition>0</ParameterPosition>
<ParameterName>param1</ParameterName>
<ParameterOperator>0</ParameterOperator>
<ParameterDescription>description1</ParameterDescription>
<IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ColumnName>column1</ColumnName>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ColumnName>column2</ColumnName>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</ScriptDetails>
<ApiId>10000</ApiId>
<ApiMode>0</ApiMode>
<ApiPath>/test/1</ApiPath>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "Status": 0,
    "VisibleRange": 0,
    "Protocols": 0,
    "WizardDetails": {
      "Script": "select a from t",
      "ServiceContentType": 0,
      "ScriptRequestParameters": [
        {
          "ColumnName": "column1",
          "DefaultValue": "default1",
          "ExampleValue": "example1",
          "ParameterDataType": 0,
          "ParameterPosition": 0,
          "ParameterName": "param1",
          "ParameterOperator": 0,
          "ParameterDescription": "description1",
          "IsRequiredParameter": true
        },
        {
          "ColumnName": "column2",
          "ExampleValue": "example2",
          "ParameterDataType": 0,
          "ParameterName": "param2",
          "ParameterDescription": "description2"
        }
      ],
      "ServicePath": "/index",
      "RegistrationErrorCodes": [
        {
          "ErrorCode": 1001,
          "ErrorMessage": "fail to call",
          "ErrorSolution": "retry"
        },
        {
          "ColumnName": "column1",
          "DefaultValue": "default1",
          "ExampleValue": "example1",
          "ParameterDataType": 0,
          "ParameterPosition": 0,
          "ParameterName": "name1",
          "ParameterOperator": 0,
          "ParameterDescription": "description1",
          "IsRequiredParameter": true
        }
      ],
      "WizardResponseParameters": [
        {
          "ColumnName": "column1",

```

```
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
    {
        "ErrorCode": 1001,
        "ErrorMessage": "fail to call",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":1}",
"WizardRequestParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2"
```

```

        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
},
"CreatedTime": "2020-06-23T00:21:01+0800",
"Description": "API description",
"ApiName": "API name",
"CreatorId": 1234567,
"RequestMethod": 0,
"Timeout": 10000,
"ProjectId": 10001,
"ResponseContentType": 0,
"OperatorId": 2345678,
"RegistrationDetails": {
    "Script": "select a from t",
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
        {
            "ColumnName": "column1",
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0,
            "ParameterName": "param1"

```

```
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServicePath": "/index",
"RegistrationErrorCodes": [
    {
        "ErrorCode": 1001,
        "ErrorMessage": "fail to call",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"WizardResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
    {
```

```

        "ErrorCode": 1001,
        "ErrorMessage": "fail to call",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":1}",
"WizardRequestParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",

```



```
        "IsRequiredParameter": true
      },
      {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
      }
    ],
    "ScriptConnection": {
      "TableName": "t",
      "ConnectionId": 123
    }
  },
  "GroupId": "abcde123456789",
  "TenantId": 10002,
  "ModifiedTime": "2020-06-23T00:21:01+0800",
  "ScriptDetails": {
    "Script": "select a from t",
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
      {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
      },
      {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
      }
    ]
  },
  "ServicePath": "/index",
  "RegistrationErrorCodes": [
    {
      "ErrorCode": 1001,
      "ErrorMessage": "fail to call",
      "ErrorSolution": "retry"
    },
    {
      "ColumnName": "column1",
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
```

```

        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"WizardResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
    {
        "ErrorCode": 1001,
        "ErrorMessage": "fail to call",
        "ErrorSolution": "retry"
    },
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":1}",
"WizardRequestParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
]

```

```
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ColumnName": "column1",
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ColumnName": "column2",
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
},
"Apild": 10000,
"ApiMode": 0,
"ApiPath": "/test/1"
},
"ErrorCode": 1031203110005,
"ErrorMessage": "The specified parameters are invalid.",
"Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6. UpdateDataServiceApi

Updates an API in the development state in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	UpdateDataServiceApi	The operation that you want to perform.
ApiDescription	String	Yes	API description	The description of the API.
ApiId	Long	Yes	10000	The ID of the API.

Parameter	Type	Required	Example	Description
ApiPath	String	Yes	/test/2	The path of the API.
ProjectId	Long	Yes	10001	The ID of the workspace.
Protocols	String	Yes	0,1	The protocols used by the API. Valid values: 0 and 1. A value of 0 indicates that the API uses HTTP. A value of 1 indicates that the API uses HTTPS. You can specify multiple protocols. Separate them with commas (,).
RegionId	String	Yes	cn-shanghai	The ID of the region.
RequestMethod	Integer	Yes	0	The request method of the API. Valid values: 0, 1, 2, and 3. A value of 0 indicates the GET method. A value of 1 indicates the POST method. A value of 2 indicates the PUT method. A value of 3 indicates the DELETE method. APIs generated in wizard or script mode support the GET and POST methods. APIs generated by registration support the GET, POST, PUT, and DELETE methods.
ResponseContentType	Integer	Yes	0	The format in which the response of the API request is returned. Valid values: 0 and 1. A value of 0 indicates the JSON format. A value of 1 indicates the XML format. APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
TenantId	Long	Yes	10001	The ID of the tenant.
Timeout	Integer	Yes	10000	The timeout period of the API request. Unit: milliseconds. Valid values: (0,30000].

Parameter	Type	Required	Example	Description
VisibleRange	Integer	Yes	0	The scope in which the API is visible. Valid values: 0 and 1. A value of 0 indicates that the API is visible within the workspace. A value of 1 indicates that the API is visible only to the API creator.
WizardDetails	String	No	null	The details of the API generated in wizard mode. For more information, see the WizardDetails parameter returned by the GetDataServiceApi operation.
ScriptDetails	String	No	null	The details of the API generated in script mode. For more information, see the scriptDetails parameter returned by the GetDataServiceApi operation.
RegistrationDetails	String	No	null	The details of the API generated by registration. For more information, see the registrationDetails parameter returned by the GetDataServiceApi operation.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the API is updated.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=UpdateDataServiceApi
&ApiDescription=API description
&Apild=10000
&ApiPath=/test/2
&ProjectId=10001
&Protocols=0,1
&RegionId=cn-shanghai
&RequestMethod=0
&ResponseContentType=0
&TenantId=10001
&Timeout=10000
&VisibleRange=0
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Activate DataWorks first.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

2.1.6.6.7. GetDataServicePublishedApi

Queries the details of an API in the published state in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDataServicePublishedApi	The operation that you want to perform.
ApiId	Long	Yes	10001	The ID of the API.
ProjectId	Long	Yes	10002	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10003	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description

Parameter	Type	Example	Description
Data	Struct	N/A	The latest information of the API in the published state.
ApiId	Long	10000	The ID of the API.
ApiMode	Integer	0	The type of the API. Valid values: 0, 1, and 2. A value of 0 indicates that the API is generated in wizard mode. A value of 1 indicates that the API is generated in script mode. A value of 2 indicates that the API is generated by registration.
ApiName	String	API name	The name of the API.
ApiPath	String	/test/1	The path of the API.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the API was created.
CreatorId	String	1234567	The ID of the Apsara Stack tenant account used by the creator of the API.
Description	String	API description	The description of the API.
GroupId	String	ab12**	The ID of the API group.
ModifiedTime	String	2020-06-23T00:21:01+0800	The time when the API was last modified.
OperatorId	String	2345678	The ID of the Apsara Stack tenant account used by the user who last modified the API.
ProjectId	Long	10001	The ID of the workspace.
Protocols	List	0	The protocols used by the API. Valid values: 0 and 1. A value of 0 indicates that the API uses HTTP. A value of 1 indicates that the API uses HTTPS. Multiple protocols are presented in a list.

Parameter	Type	Example	Description
RegistrationDetails	Struct	N/A	The details of the API generated by registration. This parameter is returned only if the API is generated by registration.
FailedResultSample	String	null	The sample error response of the API.
RegistrationErrorCodes	Array of RegistrationErrorCode	N/A	The error codes returned for the API generated by registration.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
ErrorSolution	String	retry	The solution used to resolve the issue.
RegistrationRequestParameters	Array of RegistrationRequestParameter	N/A	The request parameters of the API generated by registration.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	name1	The name of the parameter.

Parameter	Type	Example	Description
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the operator is Equal. A value of 1 indicates that the operator is Like. A value of 2 indicates that the operator is Const. A value of 3 indicates that the operator is In. APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the parameter is in the URL path of the request. A value of 1 indicates that the parameter is in the Query parameter of the request URL. A value of 2 indicates that the parameter is in the request header. A value of 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
ServiceContentType	Integer	0	The format in which the response of the API request is returned. Valid values: 0 and 1. A value of 0 indicates the JSON format. A value of 1 indicates the XML format. APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
ServiceHost	String	http://www.abc.com	The URL of the backend service.
ServicePath	String	/index	The path of the backend service.
ServiceRequestBodyDescription	String	null	The body of the request initiated to call the backend service.

Parameter	Type	Example	Description
SuccessfulResultSample	String	null	The sample success response of the API.
RequestMethod	Integer	0	The request method of the API. Valid values: 0, 1, 2, and 3. A value of 0 indicates the GET method. A value of 1 indicates the POST method. A value of 2 indicates the PUT method. A value of 3 indicates the DELETE method. APIs generated in wizard or script mode support the GET and POST methods. APIs generated by registration support the GET, POST, PUT, and DELETE methods.
ResponseContentType	Integer	0	The format in which the response of the API request is returned. Valid values: 0 and 1. A value of 0 indicates the JSON format. A value of 1 indicates the XML format. APIs generated in wizard or script mode support the JSON format. APIs generated by registration support the JSON and XML formats.
ScriptDetails	Struct	N/A	The details of the API generated in script mode. This parameter is returned only if the API is generated in script mode.
FailedResultSample	String	null	The sample error response of the API.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
Script	String	select a from t	The SQL script.
ScriptConnection	Struct	N/A	The connection information of the API generated in script mode.
ConnectionId	Long	123	The ID of the connection.
TableName	String	t	The name of the table in the data source.
ScriptErrorCodes	Array of ScriptErrorCodes	N/A	The error codes returned for the API generated in script mode.

Parameter	Type	Example	Description
ErrorCode	String	1001	The error code returned.
ErrorMessage	String	fail to call	The error message returned.
ErrorSolution	String	retry	The solution used to resolve the issue.
ScriptRequestParameters	Array of ScriptRequestParameter	N/A	The request parameters of the API generated in script mode.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: • 0: STRING • 1: INT • 2: LONG • 3: FLOAT • 4: DOUBLE • 5: BOOLEAN
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.

Parameter	Type	Example	Description
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: • 0: Equal • 1: Like • 2: Const • 3: In APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.
ParameterPosition	Integer	0	The position of the parameter. Valid values: • 0: Path • 1: Query • 2: Head • 3: Body APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
ScriptResponseParameters	Array of ScriptResponseParameter	N/A	The response parameters of the API generated in script mode.
ExampleValue	String	example2	The sample value of the parameter.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description2	The description of the parameter.

Parameter	Type	Example	Description
ParameterName	String	param2	The name of the parameter.
SuccessfulResultSample	String	null	The sample success response of the API.
Status	Integer	1	The status of the API. Valid values: 0 and 1. A value of 0 indicates that the API is not published. A value of 1 indicates that the API is published.
TenantId	Long	10001	The ID of the tenant.
Timeout	Integer	10000	The timeout period of the API request. Unit: milliseconds.
VisibleRange	Integer	0	The scope in which the API is visible. Valid values: 0 and 1. A value of 0 indicates that the API is visible within the workspace. A value of 1 indicates that the API is visible only to the API creator.
WizardDetails	Struct	N/A	The details of the API generated in wizard mode. This parameter is returned only if the API is generated in wizard mode.
FailedResultSample	String	null	The sample error response of the API.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
SuccessfulResultSample	String	null	The sample success response of the API.
WizardConnection	Struct	N/A	The connection information of the API generated in wizard mode.
ConnectionId	Long	123	The ID of the connection.
TableName	String	t	The name of the table in the data source.

Parameter	Type	Example	Description
WizardErrorCodes	Array of WizardErrorCodes	N/A	The error codes returned for the API generated in wizard mode.
ErrorCode	String	1001	The error code returned.
ErrorMessage	String	fail to call	The error message returned.
ErrorSolution	String	retry	The solution used to resolve the issue.
WizardRequestParameters	Array of WizardRequestParameter	N/A	The request parameters of the API generated in wizard mode.
DefaultValue	String	default1	The default value of the parameter.
ExampleValue	String	example1	The sample value of the parameter.
IsRequiredParameter	Boolean	true	Indicates whether the request parameter is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.

Parameter	Type	Example	Description
ParameterOperator	Integer	0	The operator used for the value of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the operator is Equal. A value of 1 indicates that the operator is Like. A value of 2 indicates that the operator is Const. A value of 3 indicates that the operator is In. APIs generated in wizard mode support the Equal, Like, and In operators. APIs generated in script mode support the Equal operator. APIs generated by registration support the Equal and Const operators.
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. A value of 0 indicates that the parameter is in the URL path of the request. A value of 1 indicates that the parameter is in the Query parameter of the request URL. A value of 2 indicates that the parameter is in the request header. A value of 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only the Query position. APIs generated by registration whose request method is GET or DELETE support the Query and Head positions. APIs generated by registration whose request method is PUT or POST support the Query, Head, and Body positions.
WizardResponseParameters	Array of WizardResponseParameter	N/A	The response parameters of the API generated in wizard mode.
ExampleValue	String	example2	The sample value of the parameter.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. A value of 0 indicates the STRING data type. A value of 1 indicates the INT data type. A value of 2 indicates the LONG data type. A value of 3 indicates the FLOAT data type. A value of 4 indicates the DOUBLE data type. A value of 5 indicates the BOOLEAN data type.
ParameterDescription	String	description2	The description of the parameter.

Parameter	Type	Example	Description
ParameterName	String	param2	The name of the parameter.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

The details about the APIs of different types are returned in different parameters. The details about the API generated in wizard mode are returned in the WizardDetails parameter. The details about the API generated in script mode are returned in the ScriptDetails parameter. The details about the API generated by registration are returned in the RegistrationDetails parameter.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDataServicePublishedApi
&ApId=10001
&ProjectId=10002
&RegionId=cn-shanghai
&TenantId=10003
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <Status>1</Status>
  <VisibleRange>0</VisibleRange>
  <Protocols>0</Protocols>
  <WizardDetails>
    <Script>select a from t</Script>
    <ScriptErrorCodes>
      <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
      <ErrorMessage>fail to call</ErrorMessage>
      <ErrorSolution>try</ErrorSolution>
    </ScriptErrorCodes>
  </ScriptErrorCodes>
```

```
<DefaultValue>default1</DefaultValue>
<ExampleValue>example1</ExampleValue>
<ParameterDataType>0</ParameterDataType>
<ParameterPosition>0</ParameterPosition>
<ParameterName>param1</ParameterName>
<ParameterOperator>0</ParameterOperator>
<ParameterDescription>description1</ParameterDescription>
<IsRequiredParameter>true</IsRequiredParameter>
</ScriptErrorCodes>
<ScriptErrorCodes>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptErrorCodes>
<ServiceContentType>0</ServiceContentType>
<ScriptRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>try</ErrorSolution>
</ScriptRequestParameters>
<ScriptRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptRequestParameters>
<ScriptRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptRequestParameters>
<ServicePath>/index</ServicePath>
<WizardErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardErrorCodes>
<WizardErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardErrorCodes>
<WizardErrorCodes>
  <ExampleValue>example2</ExampleValue>
```

```

    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </WizardErrorCodes>
  <RegistrationErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationErrorCodes>
  <RegistrationErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </RegistrationErrorCodes>
  <WizardResponseParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>fail to call</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </WizardResponseParameters>
  <WizardResponseParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </WizardResponseParameters>
  <WizardResponseParameters>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </WizardResponseParameters>
  <ServiceHost>http://www.abc.com</ServiceHost>
  <RegistrationRequestParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationRequestParameters>
  <RegistrationRequestParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>

```

```
</ParameterDescription> </ParameterDescription> /> </ParameterDescription>
  </IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardRequestParameters>
<WizardRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>try</ErrorSolution>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
```

```

    <ConnectionId>123</ConnectionId>
  </ScriptConnection>
</WizardDetails>
<CreatedTime>2020-06-23T00:21:01+0800</CreatedTime>
<Description>API description</Description>
<ApiName>API name</ApiName>
<CreatorId>1234567</CreatorId>
<RequestMethod>0</RequestMethod>
<Timeout>10000</Timeout>
<ProjectId>10001</ProjectId>
<ResponseContentType>0</ResponseContentType>
<OperatorId>2345678</OperatorId>
<RegistrationDetails>
  <Script>select a from t</Script>
  <ScriptErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>fail to call</ErrorMessage>
    <ErrorSolution>try</ErrorSolution>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptErrorCodes>
  <ServiceContentType>0</ServiceContentType>
  <ScriptRequestParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>try</ErrorSolution>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>

```

```
<ParameterName>param2</ParameterName>
<ParameterDescription>description2</ParameterDescription>
</ScriptRequestParameters>
<ServicePath>/index</ServicePath>
<WizardErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardErrorCodes>
<WizardErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardErrorCodes>
<WizardErrorCodes>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardErrorCodes>
<RegistrationErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationErrorCodes>
<RegistrationErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardResponseParameters>
<WizardResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
```

```

</WizardResponseParameters>
<WizardResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>fail to call</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardRequestParameters>
<WizardRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>

```



```
<errorMessage>The connection does not exist.</errorMessage>
<ErrorSolution>try</ErrorSolution>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</RegistrationDetails>
<GroupId>abcde123456789</GroupId>
<TenantId>10001</TenantId>
<ModifiedTime>2020-06-23T00:21:01+0800</ModifiedTime>
<ScriptDetails>
  <Script>select a from t</Script>
  <ScriptErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>try</ErrorSolution>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptErrorCodes>
  <ServiceContentType>0</ServiceContentType>
  <ScriptRequestParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>try</ErrorSolution>
```

```

    </ErrorSolution>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptRequestParameters>
  <ScriptRequestParameters>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptRequestParameters>
  <ServicePath>/index</ServicePath>
  <WizardErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </WizardErrorCodes>
  <WizardErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </WizardErrorCodes>
  <WizardErrorCodes>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </WizardErrorCodes>
  <RegistrationErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </RegistrationErrorCodes>
  <RegistrationErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </RegistrationErrorCodes>

```

```
<WizardResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardResponseParameters>
<WizardResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ErrorCode>1001</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardRequestParameters>
<WizardRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
```

```

<WizardRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>try</ErrorSolution>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</ScriptDetails>
<Apild>10000</Apild>
<ApiMode>0</ApiMode>
<ApiPath>/test/1</ApiPath>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>Normal</ErrorMessage>
<Success>true</Success>

```

JSON format

```

{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "Status": 1,
    "VisibleRange": 0.
  }
}

```

```
"Protocols": 0,
"WizardDetails": {
  "Script": "select a from t",
  "ScriptErrorCodes": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "try"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    },
    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ],
  "ServiceContentType": 0,
  "ScriptRequestParameters": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "try"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    },
    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ],
  "ServicePath": "/index",
  "WizardErrorCodes": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
```

```

    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"RegistrationErrorCodes": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
],
"WizardResponseParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {

```

... ..

```

"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "try"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"ScriptConnection": {
  "TableName": "t",
  "ConnectionId": 123
}
},
"CreatedTime": "2020-06-23T00:21:01+0800",
"Description": "API description",
"ApiName": "API name",
"CreatorId": 1234567,
"RequestMethod": 0,
"Timeout": 10000,
"ProjectId": 10001,
"ResponseContentType": 0,
"OperatorId": 2345678,
"RegistrationDetails": {
  "Script": "select a from t",
  "ScriptErrorCodes": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "try"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    }
  ]
}

```



```
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceContentType": 0,
"ScriptRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "try"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServicePath": "/index",
"WizardErrorCodes": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
```

```

    "ParameterDescription": "description1",
  },
],
"RegistrationErrorCodes": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
],
"WizardResponseParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,

```

```
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\\":1}",
"WizardRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\\": false}",
"ScriptResponseParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "try"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
```

```

        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
},
"GroupId": "abcde123456789",
"TenantId": 10001,
"ModifiedTime": "2020-06-23T00:21:01+0800",
"ScriptDetails": {
    "Script": "select a from t",
    "ScriptErrorCodes": [
        {
            "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
            "ErrorMessage": "The connection does not exist.",
            "ErrorSolution": "try"
        },
        {
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0,
            "ParameterName": "param1",
            "ParameterOperator": 0,
            "ParameterDescription": "description1",
            "IsRequiredParameter": true
        },
        {
            "ExampleValue": "example2",
            "ParameterDataType": 0,
            "ParameterName": "param2",
            "ParameterDescription": "description2"
        }
    ],
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
        {
            "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
            "ErrorMessage": "The connection does not exist.",
            "ErrorSolution": "try"
        },
        {
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0,
            "ParameterName": "param1",
            "ParameterOperator": 0,
            "ParameterDescription": "description1",

```

```
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"ServicePath": "/index",
"WizardErrorCodes": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"RegistrationErrorCodes": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
],
"WizardResponseParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
```

```

      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    },
    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ],
  "ServiceHost": "http://www.abc.com",
  "RegistrationRequestParameters": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "name1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    }
  ],
  "IsPagedResponse": true,
  "ServiceRequestBodyDescription": "{\"abc\":1}",
  "WizardRequestParameters": [
    {
      "ErrorCode": 1001,
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    }
  ]
}

```

```
,
{
  "ExampleValue": "example2",
  "ParameterDataType": 0,
  "ParameterName": "param2",
  "ParameterDescription": "description2"
}
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
  "TableName": "t",
  "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
{
  "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
  "ErrorMessage": "The connection does not exist.",
  "ErrorSolution": "try"
},
{
  "DefaultValue": "default1",
  "ExampleValue": "example1",
  "ParameterDataType": 0,
  "ParameterPosition": 0,
  "ParameterName": "param1",
  "ParameterOperator": 0,
  "ParameterDescription": "description1",
  "IsRequiredParameter": true
},
{
  "ExampleValue": "example2",
  "ParameterDataType": 0,
  "ParameterName": "param2",
  "ParameterDescription": "description2"
}
],
"ScriptConnection": {
  "TableName": "t",
  "ConnectionId": 123
}
},
"Apild": 10000,
"ApiMode": 0,
"ApiPath": "/test/1"
},
"ErrorCode": 0,
"ErrorMessage": "Normal",
"Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Activate DataWorks first.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

2.1.6.6.8. ListDataServicePublishedApis

Queries the information about APIs in the published state in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServicePublishedApis	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.

Parameter	Type	Required	Example	Description
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectId	Long	Yes	10000	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10001	The tenant ID.
ApiNameKeyword	String	No	My API name	The keyword in the API name. The keyword can be used to search for the API whose name contains the keyword.
ApiPathKeyword	String	No	/test/	The keyword in the path of the API. The keyword can be used to search for the API whose path contains the keyword.
CreatorId	String	No	12345	The ID of the Alibaba Cloud account used by the creator of the API. The ID can be used to search for the API created by the creator.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The APIs in the published state.
Apis	Array of Api		The information about the APIs in the published state.
ApiId	Long	10002	The ID of the API.
ApiMode	Integer	0	The type of the API. Valid values: 0, 1, and 2. The value 0 indicates that the API is generated in wizard mode. The value 1 indicates that the API is generated in script mode. The value 2 indicates that the API is generated by registration.

Parameter	Type	Example	Description
ApiName	String	My API name	The name of the API.
ApiPath	String	/test/1	The path of the API.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the API was created.
CreatorId	String	1234567	The ID of the Alibaba Cloud account used by the creator of the API.
Description	String	Description of the test API	The description of the API.
GroupId	String	ab123	The ID of the group.
ModifiedTime	String	2020-06-23T00:21:01+0800	The time when the API was last modified.
OperatorId	String	2345678	The ID of the Alibaba Cloud account used by the user who last modified the API.
ProjectId	Long	10000	The ID of the workspace.
Protocols	List	0	The protocol used by the API. Valid values: 0 and 1. The value 0 indicates that the API uses HTTP. The value 1 indicates that the API uses HTTPS.
RegistrationDetails	Struct		The details of the API generated by registration.
FailedResultSample	String	{"success": false}	The sample error response.
RegistrationErrorCodes	Array of RegistrationErrorCode		The error codes returned for the API generated by registration.
ErrorCode	String	1001	The error code.
ErrorMessage	String	fail to call	The error message.

Parameter	Type	Example	Description
ErrorSolution	String	retry	The solution used to resolve the issue.
RegistrationRequestParameters	Array of RegistrationRequestParameter		The request parameters of the API generated by registration.
DefaultValue	String	default1	The default value of the field.
ExampleValue	String	example1	The sample value of the field.
IsRequiredParameter	Boolean	true	Indicates whether the parameter specified by ParameterName is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: • 0: STRING • 1: INT • 2: LONG • 3: FLOAT • 4: DOUBLE • 5: Boolean
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	name1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the field. Valid values: • 0: Equal • 1: Like • 2: Const • 3: In APIs generated in wizard mode support Equal, Like, and In. APIs generated in script mode support Equal. APIs generated by registration support Equal and Const.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: • 0: Path • 1: Query • 2: Head • 3: Body APIs generated in the wizard or script mode support only Query. APIs generated by registration whose request method is GET and DELETE support Query and Head. APIs generated by registration whose request method is PUT and POST support Query, Head, and Body.
ServiceContentType	Integer	0	The format in which the result of the API request is returned. Valid values: 0 and 1. The value 0 indicates the JSON format. The value 1 indicates the XML format. APIs generated in wizard or script mode support JSON. APIs generated by registration support JSON and XML.
ServiceHost	String	http://www.abc.com	The URL of the backend service.
ServicePath	String	/index	The path of the backend service.
ServiceRequestBodyDescription	String	{"abc":1}	The description of the body of the request initiated to call the backend service.
SuccessfulResultSample	String	{"success": true}	The sample success response.
RequestMethod	Integer	0	The request method of the API. Valid values: 0, 1, 2, and 3. The value 0 indicates the GET method. The value 1 indicates the POST method. The value 2 indicates the PUT method. The value 3 indicates the DELETE method. APIs generated in wizard or script mode support GET and POST. APIs generated by registration support GET, POST, PUT, and DELETE.

Parameter	Type	Example	Description
ResponseContentType	Integer	0	The format in which the result of the API request is returned. Valid values: 0 and 1. The value 0 indicates the JSON format. The value 1 indicates the XML format. APIs generated in wizard or script mode support JSON. APIs generated by registration support JSON and XML.
ScriptDetails	Struct		The details of the API generated in script mode.
FailedResultSample	String	{"success": false}	The sample error response.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
Script	String	select a from t	The SQL script.
ScriptConnection	Struct		The data source of the API generated in script mode.
ConnectionId	Long	123	The ID of the data source.
TableName	String	t	The name of the table in the data source.
ScriptErrorCodes	Array of ScriptErrorCodes		The error codes returned for the API generated in script mode.
ErrorCode	String	1001	The error code.
ErrorMessage	String	fail to call	The error message.
ErrorSolution	String	retry	The solution used to resolve the issue.
ScriptRequestParameters	Array of ScriptRequestParameter		The request parameters of the API generated in script mode.
DefaultValue	String	default1	The default value of the field.

Parameter	Type	Example	Description
ExampleValue	String	example1	The sample value of the field.
IsRequiredParameter	Boolean	true	Indicates whether the parameter specified by ParameterName is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: • 0: STRING • 1: INT • 2: LONG • 3: FLOAT • 4: DOUBLE • 5: Boolean
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the field. Valid values: • 0: Equal • 1: Like • 2: Const • 3: In APIs generated in wizard mode support Equal, Like, and In. APIs generated in script mode support Equal. APIs generated by registration support Equal and Const.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: • 0: Path • 1: Query • 2: Head • 3: Body APIs generated in wizard or script mode support only Query. APIs generated by registration whose request method is GET and DELETE support Query and Head. APIs generated by registration whose request method is PUT and POST support Query, Head, and Body.
ScriptResponseParameters	Array of ScriptResponseParameter		The response parameters of the API generated in script mode.
ExampleValue	String	example2	The sample value of the field.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: • 0: STRING • 1: INT • 2: LONG • 3: FLOAT • 4: DOUBLE • 5: Boolean
ParameterDescription	String	description2	The description of the parameter.
ParameterName	String	param2	The name of the parameter.
SuccessfulResultSample	String	{"success": true}	The sample success response.
Status	Integer	1	The status of the API. Valid values: 0 and 1. The value 0 indicates that the API is not published. The value 1 indicates that the API is published.
TenantId	Long	10001	The tenant ID.

Parameter	Type	Example	Description
Timeout	Integer	10000	The timeout of the API request. Unit: milliseconds.
VisibleRange	Integer	0	The scope in which the API is visible. Valid values: 0 and 1. The value 0 indicates that the API is visible within the workspace. The value 1 indicates that the API is visible only to the API creator.
WizardDetails	Struct		The details of the API generated in wizard mode.
FailedResultSample	String	{"success": false}	The sample error response.
IsPagedResponse	Boolean	true	Indicates whether the entries are returned by page.
SuccessfulResultSample	String	{"success": true}	The sample success response.
WizardConnection	Struct		The data source of the API generated in wizard mode.
ConnectionId	Long	123	The ID of the data source.
TableName	String	t	The name of the table in the data source.
WizardErrorCodes	Array of WizardErrorCode		The error codes returned for the API generated by registration.
ErrorCode	String	1001	The error code.
ErrorMessage	String	fail to call	The error message.
ErrorSolution	String	retry	The solution used to resolve the issue.
WizardRequestParameters	Array of WizardRequestParameter		The request parameters of the API generated by registration.

Parameter	Type	Example	Description
DefaultValue	String	default1	The default value of the field.
ExampleValue	String	example1	The sample value of the field.
IsRequiredParameter	Boolean	true	Indicates whether the parameter specified by ParameterName is required.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. The value 0 indicates the STRING data type. The value 1 indicates the INT data type. The value 2 indicates the LONG data type. The value 3 indicates the FLOAT data type. The value 4 indicates the DOUBLE data type. The value 5 indicates the Boolean data type.
ParameterDescription	String	description1	The description of the parameter.
ParameterName	String	param1	The name of the parameter.
ParameterOperator	Integer	0	The operator used for the value of the field. Valid values: 0, 1, 2, and 3. The value 0 indicates that the operator is Equal. The value 1 indicates that the operator is Like. The value 2 indicates that the operator is Const. The value 3 indicates that the operator is In. APIs generated in wizard mode support Equal, Like, and In. APIs generated in script mode support Equal. APIs generated by registration support Equal and Const.

Parameter	Type	Example	Description
ParameterPosition	Integer	0	The position of the parameter. Valid values: 0, 1, 2, and 3. The value 0 indicates that the parameter is in the URL path of the request. The value 1 indicates that the parameter is in the Query parameter of the request URL. The value 2 indicates that the parameter is in the request header. The value 3 indicates that the parameter is in the request body. APIs generated in wizard or script mode support only Query. APIs generated by registration whose request method is GET or DELETE support Query and Head. APIs generated by registration whose request method is PUT or POST support Query, Head, and Body.
WizardResponseParameters	Array of WizardResponseParameter		The response parameters of the API generated in wizard mode.
ExampleValue	String	example2	The sample value of the field.
ParameterDataType	Integer	0	The data type of the parameter. Valid values: 0, 1, 2, 3, 4, and 5. The value 0 indicates the STRING data type. The value 1 indicates the INT data type. The value 2 indicates the LONG data type. The value 3 indicates the FLOAT data type. The value 4 indicates the DOUBLE data type. The value 5 indicates the Boolean data type.
ParameterDescription	String	description2	The description of the parameter.
ParameterName	String	param2	The name of the parameter.
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	100	The total number of entries returned.

Parameter	Type	Example	Description
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

The details about the API of different types are returned in different fields. The details about the API generated in wizard mode are returned in WizardDetails fields. The details about the API generated in script mode are returned in ScriptDetails fields. The details about the API generated by registration are returned in RegistrationDetails fields.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServicePublishedApis
&PageNumber=1
&PageSize=10
&ProjectId=10000
&RegionId=cn-shanghai
&TenantId=10001
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Apis>
    <Status>1</Status>
    <VisibleRange>0</VisibleRange>
    <CreateTime>2020-06-23T00:21:01+0800</CreateTime>
    <Description>Description of the test API</Description>
    <ApiName>My API name</ApiName>
    <CreatorId>1234567</CreatorId>
```

```

<RequestMethod>0</RequestMethod>
<Timeout>10000</Timeout>
<ProjectId>10000</ProjectId>
<ResponseContentType>0</ResponseContentType>
<OperatorId>2345678</OperatorId>
<GroupId>ab123</GroupId>
<TenantId>10001</TenantId>
<ModifiedTime>2020-06-23T00:21:01+0800</ModifiedTime>
<ApId>10002</ApId>
<ApiMode>0</ApiMode>
<ApiPath>/test/1</ApiPath>
</Apis>
<Apis>
  <Protocols>0</Protocols>
</Apis>
<Apis>
  <WizardDetails>
    <Script>select a from t</Script>
    <ScriptErrorCodes>
      <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
      <ErrorMessage>fail to call</ErrorMessage>
      <ErrorSolution>retry</ErrorSolution>
    </ScriptErrorCodes>
    <ScriptErrorCodes>
      <DefaultValue>default1</DefaultValue>
      <ExampleValue>example1</ExampleValue>
      <ParameterDataType>0</ParameterDataType>
      <ParameterPosition>0</ParameterPosition>
      <ParameterName>param1</ParameterName>
      <ParameterOperator>0</ParameterOperator>
      <ParameterDescription>description1</ParameterDescription>
      <IsRequiredParameter>true</IsRequiredParameter>
    </ScriptErrorCodes>
    <ScriptErrorCodes>
      <ExampleValue>example2</ExampleValue>
      <ParameterDataType>0</ParameterDataType>
      <ParameterName>param2</ParameterName>
      <ParameterDescription>description2</ParameterDescription>
    </ScriptErrorCodes>
    <ServiceContentType>0</ServiceContentType>
    <ScriptRequestParameters>
      <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
      <ErrorMessage>The connection does not exist.</ErrorMessage>
      <ErrorSolution>retry</ErrorSolution>
    </ScriptRequestParameters>
    <ScriptRequestParameters>
      <DefaultValue>default1</DefaultValue>
      <ExampleValue>example1</ExampleValue>
      <ParameterDataType>0</ParameterDataType>
      <ParameterPosition>0</ParameterPosition>
      <ParameterName>param1</ParameterName>
      <ParameterOperator>0</ParameterOperator>
      <ParameterDescription>description1</ParameterDescription>
      <IsRequiredParameter>true</IsRequiredParameter>
    </ScriptRequestParameters>
  </WizardDetails>
</Apis>

```

```
</ScriptRequestParameters>
<ScriptRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptRequestParameters>
<ServicePath>/index</ServicePath>
<WizardErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardErrorCodes>
<WizardErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardErrorCodes>
<WizardErrorCodes>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardErrorCodes>
<RegistrationErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationErrorCodes>
<RegistrationErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardResponseParameters>
<WizardResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>naram1</ParameterName>
```

```

    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
</WizardRequestParameters>
<WizardRequestParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
    <TableName>t</TableName>
    <ConnectionId>123</ConnectionId>
</WizardConnection>

```

```
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</WizardDetails>
<ScriptDetails>
  <Script>select a from t</Script>
  <ScriptErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
  </ScriptErrorCodes>
  <ServiceContentType>0</ServiceContentType>
  <ScriptRequestParameters>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
```

```

    <ErrorSolution>retry</ErrorSolution>
</ScriptRequestParameters>
<ScriptRequestParameters>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
</ScriptRequestParameters>
<ScriptRequestParameters>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
</ScriptRequestParameters>
<ServicePath>/index</ServicePath>
<WizardErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
</WizardErrorCodes>
<WizardErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>
</WizardErrorCodes>
<WizardErrorCodes>
    <ExampleValue>example2</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterName>param2</ParameterName>
    <ParameterDescription>description2</ParameterDescription>
</WizardErrorCodes>
<RegistrationErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
</RegistrationErrorCodes>
<RegistrationErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>name1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>

```



```
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardResponseParameters>
<WizardResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardRequestParameters>
<WizardRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
```

```

</wizardrequestParameters>
<WizardRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</ScriptDetails>
<RegistrationDetails>
  <Script>select a from t</Script>
  <ScriptErrorCodes>
    <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
    <ErrorMessage>The connection does not exist.</ErrorMessage>
    <ErrorSolution>retry</ErrorSolution>
  </ScriptErrorCodes>
  <ScriptErrorCodes>
    <DefaultValue>default1</DefaultValue>
    <ExampleValue>example1</ExampleValue>
    <ParameterDataType>0</ParameterDataType>
    <ParameterPosition>0</ParameterPosition>
    <ParameterName>param1</ParameterName>
    <ParameterOperator>0</ParameterOperator>
    <ParameterDescription>description1</ParameterDescription>
    <IsRequiredParameter>true</IsRequiredParameter>

```

```
</IsRequiredParameter>true</IsRequiredParameter>
</ScriptErrorCodes>
<ScriptErrorCodes>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptErrorCodes>
<ServiceContentType>0</ServiceContentType>
<ScriptRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</ScriptRequestParameters>
<ScriptRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptRequestParameters>
<ScriptRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptRequestParameters>
<ServicePath>/index</ServicePath>
<WizardErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardErrorCodes>
<WizardErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardErrorCodes>
<WizardErrorCodes>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardErrorCodes>
<RegistrationErrorCodes>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
```

```

    <ErrorSolution>retry</ErrorSolution>
  </RegistrationErrorCodes>
</RegistrationErrorCodes>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationErrorCodes>
<WizardResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</WizardResponseParameters>
<WizardResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardResponseParameters>
<WizardResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardResponseParameters>
<ServiceHost>http://www.abc.com</ServiceHost>
<RegistrationRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</RegistrationRequestParameters>
<RegistrationRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>name1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</RegistrationRequestParameters>
<IsPagedResponse>true</IsPagedResponse>
<ServiceRequestBodyDescription>{"abc":1}</ServiceRequestBodyDescription>
<WizardRequestParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>

```

```
<ErrorSolution>retry</ErrorSolution>
</WizardRequestParameters>
<WizardRequestParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</WizardRequestParameters>
<WizardRequestParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</WizardRequestParameters>
<SuccessfulResultSample>{"success": true}</SuccessfulResultSample>
<WizardConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</WizardConnection>
<FailedResultSample>{"success": false}</FailedResultSample>
<ScriptResponseParameters>
  <ErrorCode>Invalid.Tenant.ConnectionNotExists</ErrorCode>
  <ErrorMessage>The connection does not exist.</ErrorMessage>
  <ErrorSolution>retry</ErrorSolution>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <DefaultValue>default1</DefaultValue>
  <ExampleValue>example1</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterPosition>0</ParameterPosition>
  <ParameterName>param1</ParameterName>
  <ParameterOperator>0</ParameterOperator>
  <ParameterDescription>description1</ParameterDescription>
  <IsRequiredParameter>true</IsRequiredParameter>
</ScriptResponseParameters>
<ScriptResponseParameters>
  <ExampleValue>example2</ExampleValue>
  <ParameterDataType>0</ParameterDataType>
  <ParameterName>param2</ParameterName>
  <ParameterDescription>description2</ParameterDescription>
</ScriptResponseParameters>
<ScriptConnection>
  <TableName>t</TableName>
  <ConnectionId>123</ConnectionId>
</ScriptConnection>
</RegistrationDetails>
</Apis>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>Normal</ErrorMessage>
```

```
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "Apis": [
      {
        "Status": 1,
        "VisibleRange": 0,
        "CreatedTime": "2020-06-23T00:21:01+0800",
        "Description": "Description of the test API",
        "ApiName": "My API name",
        "CreatorId": 1234567,
        "RequestMethod": 0,
        "Timeout": 10000,
        "ProjectId": 10000,
        "ResponseContentType": 0,
        "OperatorId": 2345678,
        "GroupId": "ab123",
        "TenantId": 10001,
        "ModifiedTime": "2020-06-23T00:21:01+0800",
        "ApId": 10002,
        "ApiMode": 0,
        "ApiPath": "/test/1"
      },
      {
        "Protocols": 0
      },
      {
        "WizardDetails": {
          "Script": "select a from t",
          "ScriptErrorCodes": [
            {
              "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
              "ErrorMessage": "fail to call",
              "ErrorSolution": "retry"
            },
            {
              "DefaultValue": "default1",
              "ExampleValue": "example1",
              "ParameterDataType": 0,
              "ParameterPosition": 0,
              "ParameterName": "param1",
              "ParameterOperator": 0,
              "ParameterDescription": "description1",
              "IsRequiredParameter": true
            }
          ]
        }
      }
    ]
  }
}
```

```
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceContentType": 0,
"ScriptRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServicePath": "/index",
"WizardErrorCodes": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
```

```

"RegistrationErrorCodes": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
],
"WizardResponseParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
]

```



```
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":\"1\"}",
"WizardRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2"
```

```

        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
},
"ScriptDetails": {
    "Script": "select a from t",
    "ScriptErrorCodes": [
        {
            "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
            "ErrorMessage": "The connection does not exist.",
            "ErrorSolution": "retry"
        },
        {
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0,
            "ParameterName": "param1",
            "ParameterOperator": 0,
            "ParameterDescription": "description1",
            "IsRequiredParameter": true
        },
        {
            "ExampleValue": "example2",
            "ParameterDataType": 0,
            "ParameterName": "param2",
            "ParameterDescription": "description2"
        }
    ],
    "ServiceContentType": 0,
    "ScriptRequestParameters": [
        {
            "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
            "ErrorMessage": "The connection does not exist.",
            "ErrorSolution": "retry"
        },
        {
            "DefaultValue": "default1",
            "ExampleValue": "example1",
            "ParameterDataType": 0,
            "ParameterPosition": 0,
            "ParameterName": "param1",
            "ParameterOperator": 0,
            "ParameterDescription": "description1",
            "IsRequiredParameter": true
        },
        {
            "ExampleValue": "example2",
            "ParameterDataType": 0,
            "ParameterName": "param2"
        }
    ]
}

```

```
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    },
],
"ServicePath": "/index",
"WizardErrorCodes": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"RegistrationErrorCodes": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"WizardResponseParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
```

```

        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "name1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":1}",
"WizardRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
]

```

```
    }
  ],
  "SuccessfulResultSample": "{\"success\": true}",
  "WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
  },
  "FailedResultSample": "{\"success\": false}",
  "ScriptResponseParameters": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    },
    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ],
  "ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
  }
},
"RegistrationDetails": {
  "Script": "select a from t",
  "ScriptErrorCodes": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    }
  ],
}
```

```

    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ],
  "ServiceContentType": 0,
  "ScriptRequestParameters": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    },
    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ],
  "ServicePath": "/index",
  "WizardErrorCodes": [
    {
      "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
      "ErrorMessage": "The connection does not exist.",
      "ErrorSolution": "retry"
    },
    {
      "DefaultValue": "default1",
      "ExampleValue": "example1",
      "ParameterDataType": 0,
      "ParameterPosition": 0,
      "ParameterName": "param1",
      "ParameterOperator": 0,
      "ParameterDescription": "description1",
      "IsRequiredParameter": true
    },
    {
      "ExampleValue": "example2",
      "ParameterDataType": 0,
      "ParameterName": "param2",
      "ParameterDescription": "description2"
    }
  ]
}

```

```
,
"RegistrationErrorCodes": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  }
],
"WizardResponseParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "param1",
    "ParameterOperator": 0,
    "ParameterDescription": "description1",
    "IsRequiredParameter": true
  },
  {
    "ExampleValue": "example2",
    "ParameterDataType": 0,
    "ParameterName": "param2",
    "ParameterDescription": "description2"
  }
],
"ServiceHost": "http://www.abc.com",
"RegistrationRequestParameters": [
  {
    "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
    "ErrorMessage": "The connection does not exist.",
    "ErrorSolution": "retry"
  },
  {
    "DefaultValue": "default1",
    "ExampleValue": "example1",
    "ParameterDataType": 0,
    "ParameterPosition": 0,
    "ParameterName": "name1",
    "ParameterOperator": 0
```

```

        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    }
],
"IsPagedResponse": true,
"ServiceRequestBodyDescription": "{\"abc\":1}",
"WizardRequestParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,
        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"SuccessfulResultSample": "{\"success\": true}",
"WizardConnection": {
    "TableName": "t",
    "ConnectionId": 123
},
"FailedResultSample": "{\"success\": false}",
"ScriptResponseParameters": [
    {
        "ErrorCode": "Invalid.Tenant.ConnectionNotExists",
        "ErrorMessage": "The connection does not exist.",
        "ErrorSolution": "retry"
    },
    {
        "DefaultValue": "default1",
        "ExampleValue": "example1",
        "ParameterDataType": 0,
        "ParameterPosition": 0,
        "ParameterName": "param1",
        "ParameterOperator": 0,
        "ParameterDescription": "description1",
        "IsRequiredParameter": true
    },
    {
        "ExampleValue": "example2",
        "ParameterDataType": 0,

```



```

        "ParameterName": "param2",
        "ParameterDescription": "description2"
    }
],
"ScriptConnection": {
    "TableName": "t",
    "ConnectionId": 123
}
}
]
},
"ErrorCode": 0,
"ErrorMessage": "Normal",
"Success": true
}

```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.9. ListDataServiceApiAuthorities

Queries the APIs whose access permissions are granted to users.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServiceApiAuthorities	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectId	Long	Yes	10000	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10001	The tenant ID.
ApiNameKeyword	String	No	My API name	The keyword in the name of the API. The keyword can be used to search for the API whose name contains the keyword.

Response parameters

Parameter	Type	Example	Description
Data	Struct		The APIs whose access permissions are granted to users.
ApiAuthorizationList	Array of ApiAuthorization		The details of the APIs whose access permissions are granted to users.
ApiId	Long	10002	The ID of the API.
ApiName	String	My API name	The name of the API.

Parameter	Type	Example	Description
ApiPath	String	/test/1	The path of the API.
ApiStatus	Integer	0	The status of the API. Valid values: 0 and 1. The value 0 indicates that the API is not published. The value 1 indicates that the API is published.
AuthorizationRecords	Array of AuthorizationRecord		The authorization records.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the access permissions of the API were granted to users.
CreatorId	String	12345	The ID of the Alibaba Cloud account used to grant access permissions on the API to users.
EndTime	String	2020-06-24T00:21:01+0800	The end time of the validity period of the permissions.
ProjectId	Long	10004	The ID of the workspace that is authorized.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the API was created.
CreatorId	String	12345	The ID of the Alibaba Cloud account used by the API owner.
GroupId	String	ab123	The ID of the group.
ModifiedTime	String	2020-06-23T00:21:01+0800	The time when the API was last updated.
ProjectId	Long	10000	The ID of the workspace.
TenantId	Long	10001	The tenant ID.
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.

Parameter	Type	Example	Description
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	100	The total number of entries returned.
ErrorCode	String	0	The error code returned.
ErrorMessage	String	Normal	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the request is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServiceApiAuthorities
&PageNumber=1
&PageSize=10
&ProjectId=10000
&RegionId=cn-shanghai
&TenantId=10001
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <ApiAuthorizationList>
    <TenantId>10001</TenantId>
    <CreatedTime>2020-06-23T00:21:01+0800</CreatedTime>
    <ModifiedTime>2020-06-23T00:21:01+0800</ModifiedTime>
    <ApiName>My API name</ApiName>
    <CreatorId>12345</CreatorId>
    <ApiStatus>0</ApiStatus>
    <ProjectId>10000</ProjectId>
    <ApId>10002</ApId>
    <ApiPath>/test/1</ApiPath>
    <GroupId>ab123</GroupId>
  </ApiAuthorizationList>
  <ApiAuthorizationList>
    <AuthorizationRecords>
      <CreatedTime>2020-06-23T00:21:01+0800</CreatedTime>
      <EndTime>2020-06-24T00:21:01+0800</EndTime>
      <CreatorId>12345</CreatorId>
      <ProjectId>10004</ProjectId>
    </AuthorizationRecords>
  </ApiAuthorizationList>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>Normal</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "ApiAuthorizationList": [
      {
        "TenantId": 10001,
        "CreatedTime": "2020-06-23T00:21:01+0800",
        "ModifiedTime": "2020-06-23T00:21:01+0800",
        "ApiName": "My API name",
        "CreatorId": 12345,
        "ApiStatus": 0,
        "ProjectId": 10000,
        "ApiId": 10002,
        "ApiPath": "/test/1",
        "GroupId": "ab123"
      },
      {
        "AuthorizationRecords": {
          "CreatedTime": "2020-06-23T00:21:01+0800",
          "EndTime": "2020-06-24T00:21:01+0800",
          "CreatorId": 12345,
          "ProjectId": 10004
        }
      }
    ]
  },
  "ErrorCode": 0,
  "ErrorMessage": "Normal",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.

HTTP status code	Error code	Error message	Description

429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted in a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.10. ListDataServiceAuthorizedApis

Queries authorized APIs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServiceAuthorizedApis	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectId	Long	Yes	10000	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.

Parameter	Type	Required	Example	Description
TenantId	Long	Yes	10001	The ID of the tenant.
ApiNameKeyword	String	No	My API name	The keyword in the name of the API. The keyword can be used to search for the API whose name contains the keyword.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The information about authorized APIs.
ApiAuthorizedList	Array of ApiAuthorized	N/A	The list of authorized APIs.
ApiId	Long	10002	The ID of the API.
ApiName	String	My API name	The name of the API.
ApiPath	String	/test/1	The path of the API.
ApiStatus	Integer	0	The status of the API. Valid values: 0 and 1. A value of 0 indicates that the API is not published. A value of 1 indicates that the API is published.
CreatedTime	String	2020-06-23T00:21:01+0800	The time when the API was created.
CreatorId	String	12345	The ID of the Apsara Stack tenant account used by the API owner.
GrantCreatedTime	String	2020-06-23T00:21:01+0800	The time when the access permissions of the API were granted.
GrantEndTime	String	2020-06-24T00:21:01+0800	The expiration time for the granted access permissions of the API.

Parameter	Type	Example	Description
GrantOperatorId	String	23456	The ID of the Apsara Stack tenant account used by the user who granted the access permissions of the API.
GroupId	String	abcde123456789	The ID of the API group.
ModifiedTime	String	2020-06-23T00:21:01+0800	The time when the API was last updated.
ProjectId	Long	10000	The ID of the workspace.
TenantId	Long	10001	The ID of the tenant.
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	100	The total number of entries returned.
ErrorCode	String	1031203110005	The error code returned.
ErrorMessage	String	The specified parameters are invalid.	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServiceAuthorizedApis
&PageNumber=1
&PageSize=10
&ProjectId=10000
&RegionId=cn-shanghai
&TenantId=10001
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <ApiAuthorizedList>
    <GrantEndTime>2020-06-24T00:21:01+0800</GrantEndTime>
    <CreatedTime>2020-06-23T00:21:01+0800</CreatedTime>
    <ApiName>My API name</ApiName>
    <CreatorId>12345</CreatorId>
    <ApiStatus>0</ApiStatus>
    <ProjectId>10000</ProjectId>
    <GrantCreatedTime>2020-06-23T00:21:01+0800</GrantCreatedTime>
    <GroupId>abcde123456789</GroupId>
    <TenantId>10001</TenantId>
    <ModifiedTime>2020-06-23T00:21:01+0800</ModifiedTime>
    <GrantOperatorId>23456</GrantOperatorId>
    <ApiId>10002</ApiId>
    <ApiPath>/test/1</ApiPath>
  </ApiAuthorizedList>
</Data>
<ErrorCode>1031203110005</ErrorCode>
<ErrorMessage>The specified parameters are invalid.</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "ApiAuthorizedList": {
      "GrantEndTime": "2020-06-24T00:21:01+0800",
      "CreatedTime": "2020-06-23T00:21:01+0800",
      "ApiName": "My API name",
      "CreatorId": 12345,
      "ApiStatus": 0,
      "ProjectId": 10000,
      "GrantCreatedTime": "2020-06-23T00:21:01+0800",
      "GroupId": "abcde123456789",
      "TenantId": 10001,
      "ModifiedTime": "2020-06-23T00:21:01+0800",
      "GrantOperatorId": 23456,
      "ApId": 10002,
      "ApiPath": "/test/1"
    }
  },
  "ErrorCode": 1031203110005,
  "ErrorMessage": "The specified parameters are invalid.",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6.11. AbolishDataServiceApi

Unpublishes an API in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	AbolishDataServiceApi	The operation that you want to perform.
ApiId	Long	Yes	10000	The ID of the API.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10002	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description
Data	Boolean	true	Indicates whether the API is unpublished.
ErrorCode	String	Invalid.Tenant.ConnectionNotExists	The error code returned.
ErrorMessage	String	The connection does not exist.	The error message returned.

Parameter	Type	Example	Description
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=AbolishDataServiceApi
&ApiId=10000
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>true</Data>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": true,
  "Success": true
}
```

2.1.6.6.12. GetDataServiceApplication

Queries the details of an application in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDataServiceApplication	The operation that you want to perform.

Parameter	Type	Required	Example	Description
ApplicationId	Long	Yes	10000	The ID of the application. You can view the application information in the API Gateway console.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10002	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The details of the application.
ApplicationCode	String	CODE123	The AppCode for simple authentication. You can select either simple authentication or signature authentication when you call an API.
ApplicationId	Long	10000	The ID of the application.
ApplicationKey	String	KEY123	The AppKey for signature authentication. You can select either simple authentication or signature authentication when you call an API.
ApplicationName	String	Test application	The name of the application.
ApplicationSecret	String	SECRET123	The AppSecret for signature authentication. You can select either simple authentication or signature authentication when you call an API.
ProjectId	Long	10001	The ID of the workspace.
ErrorCode	String	0	The error code returned.

Parameter	Type	Example	Description
ErrorMessage	String	Normal	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG****	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

The disclosure of the `ApplicationCode` or `ApplicationSecret` parameter may threaten the security of your API resources in Data Service Studio. Therefore, the values of the `ApplicationCode` and `ApplicationSecret` parameters are de-identified before they are returned.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDataServiceApplication
&ApplicationId=10000
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <ApplicationKey>KEY123</ApplicationKey>
  <ApplicationName>Test application</ApplicationName>
  <ApplicationSecret>*****</ApplicationSecret>
  <ProjectId>10001</ProjectId>
  <ApplicationId>10000</ApplicationId>
  <ApplicationCode>*****</ApplicationCode>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>Normal</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "ApplicationKey": "KEY123",
    "ApplicationName": "Test application",
    "ApplicationSecret": "*****",
    "ProjectId": 10001,
    "ApplicationId": 10000,
    "ApplicationCode": "*****"
  },
  "ErrorCode": 0,
  "ErrorMessage": "Normal",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6.13. ListDataServiceApplications

Queries the basic information of applications in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServiceApplications	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectIdList	String	Yes	10000,100001	The ID of the workspace that is used to query applications. You can specify multiple IDs. Separate them with commas (,). You must specify at least one workspace ID. You can specify a maximum of 50 workspace IDs.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	100002	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description
Data	Struct	N/A	The basic information of queried applications.
Applications	Array of Application	N/A	The basic information of applications.
ApplicationId	Long	20000	The ID of the application.
ApplicationName	String	My application	The name of the application.
ProjectId	Long	10000	The ID of the workspace.

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	100	The total number of entries returned.
ErrorCode	String	0	The error code returned.
ErrorMessage	String	Normal	The error message returned.
HttpStatusCode	Integer	200	The HTTP status code returned.
RequestId	String	0000-ABCD-EFG***	The ID of the request.
Success	Boolean	true	Indicates whether the call is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServiceApplications
&PageNumber=1
&PageSize=10
&ProjectIdList=10000,100001
&RegionId=cn-shanghai
&TenantId=100002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFG****</RequestId>
<HttpStatusCode>200</HttpStatusCode>
<Data>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <Applications>
    <ApplicationName>My application</ApplicationName>
    <ProjectId>10000</ProjectId>
    <ApplicationId>20000</ApplicationId>
  </Applications>
  <PageNumber>1</PageNumber>
</Data>
<ErrorCode>0</ErrorCode>
<ErrorMessage>Normal</ErrorMessage>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFG****",
  "HttpStatusCode": 200,
  "Data": {
    "TotalCount": 100,
    "PageSize": 10,
    "Applications": {
      "ApplicationName": "My application",
      "ProjectId": 10000,
      "ApplicationId": 20000
    },
    "PageNumber": 1
  },
  "ErrorCode": 0,
  "ErrorMessage": "Normal",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6.14. CreateDataServiceFolder

Creates a folder in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDataServiceFolder	The operation that you want to perform.
FolderName	String	Yes	Test folder	The name of the folder.
GroupId	String	Yes	1000abcd	The ID of the workflow to which the folder belongs.
ParentId	Long	Yes	0	The ID of the parent folder of the folder that you want to create. The ID of the root folder in a workflow is 0. The ID of a folder created by a user in a workflow is greater than 0.
ProjectId	Long	Yes	10001	The ID of the workspace.

Parameter	Type	Required	Example	Description
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10002	The ID of the tenant.

Response parameters

Parameter	Type	Example	Description
FolderId	Long	123	The ID of the created folder.
RequestId	String	0000-ABCD-E****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateDataServiceFolder
&FolderName=Test folder
&GroupId=1000abcd
&ParentId=0
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-E****</RequestId>
<FolderId>123</FolderId>
```

JSON format

```
{
  "RequestId": "0000-ABCD-E****",
  "FolderId": 123
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for this resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or DataWorks Ultimate Edition.

2.1.6.6.15. GetDataServiceFolder

Queries the details of a folder in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDataServiceFolder	The operation that you want to perform.
FolderId	Long	Yes	10001	The ID of the folder.
ProjectId	Long	Yes	10002	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.

Parameter	Type	Required	Example	Description
TenantId	Long	Yes	10003	The tenant ID.

Response parameters

Parameter	Type	Example	Description
Folder	Struct		The details of the folder.
CreateTime	String	2020-09-24T18:37:51+0800	The time when the folder was created.
FolderId	Long	11	The ID of the folder.
FolderName	String	test1	The name of the folder.
GroupId	String	ds_1234	The ID of the business process to which the folder belongs.
ModifiedTime	String	2020-09-24T18:37:51+0800	The time when the folder was last modified.
ParentId	Long	0	The ID of the parent folder. The ID of the root folder in a business process is 0, and the ID of a folder created by a user in a business process is greater than 0.
ProjectId	Long	10001	The ID of the workspace.
TenantId	Long	10002	The tenant ID.
RequestId	String	0000-ABCD-EF****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDataServiceFolder
&FolderId=10001
&ProjectId=10002
&RegionId=cn-shanghai
&TenantId=10003
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EF****</RequestId>
<Folder>
  <ParentId>0</ParentId>
  <TenantId>10002</TenantId>
  <CreateTime>2020-09-24T18:37:51+0800</CreateTime>
  <ModifiedTime>2020-09-24T18:37:51+0800</ModifiedTime>
  <ProjectId>10001</ProjectId>
  <FolderId>11</FolderId>
  <FolderName>test1</FolderName>
  <GroupId>ds_1234</GroupId>
</Folder>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EF****",
  "Folder": {
    "ParentId": 0,
    "TenantId": 10002,
    "CreateTime": "2020-09-24T18:37:51+0800",
    "ModifiedTime": "2020-09-24T18:37:51+0800",
    "ProjectId": 10001,
    "FolderId": 11,
    "FolderName": "test1",
    "GroupId": "ds_1234"
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
403	ResourceNotAuthorized.Api	You are not authorized to access the resources.	The error message returned because you are not authorized to access the resources.

HTTP status code	Error code	Error message	Description
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

2.1.6.6.16. ListDataServiceFolders

Queries folders in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServiceFolders	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10002	The tenant ID.
GroupId	String	No	ds_123abc	The ID of the business process to which the folder belongs.

Parameter	Type	Required	Example	Description
FolderNameKeyword	String	No	Keyword in folder names	The keyword in folder names. The keyword is used to search for folders whose names contain this keyword.

Response parameters

Parameter	Type	Example	Description
FolderPagingResult	Struct		The paging result for the folders.
Folders	Array of Folder		The folders.
CreatedTime	String	2020-09-24T18:37:51+0800	The time when the folder was created.
FolderId	Long	11	The ID of the folder.
FolderName	String	test1	The name of the folder.
GroupId	String	ds_1234	The ID of the business process to which the folder belongs.
ModifiedTime	String	2020-09-24T18:37:51+0800	The time when the folder was last modified.
ParentId	Long	0	The ID of the parent folder. The ID of the root folder in a business process is 0, and the ID of a folder created by a user in a business process is greater than 0.
ProjectId	Long	10001	The ID of the workspace.
TenantId	Long	10002	The ID of the tenant.
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.

Parameter	Type	Example	Description
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.
TotalCount	Integer	100	The total number of entries returned.
RequestId	String	0000-ABCD-EF****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServiceFolders
&PageNumber=1
&PageSize=10
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EF****</RequestId>
<FolderPagingResult>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Folders>
    <ParentId>0</ParentId>
    <TenantId>10002</TenantId>
    <CreatedTime>2020-09-24T18:37:51+0800</CreatedTime>
    <ModifiedTime>2020-09-24T18:37:51+0800</ModifiedTime>
    <ProjectId>10001</ProjectId>
    <FolderId>11</FolderId>
    <FolderName>test1</FolderName>
    <GroupId>ds_1234</GroupId>
  </Folders>
</FolderPagingResult>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EF****",
  "FolderPagingResult": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "Folders": {
      "ParentId": 0,
      "TenantId": 10002,
      "CreatedTime": "2020-09-24T18:37:51+0800",
      "ModifiedTime": "2020-09-24T18:37:51+0800",
      "ProjectId": 10001,
      "FolderId": 11,
      "FolderName": "test1",
      "GroupId": "ds_1234"
    }
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.17. CreateDataServiceGroup

Creates a business process in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDataServiceGroup	The operation that you want to perform.
ApiGatewayGroupId	String	Yes	1000abcd	The ID of the API group that is associated with the business process in the API Gateway console. You can log on to the API Gateway console and go to the Group Details page to view the ID.
Description	String	Yes	Test business process	The description of the business process.
GroupName	String	Yes	Business process name	The name of the business process.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10002	The tenant ID.

Response parameters

Parameter	Type	Example	Description
GroupId	String	ds_12345	The ID of the business process.
RequestId	String	0000-ABCD-EF****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateDataServiceGroup
&ApiGatewayGroupId=1000abcd
&Description=Test business process
&GroupName=Business process name
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EF****</RequestId>
<GroupId>ds_12345</GroupId>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EF****",
  "GroupId": "ds_12345"
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.18. GetDataServiceGroup

Queries a business process in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	GetDataServiceGroup	The operation that you want to perform.
GroupId	String	Yes	ds_123abc	The ID of the business process.
ProjectId	Long	Yes	10001	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10002	The tenant ID.

Response parameters

Parameter	Type	Example	Description
Group	Struct		The details of the business process.
ApiGatewayGroupid	String	100abc	The ID of the API group that is associated with the business process in the API Gateway console.
CreatedTime	String	2020-09-24T18:37:51+0800	The time when the business process was created.

Parameter	Type	Example	Description
CreatorId	String	10001	The user identifier (UID) of the creator of the business process. The value of this parameter may be empty for creators of some existing business processes.
Description	String	Business process description	The description of the business process.
GroupId	String	ds_123abc	The ID of the business process.
GroupName	String	Business process name	The name of the business process.
ModifiedTime	String	2020-09-24T18:37:51+0800	The time when the business process was last modified.
ProjectId	Long	10002	The ID of the business process.
TenantId	Long	10003	The tenant ID.
RequestId	String	0000-ABCD-EF****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=GetDataServiceGroup
&GroupId=ds_123abc
&ProjectId=10001
&RegionId=cn-shanghai
&TenantId=10002
&<Common request parameters>
```

Sample success responses

XML format


```
<Group>
  <GroupName>Business process name</GroupName>
  <TenantId>10003</TenantId>
  <Description>Business process description</Description>
  <CreateTime>2020-09-24T18:37:51+0800</CreateTime>
  <ModifiedTime>2020-09-24T18:37:51+0800</ModifiedTime>
  <CreatorId>10001</CreatorId>
  <ProjectId>10002</ProjectId>
  <ApiGatewayGroupId>100abc</ApiGatewayGroupId>
  <GroupId>ds_123abc</GroupId>
</Group>
<RequestId>0000-ABCD-EF****</RequestId>
```

JSON format

```
{
  "Group": {
    "GroupName": "Business process name",
    "TenantId": 10003,
    "Description": "Business process description",
    "CreateTime": "2020-09-24T18:37:51+0800",
    "ModifiedTime": "2020-09-24T18:37:51+0800",
    "CreatorId": 10001,
    "ProjectId": 10002,
    "ApiGatewayGroupId": "100abc",
    "GroupId": "ds_123abc"
  },
  "RequestId": "0000-ABCD-EF****"
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalError.System	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.

HTTP status code	Error code	Error message	Description
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.19. ListDataServiceGroups

Queries business processes in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	ListDataServiceGroups	The operation that you want to perform.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
PageSize	Integer	Yes	10	The number of entries to return on each page. Default value: 10. Maximum value: 100.
ProjectId	Long	Yes	10000	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The ID of the region.
TenantId	Long	Yes	10001	The ID of the tenant.
GroupNameKeyword	String	No	Test business process	The keyword in the name of the business process name. You can use the keyword to search for the business process whose name contains the keyword.

Response parameters

Parameter	Type	Example	Description
GroupPagingResult	Struct		The paging result for the business processes.
Groups	Array of Group		The business processes.
ApiGatewayGroupId	String	100abc	The ID of the API group that is associated with the business process in the API Gateway console.
CreatedTime	String	2020-09-24T18:37:51+0800	The time when the business process was created.
CreatorId	String	10001	The user identifier (UID) of the creator of the business process. The value of this parameter may be empty for creators of some existing business processes.
Description	String	Business process description	The description of the business process.
GroupId	String	ds_123abc	The ID of the business process.
GroupName	String	Business process name	The name of the business process.
ModifiedTime	String	2020-09-24T18:37:51+0800	The time when the business process was last modified.
ProjectId	Long	10002	The ID of the workspace.
TenantId	Long	10003	The tenant ID.
PageNumber	Integer	1	The page number of the returned page. The value of this parameter is the same as that of the PageNumber parameter in the request.
PageSize	Integer	10	The number of entries returned per page. Default value: 10. Maximum value: 100.

Parameter	Type	Example	Description
TotalCount	Integer	100	The total number of entries returned.
RequestId	String	0000-ABCD-EF****	The ID of the request.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=ListDataServiceGroups
&PageNumber=1
&PageSize=10
&ProjectId=10000
&RegionId=cn-shanghai
&TenantId=10001
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EF****</RequestId>
<GroupPagingResult>
  <TotalCount>100</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <Groups>
    <GroupName>Business process name</GroupName>
    <TenantId>10003</TenantId>
    <Description>Business process description</Description>
    <CreateTime>2020-09-24T18:37:51+0800</CreateTime>
    <ModifiedTime>2020-09-24T18:37:51+0800</ModifiedTime>
    <CreatorId>10001</CreatorId>
    <ApiGatewayGroupId>100abc</ApiGatewayGroupId>
    <ProjectId>10002</ProjectId>
    <GroupId>ds_123abc</GroupId>
  </Groups>
</GroupPagingResult>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EF****",
  "GroupPagingResult": {
    "TotalCount": 100,
    "PageSize": 10,
    "PageNumber": 1,
    "Groups": {
      "GroupName": "Business process name",
      "TenantId": 10003,
      "Description": "Business process description",
      "CreatedTime": "2020-09-24T18:37:51+0800",
      "ModifiedTime": "2020-09-24T18:37:51+0800",
      "CreatorId": 10001,
      "ApiGatewayGroupId": "100abc",
      "ProjectId": 10002,
      "GroupId": "ds_123abc"
    }
  }
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.20. CreateDataServiceApiAuthority

Grants the access permissions on an API in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	CreateDataServiceApiAuthority	The operation that you want to perform.
ApId	Long	Yes	10001	The ID of the API.
AuthorizedProjectId	Long	Yes	10002	The ID of the workspace that is authorized.
EndTime	Long	Yes	1600531564	The end time of the validity period of the permissions. The time must be a UNIX timestamp. Unit: seconds. Example: 1600531564, which indicates 2020-09-20 00:06:04 (UTC+8).
ProjectId	Long	Yes	10003	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	1004	The tenant ID.

Response parameters

Parameter	Type	Example	Description
RequestId	String	0000-ABCD-EF****	The ID of the request.
Success	Boolean	true	Indicates whether the authorization is successful.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=CreateDataServiceApiAuthority
&ApId=10001
&AuthorizedProjectId=10002
&EndTime=1600531564
&ProjectId=10003
&RegionId=cn-shanghai
&TenantId=1004
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFo****</RequestId>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EFo****",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.UserId.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.6.6.21. DeleteDataServiceApiAuthority

Revokes the access permissions on an API in DataService Studio.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DeleteDataServiceApiAuthority	The operation that you want to perform.
ApId	Long	Yes	10001	The ID of the API.
AuthorizedProjectId	Long	Yes	10002	The ID of the workspace from which you want to revoke the access permissions on the API.
ProjectId	Long	Yes	10003	The ID of the workspace.
RegionId	String	Yes	cn-shanghai	The region ID.
TenantId	Long	Yes	10004	The tenant ID.

Response parameters

Parameter	Type	Example	Description
RequestId	String	0000-ABCD-EF****	The ID of the request.
Success	Boolean	true	Indicates whether the access permissions are revoked.

Examples

Sample requests

```
http(s)://[Endpoint]/?Action=DeleteDataServiceApiAuthority
&ApiId=10001
&AuthorizedProjectId=10002
&ProjectId=10003
&RegionId=cn-shanghai
&TenantId=10004
&<Common request parameters>
```

Sample success responses

XML format

```
<RequestId>0000-ABCD-EFGH-IJKL</RequestId>
<Success>true</Success>
```

JSON format

```
{
  "RequestId": "0000-ABCD-EF****",
  "Success": true
}
```

Error codes

HTTP status code	Error code	Error message	Description
500	InternalServerError	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
500	InternalServerError.Missing	An internal system error occurred. Try again later.	The error message returned because an internal error has occurred. Try again later.
429	Throttling.Api	The request for this resource has exceeded your available limit.	The error message returned because the number of requests for the resource has exceeded the upper limit.
429	Throttling.System	The DataWorks system is busy. Try again later.	The error message returned because the DataWorks system is busy. Try again later.
429	Throttling.User	Your request is too frequent. Try again later.	The error message returned because excessive requests have been submitted within a short period of time. Try again later.

HTTP status code	Error code	Error message	Description
403	Forbidden.Access	Access is forbidden. Please first activate DataWorks Enterprise Edition or Flagship Edition.	The error message returned because you are not allowed to perform this operation. Activate DataWorks Enterprise Edition or Ultimate Edition.

2.1.7. Obtain an AccessKey pair

You can obtain an AccessKey pair from the Apsara Stack Cloud Management (ASCM) console.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, follow these steps:

1. Log on to the 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 an organization.
5. Choose **AccessKey** from the short cut menu.
6. In the AccessKey dialog box 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, follow these steps:

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

Apsara Stack AccessKey Pair You must use the AccessKey pair when you access Apsara Stack resources.		
The AccessKey pair including the AccessKey ID and AccessKey secret is the credential for you to use Apsara Stack resources with full permissions. You must keep the AccessKey pair confidential.		
Region	AccessKey ID	AccessKey Secret
cn-qingdao-env4b-d01		 Show

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

3. Realtime Compute

3.1. Developer Guide

3.1.1. Preparations

3.1.1.1. Log on to the API Tool console

You can log on to the API Tool console from the Apsara Uni-manager Management Console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
 - It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the account and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA authentication code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Operations Console User Guide*.

5. In the top navigation bar, choose **Products > Others > API Tool**.

3.1.1.2. View information about API operations

This topic describes how to view information about API operations. Before you use SDKs, obtain service names, API operation names, and API operation versions.

Context

The API operations on the API Directory page are only for reference. For information about how to call API operations, see developer guides of the corresponding cloud services. If the information on the API Directory page is inconsistent with that in the developer guide of the corresponding cloud service, the information in the developer guide prevails.

Procedure

1. Go to the API Tool console. For more information, see [Log on to the API Tool console](#).
2. In the left-side navigation pane, click **API Directory**.
3. On the **API Directory** page, select the desired service from the drop-down list, enter a keyword of an API operation name in the search box, and then click the search icon.

API Directory		
All Products ▾	cre 	
Product Name	Operation Name	API Version
cr	CreateMetadataOccurrence	2018-12-01
cr	CreateUserInfo	2016-06-07
cr	CreateNamespaceAuthorization	2016-06-07
cr	CreateNamespace	2016-06-07

4. Then, you can view the service name, API operation name, and API operation version.

3.1.1.3. Obtain an AccessKey pair

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

Prerequisites

An account that assumes the operations administrator or level-1 organization administrator role is obtained. Only an operations administrator or a level-1 organization administrator can obtain an AccessKey pair of the organization.

Context

AccessKey supports the following authorization methods:

- Use RAM to authorize the AccessKey ID and AccessKey secret to third-party requesters.
- Use STS to authorize the AccessKey ID, AccessKey secret, and security token to third-party requesters.

This topic describes how to obtain an AccessKey pair authorized by RAM. For more information about how to obtain an AccessKey pair authorized by STS, see *Obtain an STS AccessKey pair*.

We recommend that you use the AccessKey pair of a personal account to call the API operations of the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, configure the request headers that are described in the following table for access control. Otherwise, an error that indicates insufficient permissions can occur.

Parameter	Description
x-acs-regionid	The ID of the region, such as cn-hangzhou-*.
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For information about how to obtain an organization ID, see the "GetOrganizationList" section in Apsara Uni-manager Management Console Developer Guide.
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For information about how to obtain a resource set ID, see the "ListResourceGroup" section in Apsara Uni-manager Management Console Developer Guide.
x-acs-instanceid	The ID of the instance on which to perform operations. For example, you can query the ID of an Elastic Compute Service (ECS) instance from the ECS instance list.

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

Obtain the AccessKey pair of a personal account

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

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and choose **User Information** from the shortcut menu.
3. In the **Apsara Stack AccessKey Pair** section, obtain the AccessKey pair.

AccessKey					Create AccessKey Pair
The AccessKey ID and AccessKey Secret are the keys used to access the cloud service resources. They have full permissions on the account. Please keep them properly.					
AccessKey ID	AccessKey Secret	Status	Created At	Actions	
W1YRFvKH*****	*****	Enable	Sep 18, 2021, 11:05:57	Disable	Delete
IsJsAUyX*****	*****	Enable	Sep 10, 2021, 12:28:35	Disable	Delete

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

Obtain the AccessKey pair of an organization

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

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization navigation tree, click the name of the level-1 organization for which you want to obtain AccessKey pairs.
5. Click **Management AccessKey** on the right side of the page.
6. In the AccessKey message, view the AccessKey pair of the organization.

3.1.1.4. Obtain common header parameters

You must provide multiple header parameters when you call API operations. This topic describes how to obtain common header parameters. Some response parameters are also explained.

Query a region ID

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

Obtain an organization ID

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

Obtain a resource set ID

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

Obtain an instance ID

You can obtain the ID of an instance in the cloud service console of the Apsara Uni-manager Management Console. This topic illustrates how to obtain the ID of an ECS instance. Perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
3. In the left-side navigation pane, choose **Instances & Images > Instances**.
4. View the **instance ID** in the instance list.

Response parameters

The following response parameters are returned by default when you call list-type API operations for cloud product resource instances.

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

3.1.1.5. Obtain an STS AccessKey pair

This topic describes how to use STS to make API calls.

Background information

Alibaba Cloud Security Token Service (STS) allows you to manage temporary credentials to your Alibaba Cloud resources. You can use STS to make API calls.

Procedure

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

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

2. Use the following code to obtain `stsAK` , `stsSK` , and `stsToken` by using your `RAM role` .

```
public static void main(String[] args) {
    ASClient asClient = new ASClient();
    // The call source.
    asClient.setSdkSource("asapi_test");
    // Pass in the ARN.
    String assumeRole = AssumeRole(asClient, "<RAM Role>");

    JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);

    JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");

    // Obtain the STS triplet.
    String stsAK = credentials.getString("AccessKeyId");
    String stsSK = credentials.getString("AccessKeySecret");
    String stsToken = credentials.getString("SecurityToken");
}

public static String AssumeRole(ASClient asClient, String roleArn) {

    Map<String, String> headers = new HashMap<>();

    Map<String, Object> reqParams = new HashMap<>();

    reqParams.put("action", "AssumeRole");

    reqParams.put("Product", "Sts");

    reqParams.put("Version", "2015-04-01");

    reqParams.put("regionId", regionId);

    // Your AccessKey ID.
    reqParams.put("AccessKeyId", <Your AccessKey ID>);
    // Your AccessKey secret.
    reqParams.put("AccessKeySecret", <Your AccessKey secret>);

    reqParams.put("roleSessionName", "AsapiStsUser");

    reqParams.put("roleArn", roleArn);

    return asClient.doRequest(endpoint, headers, reqParams);
}
```

3. You can use STS to make API calls. The following sample code demonstrates how to call ECS API operations.


```
public static void main(String[] args) {
    ASClient asClient = new ASClient();
    asClient.setSdkSource("asapi_test");
    // Call ECS API operations.
    String ecsRes = DescribeZones(asClient,<stsAK>,<stsSK>,<stsToken>);
}

public static String DescribeZones(ASClient asClient, String stsAK, String stsSK, String stsToken) {

    Map<String, String> headers = new HashMap<>();

    Map<String, Object> reqParams = new HashMap<>();

    reqParams.put("action", "DescribeZones");

    reqParams.put("Product", "Ecs");

    reqParams.put("Version", "2014-05-26");

    reqParams.put("regionId", "<your-RegionID>");

    // The STS triplet of AccessKey ID, AccessKey secret, and security token.
    reqParams.put("AccessKeyId", <stsAK>);
    reqParams.put("AccessKeySecret", <stsSK>);
    reqParams.put("SecurityToken", <stsToken>);

    return asClient.doRequest(endpoint, headers, reqParams, HttpMethod.GET);
}
```

3.1.2. Use POP to make API calls

3.1.2.1. Preparations

Like in the public cloud, you can use the POP gateway to call API operations in Apsara Stack. This topic describes the preparations before you use the POP gateway to call API operations.

Background information

In Apsara Stack, you can use both ASAPI and POP gateways to call API operations. To use the same method and produce the same user experience as in the public cloud, we recommend that you use the POP gateway to call API operations in Apsara Stack.

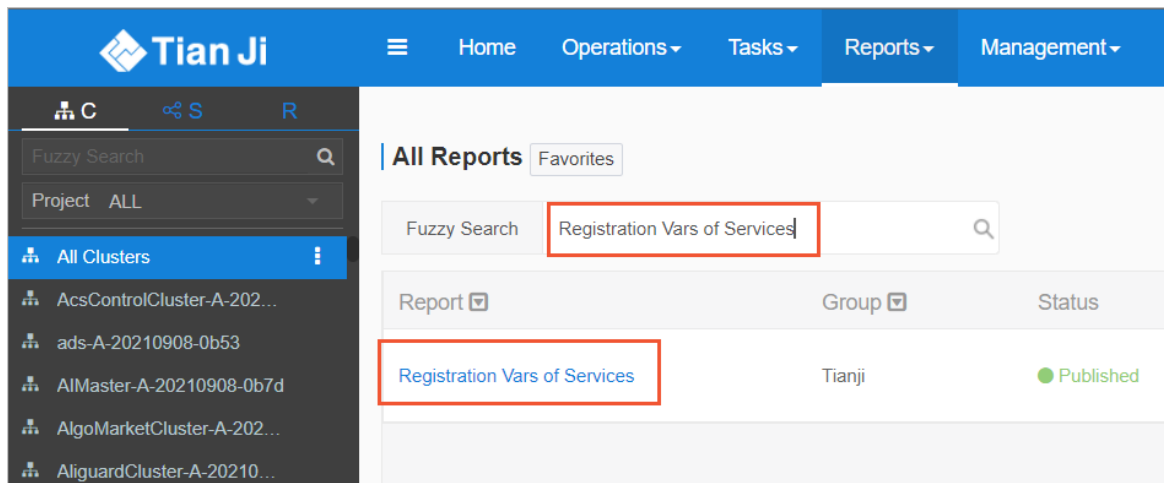
Obtain service endpoints

To use the POP gateway to call API operations, you must use the service endpoints. You can obtain the service endpoints from Alibaba Cloud onsite O&M engineers.

You can also obtain the service endpoints by yourself. The following example describes how to obtain the ECS endpoints.

1. Log on to Apsara Infrastructure Management Framework.

- i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, click **O&M**.
 - iii. In the left-side navigation pane, choose **Product Management > Products**.
 - iv. In the Apsara Stack O&M section, click **Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.
 3. On the **All Reports** page, search for **Registration Vars of Services** and click the report name.



4. On the **Registration Vars of Services** page, click the  icon next to **Service** and search for **ECS**.
5. Find the **ecs-yaochi** service, right-click the **Service Registration** column, and then select **Show More** from the shortcut menu.

On the **Details** page, view the **internal.openapi.endpoint** value, which is the **endpoint**.

Obtain common parameters

You must obtain the parameters required when you use the POP gateway to call API operations, such as the organization ID, resource set ID, region ID, and instance ID. For more information, see [Obtain common header parameters](#).

3.1.2.2. SDK for Java

Before you use POP SDK for Java to make API calls, you must first understand the RPC and ROA styles.

Prerequisites

- An **Alibaba Cloud SDK** package is downloaded for the service that you want to access. The required dependencies are installed.
- SSL certificate validation is ignored for requests over HTTPS.

Call examples

The following sample code provides an example on how to use the RPC style to make an API call. In this example, the DescribeInstances operation of Elastic Compute Service (ECS) is called, and the API version is 2014-05-26.

Sample code:

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class DescribeInstances {

    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile("<your-regionid>", "<yourAccessKeyId>", "<yourAccessSecret>");

        // Ignore SSL certificate validation.
        // HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        // clientConfig.setIgnoreSSLCerts(true);
        // Configure the time-out period for a connection.
        // clientConfig.setConnectionTimeoutMillis(3000L);
        // Set the read timeout period.
        // clientConfig.setReadTimeoutMillis(10000L);
        // profile.setHttpClientConfig(clientConfig);

        IAcsClient client = new DefaultAcsClient(profile);
        // Create an API request and configure the parameters.
        CommonRequest request = new CommonRequest();
        // Specify the request method.
        request.setSysMethod(MethodType.POST);
        // Set the service endpoint.
        request.setSysDomain("<ecsEndpoint>");
        // Specify the version of the API operation.
        request.setSysVersion("2014-05-26");
        // Specify the name of the service that you want to access and the region where the service resides.
        request.setSysLocationProduct("Ecs");
        // Specify the name of the API operation.
        request.setSysAction("DescribeInstances");

        // When you use the STS AccessKey ID to call RPC API operations, you must add the SecurityToken parameter to the header.
        // request.putBodyParameter("SecurityToken", "<your-SecurityToken>");

        // Configure operation-specific parameters.
        request.putBodyParameter("PageSize", "10");
        request.putBodyParameter("PageNumber", "1");

        // Required. The call source.
        request.putHeaderParameter("x-acs-caller-sdk-source", "AniTestDemo");
    }
}
```

```

request.putHeaderParameter("x-acf-external-source", "API-Test-Demo");
// Optional. For the historical versions before Enterprise Edition V3.15.0.
request.putHeaderParameter("x-acf-regionid", "<your-regionid>");
// Optional. The resource set ID.
request.putHeaderParameter("x-acf-resourcegroupid", "<your-resourcegroupid>");
// Required. The organization ID.
request.putHeaderParameter("x-acf-organizationid", "<your-organizationid>");
// Optional. For the modify and delete operations.
// request.putHeaderParameter("x-acf-instanceid", "<your-instanceid>");

try {
    // Initiate the request and handle the response or exceptions.
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    System.out.println("ErrCode:" + e.getErrCode());
    System.out.println("ErrMsg:" + e.getErrMsg());
    System.out.println("RequestId:" + e.getRequestId());
}
}
}

```

The following sample code provides an example on how to use the ROA style to make an API call. In this example, the GetNamespaceList operation of Container Registry is called, and the API version is 2016-06-07.

Sample code:

```

import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class GetNamespaceList {

    public static void main(String[] args) {
        // Create and initialize a DefaultAcsClient instance.
        DefaultProfile profile = DefaultProfile.getProfile("<your-regionid>", "<yourAccessKeyId>", "<yourAccessSecret>");

        // Ignore SSL certificate validation.
        // HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        // clientConfig.setIgnoreSSLCerts(true);
        // Configure the time-out period for a connection.
        // clientConfig.setConnectionTimeoutMillis(3000L);
        // Set the read timeout period.
        // clientConfig.setReadTimeoutMillis(10000L);
        // profile.setHttpClientConfig(clientConfig);
    }
}

```

```
// Create an API request and configure the parameters.
CommonRequest request = new CommonRequest();

// Set the request protocol to HTTPS.
request.setSysProtocol(ProtocolType.HTTPS);
// Specify the request method.
request.setSysMethod(MethodType.GET);
// Set the service endpoint.
request.setSysDomain("<crEndpoint>");
// Specify the version of the API operation.
request.setSysVersion("2016-06-07");
// Configure the uriPattern parameter, which is required for the ROA style.
request.setSysUriPattern("/namespace");
// Specify the name of the service that you want to access and the region where the service resides.
request.setSysLocationProduct("cr");
// Specify the name of the API operation.
request.setSysAction("GetNamespaceList");

// Required. The call source.
request.putHeadParameter("x-acs-caller-sdk-source", "ApiTestDemo");
// Optional. For the historical versions before Enterprise Edition V3.15.0.
request.putHeadParameter("x-acs-regionid", "<your-regionid>");
// Optional. The resource set ID.
request.putHeadParameter("x-acs-resourcegroupid", "<your-resourcegroupid>");
// Required. The organization ID.
request.putHeadParameter("x-acs-organizationid", "<your-organizationid>");
// Optional. For the modify and delete operations.
// request.putHeadParameter("x-acs-instanceid", "<your-instanceid>");

// When you use the STS AccessKey ID to call ROA API operations, you must add the x-acs-security-token
parameter to the header.
// request.putHeadParameter("x-acs-security-token", "<your-securitytoken>");

try {
    // Initiate the request and handle the response or exceptions.
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    System.out.println("ErrCode:" + e.getErrCode());
    System.out.println("ErrMsg:" + e.getErrMsg());
    System.out.println("RequestId:" + e.getRequestId());
}
}
```

 Notice

If you send a request over HTTPS, you must enable the request to ignore SSL certificate validation.

3.1.2.3. SDK for Go

This topic describes how to use POP SDK for Go.

Prerequisites

An **Alibaba Cloud SDK** package is downloaded for the service that you want to access. The required dependencies are installed.

Background information

Both RPC and ROA styles are supported for API calls.

Call examples

The following sample code provides an example on how to use the RPC style to make an API call. In this example, the DescribeInstances operation of Elastic Compute Service (ECS) is called, and the API version is 2014-05-26.

Sample code:

```
package main

import (
    "fmt"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
)

func main() {
    client, err := sdk.NewClientWithAccessKey("<your-RegionID>", "<AccessKey ID>", "<AccessKey Secret>")

    request := requests.NewCommonRequest()

    request.Method = "POST"
    request.Domain = "<ECSEndpoint>"
    request.Version = "2014-05-26"
    request.ApiName = "DescribeInstances"
    request.Product = "Ecs"
    // When you use the STS AccessKey ID to call RPC API operations, you must add the SecurityToken parameter to the header.
    // request.QueryParams["SecurityToken"] = "<your-SecurityToken>"

    // Required. The call source.
    request.Headers["x-acs-caller-sdk-source"] = "ApiTestDemo"
    // Optional. For the historical versions before Enterprise Edition V3.15.0.
    request.Headers["x-acs-regionid"] = "<your-RegionID>"
    // Optional. The resource set ID.
    request.Headers["x-acs-resourcegroupid"] = "<your-ResourceGroupID>"
    // Required. The organization ID.
    request.Headers["x-acs-organizationid"] = "<your-OrganizationID>"
    // Optional. For the modify and delete operations.
    // request.Headers["x-acs-instanceid"] = "<your-InstanceID>"

    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Print(response.GetHttpContentString())
}
```

The following sample code provides an example on how to use the ROA style to make an API call. In this example, the GetNamespaceList operation of Container Registry is called, and the API version is 2016-06-07.

The following sample code shows how to call API operations:

```
package main

import (
    "fmt"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "os"
)

func main() {
    client, err := sdk.NewClientWithAccessKey("<your-regionid>", "<yourAccessKeyId>", "<yourAccessSecret>")
    request := requests.NewCommonRequest()

    request.Method = "GET"
    request.Domain = "<crEndpoint>"
    request.Version = "2016-06-07"
    request.ApiName = "GetNamespaceList"
    request.Product = "cr"

    request.PathPattern = "/namespace"
    // When you use the STS AccessKey ID to call ROA API operations, you must add the x-acs-security-token parameter to the header.
    // request.Headers["x-acs-security-token"] = "<your-SecurityToken>"
    // Required. The call source.
    request.Headers["x-acs-caller-sdk-source"] = "ApiTestDemo"
    // Optional. For the historical versions before Enterprise Edition V3.15.0.
    request.Headers["x-acs-regionid"] = "<your-RegionID>"
    // Optional. The resource set ID.
    request.Headers["x-acs-resourcegroupid"] = "<your-ResourceGroupID>"
    // Required. The organization ID.
    request.Headers["x-acs-organizationid"] = "<your-OrganizationID>"
    // Optional. For the modify and delete operations.
    // request.Headers["x-acs-instanceid"] = "<your-InstanceID>"

    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Print(response.GetHttpContentString())
}
```

Configure proxies

You can configure proxies in one of the following ways when you call API operations:

- Configure proxies as environment variables.

Set the HTTP_PROXY and HTTPS_PROXY environment variables, or set the NO_PROXY environment variable.

Sample code:


```
import (
    "os"
)
func main(){
    os.Setenv("HTTP_PROXY", "http://x.x.x.x:xxxx") // Configure the HTTP_PROXY environment variable.
    os.Setenv("HTTPS_PROXY", "https://x.x.x.x:xxxx") // Configure the HTTPS_PROXY environment variable.
}
```

- Configure proxies on the client.

Sample code:

```
// The proxies that are configured on the client take precedence over those that are configured by using environment variables.
client.SetHttpProxy("http://x.x.x.x:xxxx") // Configure the HTTP proxy.
client.GetHttpProxy() // Query the HTTP proxy.

client.SetHttpsProxy("https://x.x.x.x:xxxx") // Configure the HTTPS proxy.
client.GetHttpsProxy() // Query the HTTPS proxy.

client.SetNoProxy("127.0.0.1,localhost") // Configure the proxy whitelist.
client.GetNoProxy() // Query the proxy whitelist.
```

3.1.2.4. SDK for Python

This topic describes how to use POP SDK for Python.

Prerequisites

An [Alibaba Cloud SDK](#) package is downloaded for the service that you want to access. The required dependencies are installed.

Background information

Only the remote procedure call (RPC) style is supported for API calls.

Call examples

The following sample code provides an example on how to make an API call. In this example, the DescribeInstances operation of Elastic Compute Service (ECS) is called, and the API version is 2014-05-26.

Sample code:

```
from aliyunsdkcore.client import AcsClient
import os
from aliyunsdkcore.request import CommonRequest

# Create an AcsClient object.
client = AcsClient(
    "<AccessKey ID>",
    "<AccessKey Secret>",
    "<your-RegionID>",
    timeout=1000, # Set the timeout period.
    verify=False, # Ignore certificate validation. or verify='<The location where the certificate is stored.>'
)
```

```

);
# Set the following parameters for the HTTPS request protocol:
# client.set_verify(False)
# os.environ["HTTPS_PROXY"] = "http://127.0.0.1:8080"

# os.environ["HTTP_PROXY"] = "http://127.0.0.1:8080"
def hello_ecs_instance():
    # Create a request and set the parameters.
    request = CommonRequest()

    # When you use the STS AccessKey ID to call RPC API operations, you must add the SecurityToken parameter to the header.
    # params = {"PageSize": "10", "PageNumber": "1", "SecurityToken": "<your-SecurityToken>"};
    params = {"PageSize": "10", "PageNumber": "1"};
    request.set_body_params(params)

    headers = {
        # Required. The call source.
        "x-acs-caller-sdk-source": "ApiTestDemo",
        # Optional. For the historical versions before Enterprise Edition V3.15.0.
        "x-acs-regionid": "<your-RegionID>",
        # Optional. The resource set ID.
        "x-acs-resourcegroupid": "<your-ResourceGroupID>",
        # Required. The organization ID.
        "x-acs-organizationid": "<your-OrganizationID>",
        # Optional. For the modify and delete operations.
        "x-acs-instanceid": "<your-InstanceID>"
    }
    request.set_headers(headers)
    request.set_version("2014-05-26")
    request.set_action_name("DescribeInstances")
    request.set_endpoint("<ecsEndpoint>") // Replace ecsEndpoint with the real endpoint.
    request.set_product("Ecs")
    request.set_method("POST")

    # Initiate the API request and obtain the response.
    response = client.do_action_with_exception(request)
    print(response)

if __name__ == '__main__':
    print("Hello Aliyun OpenApi!")
    hello_ecs_instance()

```

Configure proxies

You can configure HTTP_PROXY and HTTPS_PROXY environment variables when you call API operations.

Sample code:

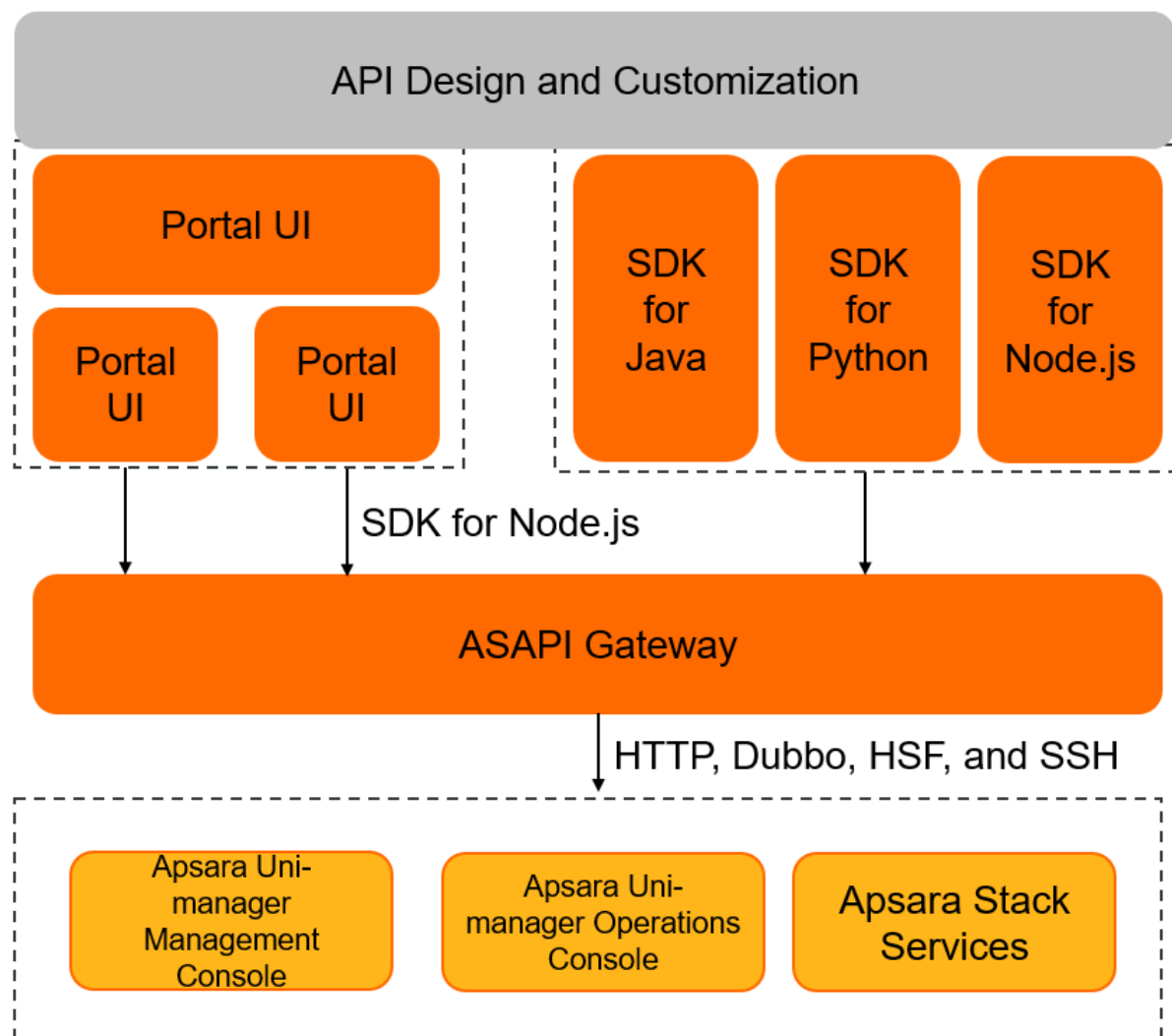
```
# Configure the HTTPS proxy.  
os.environ["HTTPS_PROXY"] = "http://x.x.x.x:xxxx"  
  
# Configure the HTTP proxy.  
os.environ["HTTP_PROXY"] = "https://x.x.x.x:xxxx"
```

3.1.3. Use ASAPI to make API calls

3.1.3.1. ASAPI overview

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

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



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

- Make API calls to implement enterprise-grade management. For example, you can manage

organizations, users, roles, resource sets, and logon policies. These operations are provided by the Apsara Uni-manager Management Console.

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

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

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

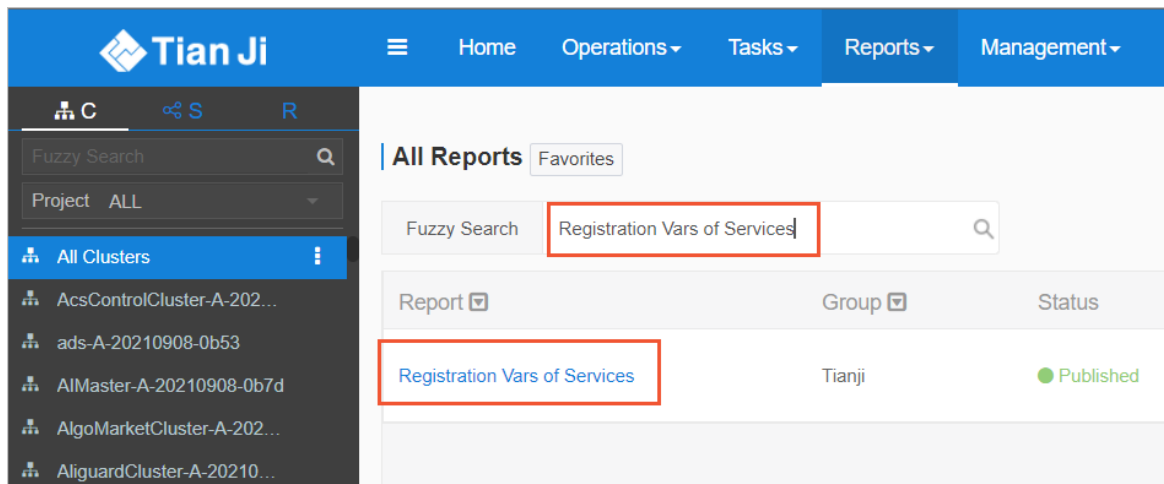
3.1.3.2. Generate an endpoint for ASAPI

This topic describes three methods used to generate an endpoint for Apsara Stack API (ASAPI).

Method 1: Use Registration Vars of Services to generate an endpoint

1. Log on to Apsara Infrastructure Management Framework.
 - i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, click **O&M**.
 - iii. In the left-side navigation pane, choose **Product Management > Products**.
 - iv. In the Apsara Stack O&M section, click **Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.

- On the **All Reports** page, search for **Registration Vars of Services** and click the report name.



- On the **Registration Vars of Services** page, click the  icon next to **Service** and search for **asapi**.
- Find the **asapi** service, right-click in the **Service Registration** column, and then select **Show More** from the shortcut menu.
In the **Details** dialog box, obtain the value of **asapi.public.endpoint**.
- Generate an endpoint for ASAPI based on the following format:

```
https://<asapi.public.endpoint>/asapi/v3
```

Method 2: Use cluster resources to generate an endpoint

- Log on to Apsara Infrastructure Management Framework.
 - Log on to the Apsara Uni-manager Operations Console.
 - In the top navigation bar, click **O&M**.
 - In the left-side navigation pane, choose **Product Management > Products**.
 - In the Apsara Stack O&M section, click **Apsara Infrastructure Management Framework**.
- Go to the cluster operations page.
 - In the left-side navigation pane, choose **Operations > Cluster Operations**.
 - In the **Cluster** search box, enter **ascm**.
 - Find the cluster that you want to manage and click **Operations** in the **Actions** column.
- Obtain the value of the domain parameter.
 - Click the **Cluster Resources** tab.
 - In the **Name** field, search for **asapi_dns_public**.

- iii. Click **Details** in the **Application Parameter** column.

In the **Application Parameter** message, obtain the value of the **domain** parameter.

```
{
  "domain": "public.asapi.<region>.<internet-domain>.com",
  "ha_apply_type": "active-active",
  "name": "asapi_dns_public",
  "vip": "10.10.10.10"
}
```

4. Generate an endpoint for ASAPI based on the following format:

```
https://<domain>/asapi/v3
```

Method 3: Concatenate strings to generate an endpoint

You can also concatenate strings to generate an endpoint based on the following format:

```
https://public.asapi.<region>.<internet-domain>/asapi/v3
```

Note

1. Replace <region> with your actual region ID. For information about how to query a region ID, see [Query a region ID](#).
2. Replace <internet-domain> with the root domain of your Apsara Stack service. To obtain the root domain, contact Apsara Stack O&M personnel. Take note of this point: Use the internet-domain parameter instead of the intranet-domain parameter.

If the **region** value is `cn-****-****-d01` and the **internet-domain** value is `inter.****.****.com`, the following endpoint can be generated:

An example of a complete endpoint:

```
https://public.asapi.cn-****-****-d01.inter.****.****.com/asapi/v3
```

3.1.3.3. Obtain Apsara Stack SDKs

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

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

3.1.3.4. Common request parameters

Common request parameters must be included in all API requests.

Sample common parameters

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

Parameter	Type	Required	Description
Action	String	Yes	The operation that you want to perform. For more information, see View the information about API operations .
Product	String	Yes	The name of the service.
Version	String	Yes	The version of the API operation. For more information, see View information about API operations .
RegionId	String	Yes	The ID of the region. For more information, see Query a region ID .
AccessKeyId	String	Yes	The AccessKey ID used to connect to the service. For more information, see Obtain an AccessKey pair .
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. Valid values: HMAC-SHA1 and HMAC-SHA256.

Parameter	Type	Required	Description
SignatureVersion	String	Yes	The version of the signature encryption algorithm. Valid values: 1.0 and 2.0.
SignatureNonce	String	Yes	A unique, random number used to prevent replay attacks. You must use different numbers for different requests.
Timestamp	String	Yes	The timestamp of the request. Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC. Example: 2018-01-01T12:00:00Z.
Format	String	Yes	The format in which to return the response. Set the value to JSON.

Note

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

3.1.3.5. SDK description

3.1.3.5.1. SDK for Java

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

Prerequisites

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

- Apsara Stack SDK for Java 1.8 or later is installed.

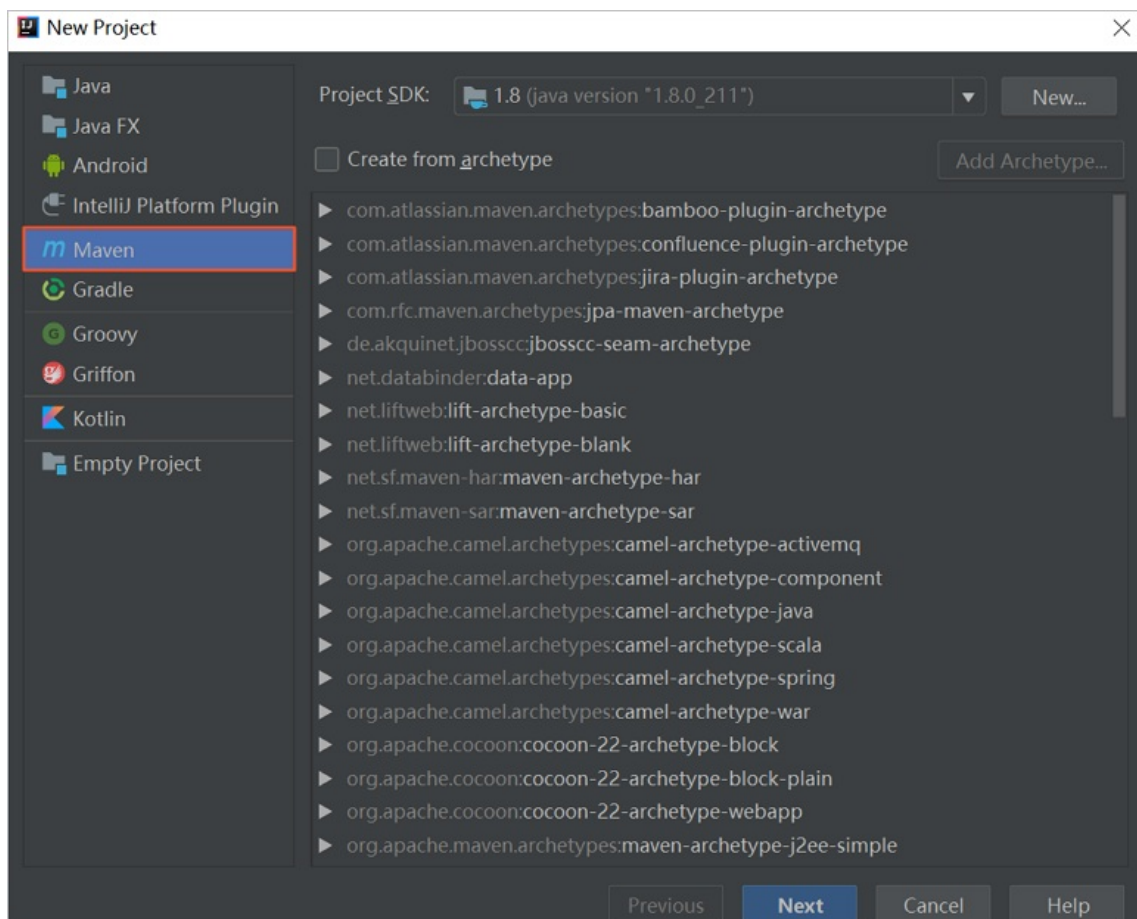
Background information

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

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

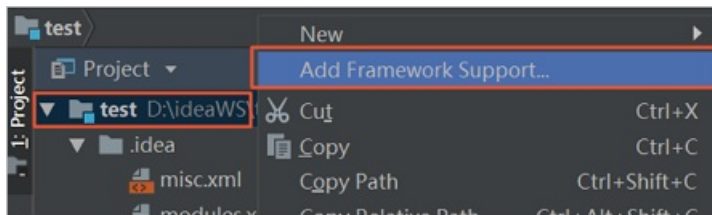
Procedure

1. Use one of the following methods to configure Maven in IntelliJ IDEA. Maven is a tool used for project management.
 - Use Maven that is integrated in IntelliJ IDEA.
 - Download Maven based on the OS that you are using from the official website and configure Maven. For more information, see [Download Maven](#).
2. Obtain the Maven dependencies for Apsara Stack SDK for Java from the official website.
3. Use one of the following methods to create a Maven project:
 - Method 1: Create a Maven project in IntelliJ IDEA.

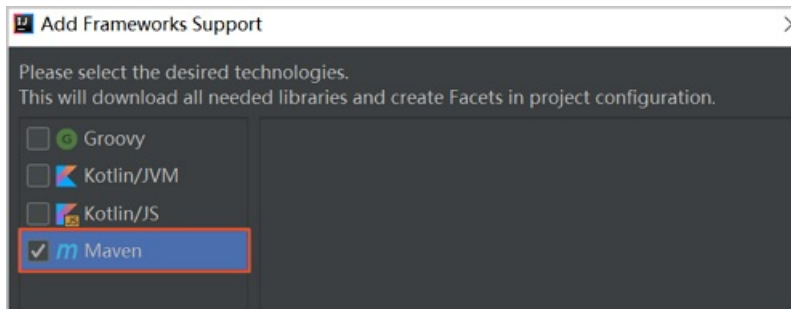


- Method 2: Convert an existing project to a Maven project.

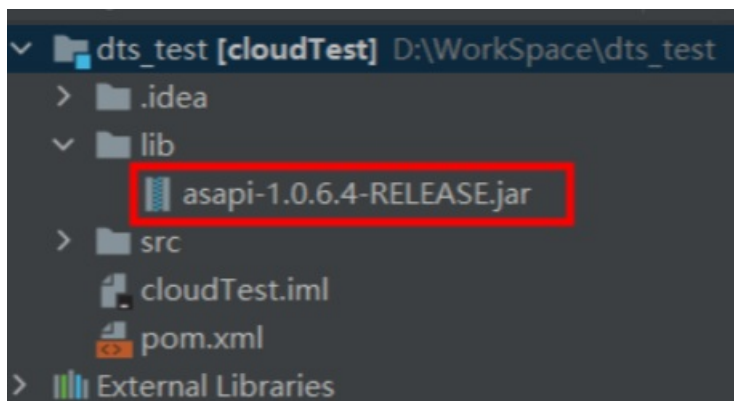
- a. Right-click the project that you want to convert and select **Add Framework Support....**



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



4. Create a folder named **libs** in the root directory of the project. Then, add the JAR file of Apsara Stack SDK for Java to the **libs** folder, as shown in the following figure.



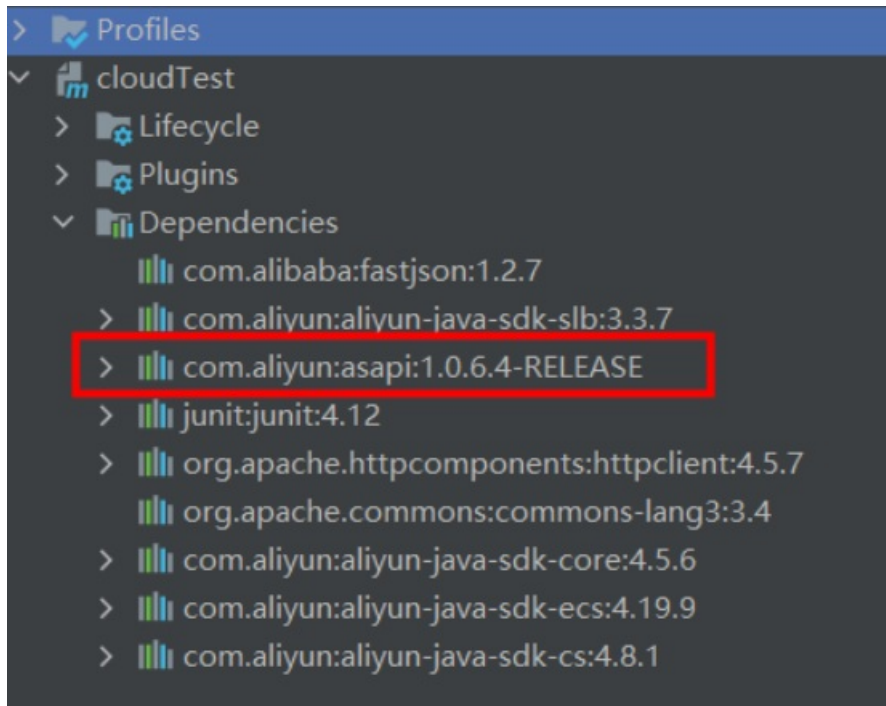
5. Add the dependencies of Apsara Stack SDK for Java to the **pom.xml** file in the project directory.

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>asapi</artifactId>
    <version>1.0.6.4-RELEASE</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/libs/asapi-1.0.6.4-RELEASE.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>alicrypto-java-aliyun</artifactId>
    <version>1.0.4</version>
    <systemPath>${project.basedir}/libs/alicrypto-java-aliyun-1.0.4.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>gmsse</artifactId>
    <version>1.1.0</version>
    <systemPath>${project.basedir}/libs/gmsse-1.1.0.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.7</version>
    <systemPath>${project.basedir}/libs/httpclient-4.5.7.jar</systemPath>
  </dependency>
</dependencies>
```

 **Note** Replace `1.0.6.4-RELEASE` with the actual version of Apsara Stack SDK for Java.

6. On the right side of IntelliJ IDEA, view information in the Maven section.

If the information in the following figure appears, the dependencies are added.



Use Apsara Stack SDK for Java

You can use the GET or POST method to call API operations provided by ASAPI. The following sample code shows how to call API operations:

```
// 1. Create a connection to an ASClient object.
ASClient client = new ASClient();
// 2. Configure the identifier of the requester. This parameter is required and used only for identification. You can set this parameter to a random value.
client.setSdkSource("<The identifier of the requester>");
Map<String, Object> parameters = new HashMap<String, Object>();
// 3. Configure the parameters for authorization. The parameter values vary based on your environment.
// The AccessKey ID of your Apsara Stack tenant account.
parameters.put("AccessKeyId", "<AccessKey ID>");
// The AccessKey secret of your Apsara Stack tenant account.
parameters.put("AccessKeySecret", "<AccessKey Secret>");
// The ID of the region.
parameters.put("RegionId", "<your RegionID>");
// 4. Configure parameters for an API call. In this example, the Product, Action, and Version parameters are configured.
parameters.put("Product", "<The name of the service>");
parameters.put("Action", "<The name of the API operation>");
parameters.put("Version", "<The version of the API operation>");
// When you use the STS AccessKey ID to call RPC API operations, you must add the SecurityToken parameter to the header.
// parameters.put("SecurityToken", "<your-securitytoken>");
// When you use the STS AccessKey ID to call ROA API operations, you must add the x-acis-security-token parameter to the header.
// headers.put("x-acis-security-token", "<your-securitytoken>");
// 5. Configure operation-specific parameters.
parameters.put("<The name of the operation-specific parameter>", "<The value of the operation-specific parameter>");
// 6. Configure request headers and the encryption method for the signature string. Valid values of the SignatureMethod parameter: HMAC-SHA1 and HMAC-SHA256. By default, HMAC-SHA256 is used.
Map<String, String> headers = new HashMap<String, String>();
//headers.put("SignatureMethod", "HMAC-SHA1");
headers.put("x-acis-regionid", "<your-regionid>");
headers.put("x-acis-resourcegroupid", "<your-resourcegroupid>");
headers.put("x-acis-organizationid", "<your-organizationid>");
headers.put("x-acis-instanceid", "<your-instanceid>");
// 7. Make an API call. We recommend that you use the doPost() method.
// Configure the time-out period for a connection.
//client.setConnectTimeout(10000);
// Configure the time-out period for a request.
// client.setSocketTimeout(10000);
String result = client.doPost(endpoint, headers, parameters);
System.out.println(result);
```

Configure proxies

If you need to use a proxy to make API calls, use the following sample code to configure the proxy:

```
Cloud cloud = new Cloud();
// Configure the IP address of the proxy.
cloud.setConfig("proxy_host", "<IP>");
// Configure the port used by the proxy.
cloud.setConfig("proxy_port", "<Port>");
Create a connection to an ASClient object.
ASClient client = new ASClient(cloud);
```

Header parameters

Parameter	Description
x-acs-regionid	The ID of the region. For more information about how to obtain a region ID, see Query a region ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console.
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console.
x-acs-instanceid	The ID of the instance to which you make API calls.

3.1.3.5.2. SDK for Python

This topic describes how to obtain and use Apsara Stack SDK for Python to call API operations.

Preparations

- An authorized account and an AccessKey pair are obtained. The AccessKey pair consists of an AccessKey ID and an AccessKey secret. For more information about how to obtain an AccessKey pair in the Apsara Uni-manager Management Console, see [Obtain an AccessKey pair](#).
- The endpoint of ASAPI is obtained. For more information, see [Generate an endpoint for ASAPI](#).

Install Apsara Stack SDK for Python

- Download the required version of Apsara Stack SDK for Python. For more information, see [Obtain Apsara Stack SDKs](#).
- Upload the compressed file of Apsara Stack SDK for Python to the host from which you want to send API requests and extract the file. Then, run the following code in the directory of the extracted files to install Apsara Stack SDK for Python:

```
# Install Apsara Stack SDK for Python 2.x.
pip install aliyun-python2-sdk-asapi-2.4.7.tar.gz
# Install Apsara Stack SDK for Python 3.x.
pip3 install aliyun-python3-sdk-asapi-2.4.7.tar.gz
```

 Note

Replace `2.4.7` with the actual version of Apsara Stack SDK for Python.

Sample requests

You can use the GET or POST method to call API operations provided by ASAPI.

```
# 1. Import dependencies.
from aliyunsdkasapi.ASClient import ASClient
from aliyunsdkasapi.AsapiRequest import AsapiRequest

if __name__ == '__main__':
    # 2. Create a request. The Product parameter specifies the name of the service. The Version parameter specifies the version of the API operation. The Action parameter specifies the operation that you want to perform. The asapi-endpoint parameter specifies the endpoint of ASAPI.
    req = AsapiRequest("<Product>", "<Version>", "<Action>", "<asapi-endpoint>")
    # 3. Configure operation-specific parameters.
    req.add_body_params("<key>", "<value>")
    # When you use the STS AccessKey ID to call RPC API operations, you must add the SecurityToken parameter to the header.
    # req.add_body_params("SecurityToken", "<your-securitytoken>");
    # When you use the STS AccessKey ID to call ROA API operations, you must add the x-acis-security-token parameter to the header.
    # req.add_header("x-acis-security-token", "<your-securitytoken>");
    # 4. Configure request headers.
    req.add_header("x-acis-resourcegroupid", "<your-resourcegroupid>");
    req.add_header("x-acis-organizationid", "<your-organizationid>");
    req.add_header("x-acis-regionid", "<your-regionid>");
    req.add_header("x-acis-instanceid", "<your-instanceid>");
    # 5. Configure a request method.
    req.set_method("POST")
    # 6. Configure initialization parameters for an ASClient object. The parameter values vary based on your environment. The timeout parameter specifies the time-out period for a request. The cert_file parameter specifies the certificate. The verify parameter specifies whether to verify the certificate.
    as_client = ASClient("<accesskeyid>", "<accesskeysecret>", "<regionid>", timeout=1000,
        cert_file=None,
        verify=False)
    # Configure the identifier of the requester. This parameter is required and used only for identification. You can set this parameter to a random value.
    as_client.setSdkSource("<The identifier of the requester>")
    response = as_client.do_request(req)
    print(response)
```

Header parameters

Parameter	Description

Parameter	Description
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console.
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console.
x-acs-regionid	The ID of the region. For more information about how to obtain a region ID, see Query a region ID .
x-acs-instanceid	The ID of the instance on which you want to perform operations.

Configure proxies

Perform the following steps to configure a proxy:

1. Import OS dependencies.

```
import os
```

2. Configure one or more proxies.

```
# Before you make an API call, configure one or more proxies.
os.environ['http_proxy'] = "http://x.x.x.x:xxxx"
os.environ['https_proxy'] = "https://x.x.x.x:xxxx"
```

3.1.3.5.3. SDK for Node.js

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

Preparations

- An authorized account and an AccessKey pair are obtained. The AccessKey pair consists of an AccessKey ID and an AccessKey secret. For more information about how to obtain an AccessKey pair in the Apsara Uni-manager Management Console, see [Obtain an AccessKey pair](#).
- The endpoint of Apsara Stack API (ASAPI) is obtained. For more information, see [Generate an endpoint for the ASAPI](#).
- For more information about how to obtain the region ID, see [Obtain the endpoint of ASAPI](#).

Add dependencies

Run the following command to install Apsara Stack SDK for Node.js:

```
tnpm i @ali/asapi-nodejs-sdk@0.0.33 --save
```

Use Apsara Stack SDK for Node.js

The following sample code shows how to use Apsara Stack SDK for Node.js to make API calls:

```
// 1. Import the dependencies of Apsara Stack SDK for Node.js.
const Controller = require('egg').Controller;
const ASAPIClient = require('@ali/asapi-nodejs-sdk');
// 2. Configure the parameters.
const cnofig = {
  API_GATEWAY: '<The endpoint of ASAPI>',
  regionId: '<The ID of the region in which the Apsara Stack service is deployed>',
  accessKeyId: '<The AccessKey ID of your Apsara Stack tenant account>',
  accessKeySecret: '<The AccessKey secret of your Apsara Stack tenant account>'
};
// 3. Initialize a client for API calls.
const AsapiClient = new ASAPIClient(cnofig, this.ctx);
// 4. Configure request parameters.
const params = {
  "Action": "<The name of the API operation>",
  "Product": "<The name of the service that provides the API operation>",
  "Version": "<The version of the API operation>",
  "regionId": '<The ID of the region in which the Apsara Stack service is deployed>',
  // When you use the STS AccessKey ID to call RPC API operations, you must add the SecurityToken parameter to the header.
  // "SecurityToken": "<your-securitytoken>",
  "<The name of the operation-specific parameter required by the API operation>": "<The value of the operation-specific parameter>"
}
// 5. Configure request headers.
// When you use the STS AccessKey ID to call ROA API operations, you must add the x-acis-security-token parameter to the header.
// AsapiClient.addHeader("x-acis-security-token", "<your-securitytoken>")
AsapiClient.addHeader('x-acis-caller-sdk-source', '<The identifier of the requester>')
AsapiClient.addHeader('x-acis-roleid', '<The ID of the role>')
AsapiClient.addHeader('x-acis-userid', '<The ID of the user>')
AsapiClient.addHeader('x-acis-organizationid', '<your-organizationid>')
AsapiClient.addHeader("x-acis-regionid", "<your-regionid>");
AsapiClient.addHeader("x-acis-resourcegroupid", "<your-resourcegroupid>");
AsapiClient.addHeader("x-acis-instanceid", "<your-instanceid>");
// 6. Make an API call and obtain the response.
try {
  const res = await AsapiClient.doRequest(cnofig.API_GATEWAY, params, {
    method: 'POST', // The request method.
    formatParams: false, // Specifies whether to format parameters and capitalize the first letter of each parameter.
    enableProxy: true, // Specifies whether to configure a proxy.
    proxy: 'http://x.x.x.x:xxxx',
    timeout: 10000, // The time-out period.
    // ca: , // The certificate issued by a certificate authority (CA). For more information, visit https://nodejs.org/api/https.html#https_https_request_options_callback.
  })
  console.log(res)
} catch (e) {
  console.log(e)
}
```

Disable certificate validation

The following sample code shows how to disable certificate validation:

```
Configure the environment variable.  
NODE_TLS_REJECT_UNAUTHORIZED=0
```

Request headers

Parameter	Description
x-acs-regionid	The ID of the region. For more information about how to obtain a region ID, see Query a region ID .
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console.
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console.
x-acs-instanceid	The ID of the instance on which you want to perform operations.
x-acs-caller-sdk-source	The custom name of the requester or the system. This parameter is used to identify the source of the API request.
x-acs-roleid	The ID of the role.
x-acs-userid	The ID of the user.

3.1.3.6. HTTP requests

3.1.3.6.1. Overview

This topic describes how to make API requests and is intended for users who use API URLs to initiate HTTP GET requests.

If you are using an SDK to initiate requests, you can skip this topic.

The request URL consists of different parameters and has a fixed syntax. The URL contains common request parameters, your signature, and operation-specific parameters. Sample request URLs are provided for each API operation. These URLs are not encoded to make them easy to read. Before you make a request, you must encode the request URL. After the authentication process is complete based on your signature, the results are returned to you. Response parameters are returned if the call is successful, whereas an error message is returned if the call fails. You can troubleshoot issues based on the error message.

3.1.3.6.2. Request structure

You can use API URLs to initiate HTTP GET requests. Each URL must include request parameters. This topic describes the syntax of HTTP GET requests and provides the endpoint of ASAPI.

Sample request

GetOrganizationList is used in the following unencoded URL request:

```
https://public.asapi.aliyuns.com/asapi/v3?Action=GetOrganizationList&Product=ascm&Version=2019-05-10&Id=<Organization ID>&<Common parameters>
```

- `https` : the protocol used to transmit the request.
- `public.asapi.aliyuns.com/asapi/v3` : the endpoint of ASAPI.
- `Action=GetOrganizationList Product=ascm Version=2019-05-10` : the API operation.
- `Id` : the ID of the organization.
- `<Common request parameters>` : common parameters specified in the system.

Protocols

ASAPI supports only the HTTP request protocol.

Endpoints

For more information about how to obtain the endpoint of ASAPI, see [Generate an endpoint for ASAPI](#).

You must use the `Action`, `Product`, and `Version` parameters to specify an API operation. For example, you can set `Action` to `GetOrganizationList`, `Product` to `ascm`, and `Version` to `2019-05-10`. You must also specify other operation-specific request parameters and common request parameters. For more information, see [Common request parameters](#).

3.1.3.6.3. Request signatures

This topic describes how to sign HTTP API requests. ASAPI uses the request signature to verify the identity of the request sender. ASAPI implements symmetric encryption by using an AccessKey pair to verify the identity of the request sender.

Note

- An AccessKey pair consists of an AccessKey ID and an AccessKey secret, which are issued to each user or organization in the Apsara Uni-manager Management Console.
- The AccessKey ID is used to verify identities of users, and the AccessKey secret is used to encrypt and verify signature strings. You must keep your AccessKey secret confidential.
- ASAPI provides SDKs for Node.js, Python, and Java to facilitate complex signature calculation. For more information, see [Obtain the core package for ASAPI](#).

Step 1: Create a canonicalized query string

1. Arrange the request parameters in alphabetical order, including all common and operation-specific parameters except the **Signature** parameter.

 Note

If you use the GET method to send a request, these parameters are connected by ampersands (&) and included in the request URL. The first request parameter follows a question mark (?).

2. Encode the request parameters and their values in UTF-8 based on RFC 3986. The following rules apply in the encoding process:
 - Letters, digits, and special characters such as hyphens (-), underscores (_), and periods (.) do not need to be encoded.
 - Other characters must be percent encoded in the %XY format. XY represents the ASCII code of the characters in hexadecimal notation. For example, double quotation marks (") are encoded as %22 .
 - Extended UTF-8 characters are encoded in the %XY%ZA... format.
 - Spaces must be encoded as %20 . Do not encode spaces as plus signs (+).

This encoding method is similar to but different from the **application/x-www-form-urlencoded** MIME encoding algorithm.

If you use **java.net.URLEncoder** from the Java standard library, use **percentEncode** to encode request parameters and their values. In the encoded query string, replace the plus signs (+) with %20 , asterisks (*) with %2A , and %7E with tildes (~). This way, you can obtain an encoded string that matches the preceding encoding rules.

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedOperationException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A")
        .replace("%7E", "~") : null;
}
```

3. Use equal signs (=) to connect the encoded request parameters and their values.
4. Use ampersands (&) to connect the encoded request parameters. These parameters must be arranged in the same order as those in Step 1.

Then, a canonicalized query string that follows the request structure is obtained. For more information, see [Request structure](#).

Step 2: Create a string-to-sign

1. Create a `string-to-sign` . You can also use `percentEncode` to encode the canonicalized query string created in Step 1 based on the following rules:

```
StringToSign=
HTTPMethod + "&" + //HTTPMethod: the HTTP method that is used to send a request, such as GET.
percentEncode("/") + "&" + //percentEncode("/"): Encode the forward slash (/) in UTF-8 as %2F.
percentEncode(CanonicalizedQueryString) // Encode the canonicalized query string created in Step 1.
```

2. Calculate the [RFC 2104](#)-compliant HMAC-SHA1 or HMAC-256 value of `StringToSign` . In the following example, Java Base64 is used:

```
String signature = Base64.getEncoder().encodeToString(signData);
```

 Note

When you calculate the signature, the key value specified by RFC 2104 is your `AccessKey secret` plus an ampersand (&), which has an ASCII value of 38. For more information, see [Obtain an AccessKey pair](#).

3. Encode the `Signature` parameter based on [RFC 3986](#) and add it to the canonicalized query string.

Example 1: Concatenate parameters

`GetOrganizationList` is used in this example to query regions. For example, `AccessKeyId` is set to `testid` and `AccessKeySecret` is set to `testsecret`. Perform the following operations:

1. Construct a canonicalized query string.

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

2. Create a `string-to-sign` .

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DGetOrganizationList%26Format%3DJSON%26Id%3D1%26Product%3Dascm%26RegionId%3Dcn-region-test%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D43967c6e-e3ad-44fa-b06b-fcf3db5f39d2%26SignatureVersion%3D1.0%26Timestamp%3D2020-09-15T08%253A50%253A40Z%26Version%3D2019-05-10
```

3. Calculate the signature.

Calculate the signature. The key value used for calculation is `testsecret&` because

`AccessKeySecret` is set to `testsecret`. The calculated signature is

`OLeaidS1JvxuMvnyH0wuJ+uX5qY=` . Java Base64 is used in the following sample code:

```
String signature = Base64.getEncoder().encodeToString(signData);
```

 Encode

`Signature=hvqv9LI5RE6kQ4YsQXFAf9VuRIY=` based on RFC 3986 and add it to the canonicalized query string.

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

You can use browsers or tools such as cURL and Wget to initiate an HTTP request for calling the

`DescribeRegions` operation to query the Alibaba Cloud region list.

Example 2: Use the standard library of the programming language

`GetOrganizationList` is used in this example to query regions. For example, `AccessKeyId` is set to `testid`, `AccessKeySecret` is set to `testsecret`, and all request parameters are stored in a Java `Map<String, String>` object.

1. Predefine an encoding method.

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedOperationException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("'", "%2A")
        .replace("%7E", "~") : null;
}
```

2. Predefine the time format for the `Timestamp` parameter. Specify the `Timestamp` parameter in the ISO 8601 format. The time must be in UTC+0.

```
private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static String formatIso8601Date(Date date) {
    SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
    df.setTimeZone(new SimpleTimeZone(0, "GMT"));
    return df.format(date);
}
```

3. Create a request string.

```
final String HTTP_METHOD = "GET";
Map parameters = new HashMap();
// Specify request parameters.
parameters.put("Product", "ascm");
parameters.put("Action", "GetOrganizationList");
parameters.put("Version", "2019-05-10");
parameters.put("AccessKeyId", "testid");
parameters.put("RegionId", "cn-region-test");
parameters.put("Timestamp", formatIso8601Date(new Date()));
parameters.put("SignatureMethod", "HMAC-SHA1");
parameters.put("SignatureVersion", "1.0");
parameters.put("SignatureNonce", UUID.randomUUID().toString());
parameters.put("Format", "JSON");
parameters.put("Id", "1");
// Arrange request parameters.
String[] sortedKeys = parameters.keySet().toArray(new String[{}]);
Arrays.sort(sortedKeys);
final String SEPARATOR = "&";
// Create a string-to-sign.
StringBuilder stringToSign = new StringBuilder();
stringToSign.append(HTTP_METHOD).append(SEPARATOR);
stringToSign.append(percentEncode("/")).append(SEPARATOR);
StringBuilder canonicalizedQueryString = new StringBuilder();
for(String key : sortedKeys) {
    // Encode the key and the value.
    canonicalizedQueryString.append("&")
        .append(percentEncode(key)).append("=")
        .append(percentEncode(parameters.get(key)));
}
//Encode the canonicalized query string.
stringToSign.append(percentEncode(
    canonicalizedQueryString.toString().substring(1)));
```

4. The signature. The key value used for calculation is `testsecret&` because `AccessKeySecret` is set to `testsecret`. The calculated signature is `OLeaidS1JvxuMvnyHOWuJ%2BuX5qY%3D`.

```
// The following sample code shows how to calculate the signature:
final String ALGORITHM = "HmacSHA1";
final String ENCODING = "UTF-8";
key = "testsecret&";
Mac mac = Mac.getInstance(ALGORITHM);
mac.init(new SecretKeySpec(key.getBytes(ENCODING), ALGORITHM));
byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));
String signature = Base64.getEncoder().encodeToString(signData);
```

Encode the Signature parameter based on [RFC 3986](#) and add it to the canonicalized query string. The following URL is obtained:

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

5. Use browsers or tools such as cURL and Wget to initiate an HTTP request.

3.1.3.6.4. Responses

Responses are returned only in the JSON format. If a return code is greater than or equal to 200 and less than 300, a success response is returned. Otherwise, an error response is returned.

Sample success responses

If a request is successful, the request ID and operation-specific parameter values are returned. For more information, see the developer guide of the corresponding Apsara Stack service.

```
"successResponse":true,
"eagleEyeTraceId":"xxxxxxxxxxxxxx",
"asapiSuccess":true,
"code":"200",
"cost":12,
"asapiRequestId":"xxxxxxxxxxxxxx",
"data":[
{
  "muserId":"aliyuntest",
  "internal":false,
  "level":"0",
  "supportRegions":"cn-*",
  "personNum":100,
  "mtime":1553247962000,
  "uuid":"1",
  "parentId":0,
  "supportRegionList":[
    "cn-*"
  ],
  "name":"root",
  "alias":"root",
  "ctime":1553247962000,
  "id":1,
  "resourceGroupNum":100,
  "cuserId":"aliyuntest"
}
],
"requestId":"xxxxxxxxxxxxxx",
"success":true,
"message":"success"
}
```

Sample error responses

If a request fails, the following information is returned. For more information, see the developer guide of the corresponding Apsara Stack service.


```
{
  "code": "asapi.server.api.notfound", // The error code.
  "eagleEyeTraceId": "xxx-xxx-xxx-xxx", // The trace ID provided by EagleEye.
  "asapiSuccess": false, // Indicates whether the request is successful.
  "hostId": "xxxx", // The ID of the host.
  "message": "Failed to ....", // The message returned.
  "asapiErrorMessage": "Failed to ....", // The error message provided by ASAPI.
  "asapiErrorCode": "asapi.server.api.notfound" // The error code provided by ASAPI.
}
```

3.1.4. Flink SQL

3.1.4.1. Overview

As a programming language that conforms to standard SQL semantics, Flink SQL is designed to simplify the computational model, making it easy for users to use Realtime Compute. Flink SQL has the following advantages:

- Low development cost. Flink SQL enables you to implement complex and varied business logic by using simple SQL statements instead of thousands of lines of Apache Storm code.
- Low maintenance cost. Flink SQL eliminates the need for you to maintain thousands of lines of Storm code.

When you use Apache Storm code, system exceptions, errors, or even breakdown may occur and cause data inaccuracies.

Flink SQL features specified semantics to resolve the preceding issues. Realtime Compute has the following advantages:

- Data accuracy. Realtime Compute provides unique exactly-once semantics to ensure data accuracy in various failover scenarios.
- Performance optimization. Realtime Compute integrates SQL optimization logic to translate SQL code into underlying Apache Storm code that can be used by universal developers. This helps to improve SQL code compilation efficiency.
- Programming extensibility. Realtime Compute supports user-defined functions (UDFs). You can customize functions to implement a custom business logic or a logic that cannot be achieved by SQL statements.
- System scalability. Realtime Compute supports auto scaling upon data surges (for example, during the Double 11) while ensuring job and system stability.

3.1.4.2. Keywords

This topic describes the keywords in Realtime Compute.

Realtime Compute keywords

Common keywords

Keyword type	Keywords
Data type	VARCHAR, INT, BIGINT, DOUBLE, DATE, BOOLEAN, TINYINT, SMALLINT, FLOAT, DECIMAL, and VARBINARY

Keyword type	Keywords
DDL statement	CREATE TABLE, CREATE FUNCTION, and CREATE VIEW
DML statement	INSERT INTO
SELECT clause	SELECT FROM, WHERE, GROUP BY, and JOIN

Naming conventions

Names of source tables, result tables, views, and aliases must follow the standard database naming conventions. The names must start with a letter, and can only contain letters, numbers, and underscores (_).

Reserved keywords

The following strings are reserved as keywords for later use. If you want to use any of the following keywords as a field name, enclose the keyword in backticks (` `), for example, `value` .

A. ABS. ABSOLUTE. ACTION. ADA. ADD. ADMIN. AFTER. ALL. ALLOCATE. ALLOW. ALTER. ALWAYS. AND. ANY. ARE. ARRAY. AS. ASC. ASENSITIVE. ASSERTION. ASSIGNMENT. ASYMMETRIC. AT. ATOMIC. ATTRIBUTE. ATTRIBUTES. AUTHORIZATION. AVG. BEFORE. BEGIN. BERNULLI. BETWEEN. BIGINT. BINARY. BIT. BLOB. BOOLEAN. BOTH. BREADTH. BY. C. CALL. CALLED. CARDINALITY. CASCADE. CASCADED. CASE. CAST. CATALOG. CATALOG NAME. CFIL. CEILING. CENTURY. CHAIN. CHAR. CHARACTER. CHARACTERISTICS. CHARACTERS. CHARACTER LENGTH. CHARACTER SET CATALOG. CHARACTER SET NAME. CHARACTER SET SCHEMA. CHAR LENGTH. CHECK. CLASS. ORIGIN. CLOB. CLOSE. COALESCE. COBOL. COLLATE. COLLATION. COLLATION CATALOG. COLLATION NAME. COLLATION SCHEMA. COLLECT. COLUMN. COLUMN NAME. COMMAND FUNCTION. COMMAND FUNCTION CODE. COMMIT. COMMITTED. CONDITION. CONDITION NUMBER. CONNECT. CONNECTION. CONNECTION NAME. CONSTRAINT. CONSTRAINTS. CONSTRAINT CATALOG. CONSTRAINT NAME. CONSTRAINT SCHEMA. CONSTRUCTOR. CONTAINS. CONTINUE. CONVERT. CORR. CORRESPONDING. COUNT. COVAR POP. COVAR SAMP. CREATE. CROSS. CUBE. CUME DIST. CURRENT. CURRENT CATALOG. CURRENT DATE. CURRENT DEFAULT TRANSFORM GROUP. CURRENT PATH. CURRENT ROLE. CURRENT SCHEMA. CURRENT TIME. CURRENT TIMESTAMP. CURRENT TRANSFORM GROUP FOR TYPE. CURRENT USER. CURSOR. CURSOR NAME. CYCLE. DATA. DATABASE. DATE. DATETIME. INTERVAL. CODE. DATETIME. INTERVAL. PRECISION. DAY. DEALLOCATE. DEC. DECADE. DECIMAL. DECLARE. DEFAULT. DEFAULTS. DEFERRABLE. DEFERRED. DEFINED. DEFINER. DEGREE. DELETE. DENSE RANK. DEPTH. DEREF. DERIVED. DESC. DESCRIBE. DESCRIPTION. DESCRIPTOR. DETERMINISTIC. DIAGNOSTICS. DISALLOW. DISCONNECT. DISPATCH. DISTINCT. DOMAIN. DOUBLE. DOW. DOY. DROP. DYNAMIC. DYNAMIC FUNCTION. DYNAMIC FUNCTION CODE. EACH. ELEMENT. ELSE. END. END-EXEC. EPOCH. EQUALS. ESCAPE. EVERY. EXCEPT. EXCEPTION. EXCLUDE. EXCLUDING. EXEC. EXECUTE. EXISTS. EXP. EXPLAIN. EXTEND. EXTERNAL. EXTRACT. FALSE. FETCH. FILTER. FINAL. FIRST. FIRST VALUE. FLOAT. FLOOR. FOLLOWING. FOR. FOREIGN. FORTRAN. FOUND. FRAC. SECOND. FREE. FROM. FULL. FUNCTION. FUSION. G. GENERAL. GENERATED. GET. GLOBAL. GO. GOTO. GRANT. GRANTED. GROUP. GROUPING. HAVING. HIERARCHY. HOLD. HOUR. IDENTITY. IMMEDIATE. IMPLEMENTATION. IMPORT. IN. INCLUDING. INCREMENT. INDICATOR. INITIALLY. INNER. INOUT. INPUT. INSENSITIVE. INSERT. INSTANCE. INSTANTIABLE. INT. INTEGER. INTERSECT. INTERSECTION. INTERVAL. INTO. INVOKER. IS. ISOLATION. JAVA. JOIN. K. KEY. KEY MEMBER. KEY TYPE. LABEL. LANGUAGE. LARGE. LAST. LAST VALUE. LATERAL. LEADING. LEFT. LENGTH. LEVEL. LIBRARY. LIKE. LIMIT. IN. LOCAL. LOCAL TIME. LOCALTIMESTAMP. LOCATOR. LOWER. M. MAP. MATCH. MATCHED. MAX. MAXVALUE. MEMBER. MERGE. MESSAGE. LENGTH. MESSAGE. OCTET. LENGTH. MESSAGE. TEXT. METHOD. MICROSECOND. MILLENNIUM. MIN. MINUTE. MINVALUE. MOD. MODIFIES. MODULE. MONTH. MORE. MULTiset. MUMPS. NAME. NAMES. NATIONAL. NATURAL. NCHAR. NCI. OR. NESTING. NEW. NEXT. NO. NONE. NORMALIZE. NORMALIZED. NOT. NULL. NULLABLE. NULLIF. NULLS. NUMBER. NUMERIC. OBJECT. OCTETS. OCTET. LENGTH. OF. OFFSET. OLD. ON. ONLY. OPEN. OPTION. OPTIONS. OR. ORDER. ORDERING. ORDINALITY. OTHERS. OUT. OUTER. OUTPUT. OVER. OVERLAPS. OVERLAY. OVERRIDING. PAD. PARAMETER. PARAMETER_MODE. PARAMETER_NAME. PARAMETER_ORDINAL_POSITION. PARAMETER_SPECIFIC_CATALOG.

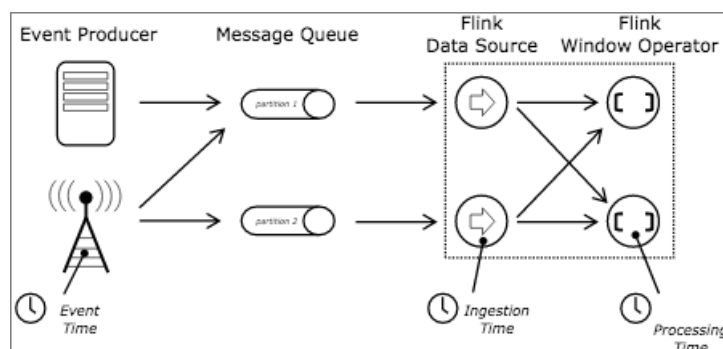
PARAMETER SPECIFIC NAME, PARAMETER SPECIFIC SCHEMA, PARTIAL, PARTITION, PASCAL, PASSTHROUGH, PATH, PERCENTILE CONT, PERCENTILE DISC, PERCENT RANK, PLACING, PLAN, PLI, POSITION, POWER, PRECEDING, PRECISION, PREPARE, PRESERVE, PRIMARY, PRIOR, PRIVILEGES, PROCEDURE, PUBLIC, QUARTER, RANGE, RANK, READ, READS, REAL, RECURSIVE, REF, REFERENCES, REFERENCING, REGR_AVGX, REGR_AVGY, REGR_COUNT, REGR_INTERCEPT, REGR_R2, REGR_SLOPE, REGR_SXX, REGR_SXY, REGR_SYY, RELATIVE, RELEASE, REPEATABLE, RESET, RESTART, RESTRICT, RESULT, RETURN, RETURNED_CARDINALITY, RETURNED_LENGTH, RETURNED_OCTET_LENGTH, RETURNED_SQLSTATE, RETURNS, REVOKE, RIGHT, ROLE, ROLLBACK, ROLLUP, ROUTINE, ROUTINE CATALOG, ROUTINE NAME, ROUTINE SCHEMA, ROW, ROWS, ROW_COUNT, ROW_NUMBER, SAVEPOINT, SCALE, SCHEMA, SCHEMA NAME, SCOPE, SCOPE CATALOGS, SCOPE NAME, SCOPE SCHEMA, SCROLL, SEARCH, SECOND, SECTION, SECURITY, SELECT, SELF, SENSITIVE, SEQUENCE, SERIALIZABLE, SERVER, SERVER NAME, SESSION, SESSION USER, SET, SETS, SIMILAR, SIMPLE, SIZE, SMALLINT, SOME, SOURCE, SPACE, SPECIFIC, SPECIFICTYPE, SPECIFIC NAME, SOL, SOL_EXCEPTION, SOLSTATE, SOLWARNING, SOL_TSI_DAY, SOL_TSI_FRAC_SECOND, SOL_TSI_HOUR, SOL_TSI_MICROSECOND, SOL_TSI_MINUTE, SOL_TSI_MONTH, SOL_TSI_QUARTER, SOL_TSI_SECOND, SOL_TSI_WEEK, SOL_TSI_YEAR, SORT, START, STATE, STATEMENT, STATIC, STDDEV_POP, STDDEV_SAMP, STREAM, STRUCTURE, STYLE, SUBCLASS_ORIGIN, SUBMULTISET, SUBSTITUTE, SUBSTRING, SUM, SYMMETRIC, SYSTEM, SYSTEM USER, TABLE, TABLESAMPLE, TABLE_NAME, TEMPORARY, THEN, TIES, TIME, TIMESTAMP, TIMESTAMPADD, TIMESTAMPDIFF, TIMEZONE_HOUR, TIMEZONE_MINUTE, TINYINT, TO, TOP_LEVEL_COUNT, TRAILING, TRANSACTION, TRANSACTIONS_ACTIVE, TRANSACTIONS_COMMITTED, TRANSACTIONS_ROLLED_BACK, TRANSFORM, TRANSFORMS, TRANSLATE, TRANSLATION, TREAT, TRIGGER, TRIGGER CATALOG, TRIGGER NAME, TRIGGER SCHEMA, TRIM, TRUE, TYPE, UNESCAPE, UNBOUNDED, UNCOMMITTED, UNDER, UNION, UNIQUE, UNKNOWN, UNNAMED, UNNEST, UPDATE, UPPER, UPSERT, USAGE, USER, USER_DEFINED_TYPE_CATALOG, USER_DEFINED_TYPE_CODE, USER_DEFINED_TYPE_NAME, USER_DEFINED_TYPE_SCHEMA, USING, VALUE, VALUES, VARBINARY, VARCHAR, VARYING, VAR_POP, VAR_SAMP, VERSION, VIEW, WEEK, WHEN, WHENEVER, WHERE, WIDTH_BUCKET, WINDOW, WITH, WITHIN, WITHOUT, WORK, WRAPPER, WRITE, XML, YEAR, ZONE

3.1.4.3. Basic concepts

3.1.4.3.1. Time attributes

This topic describes the following time attributes that are supported by Flink SQL: event time and processing time.

Flink supports three time attributes for the processing of streaming data: processing time, event time, and ingestion time.



Flink SQL supports only two of the three time attributes:

- Event time: the event time that you provide in the data store. In most cases, the event time is the original time when the data is created.
- Processing time: the local system time when the system processes an event. The unit is milliseconds.

Event Time

The event time is also known as rowtime. The event time attribute must be declared in the data definition language (DDL) statement that you execute to create a source table. You can declare a field in the source table as the rowtime field. Note that you can declare a field of only the `TIMESTAMP` type as the rowtime field. In the future, you can declare a field of the `LONG` type as the rowtime field. If the source table does not contain a `TIMESTAMP` column, you can use a computed column to create a `TIMESTAMP` column based on an existing column. For more information, see [Computed column](#).

In some scenarios, the order in which data records are received may be different from the order in which they are processed. The possible causes include out-of-order input data and network jitters. The network jitters may be caused by network congestions and transmission latencies. Before you define a rowtime field, define a computing method for watermarks in an explicit way. For more information, see [Watermark](#).

In the following example, data is aggregated by using event time-based window functions:

```
CREATE TABLE tt_stream (
  a VARCHAR,
  b VARCHAR,
  ts TIMESTAMP,
  WATERMARK wk1 FOR ts as withOffset (ts, 1000) --Define a computing method for watermarks.
) WITH (
  type = 'sls',
  topic = '<yourTopicName>',
  accessId = '<yourAccessId>',
  accessKey = '<yourAccessSecret>'
);
CREATE TABLE rds_output (
  id VARCHAR,
  win_start TIMESTAMP,
  win_end TIMESTAMP,
  cnt BIGINT
) WITH (
  type = 'rds',
  url = 'jdbc:mysql://****3306/test',
  tableName = '<yourTableName>',
  userName = '<yourUserName>',
  password = '<yourPassword>'
);
INSERT
  INTO rds_output
SELECT
  a AS id,
  SESSION_START (ts, INTERVAL '1' SECOND) AS win_start,
  SESSION_END (ts, INTERVAL '1' SECOND) AS win_end,
  COUNT (a) AS cnt
FROM
  tt_stream
GROUP
  BY SESSION (ts, INTERVAL '1' SECOND),
  a
```

Processing Time

The processing time is generated by the system and is not included in the raw data. Therefore, you must explicitly define a processing time column when you declare the source table.

```
fileName as PROCTIME()
```

In the following example, processing time-based window functions are used to aggregate data:

```
CREATE TABLE mq_stream (  
  a VARCHAR,  
  b VARCHAR,  
  c BIGINT,  
  ts AS PROCTIME () --Explicitly define a processing time column when you declare the source table.  
) WITH (  
  type = 'mq',  
  topic = '<yourTopic>',  
  accessId = '<yourAccessId>',  
  accessKey = '<yourAccessSecret>'  
)  
;  
CREATE TABLE rds_output (  
  id VARCHAR,  
  win_start TIMESTAMP,  
  win_end TIMESTAMP,  
  cnt BIGINT  
) with (  
  type = 'rds',  
  url = '<yourDatebaseURL>',  
  tableName = '<yourDatabasTableName>',  
  userName = '<yourUserName>',  
  password = '<yourPassword>'  
)  
;  
INSERT  
  INTO rds_output  
SELECT  
  a AS id,  
  SESSION_START (ts, INTERVAL '1' SECOND) AS win_start,  
  SESSION_END (ts, INTERVAL '1' SECOND) AS win_end,  
  COUNT (a) AS cnt  
FROM  
  mq_stream  
GROUP  
  BY SESSION (ts, INTERVAL '1' SECOND),  
  a
```

Expiration of time attributes

The time attribute of fields no longer takes effect after one of the following operations is completed:

- GROUP BY operations on the fields that are not defined as time attribute fields, except the GROUP BY operations in tumbling, sliding, and session windows. For more information, see [Tumbling windows](#), [Sliding windows](#), and [Session window](#).
- JOIN operations on two data streams.

 **Note** For more information, see [JOIN statements](#).

- MATCH_RECOGNIZE operations in complex event processing (CEP) statements. For more information, see [CEP statements](#).
- PARTITION BY operations in OVER windows. For more information, see [OVER windows](#).
- UNION operations. UNION is equivalent to the combination of RETRACT and UNION ALL.

If you use time-based window functions for computing after the preceding operations, errors are returned, such as `org.apache.flink.table.api.ValidationException: Window can only be defined over a time attribute column.`

3.1.4.3.2. Watermark

Realtime Compute for Apache Flink aggregates data by using window functions based on the time attribute. To use window functions based on the event time of a job, you must define a watermark when you declare a source table.

A watermark is used to measure the progress of [Event Time](#). It is a hidden data attribute. A watermark is defined in the DDL statement of the source table. Flink provides the following statement to define a watermark:

```
WATERMARK [watermarkName] FOR <rowtime_field> AS withOffset(<rowtime_field>, offset)
```

 **Note** For more information about the time attributes of Realtime Compute for Apache Flink, see [Time attributes](#).

Parameter	Required	Description
watermarkName	No	The name of the watermark.
<rowtime_field>	Yes	The column used to generate the watermark. <rowtime_field> is identified as the event time column and must be a column of the TIMESTAMP type defined in the table. You can use <rowtime_field> to define a window in the job code.
withOffset	Yes	The policy to generate a watermark. In this example, the watermark value is generated by using the following formula: <code><rowtime_field> - offset</code> . The first parameter in <code>withOffset</code> must be <rowtime_field>.
offset	Yes	The offset between the watermark value and event time, in milliseconds.

A specific field in a data record indicates the time when the record was generated. For example, a table contains a `rowtime` field whose data type is `TIMESTAMP`, and one of its values is `1501750584000 (2017-08-03 08:56:24.000)`. If you want to define a watermark based on the `rowtime` field and configure a 4-second offset, add the following definition:

WATERMARK FOR rowtime AS withOffset(rowtime, 4000)

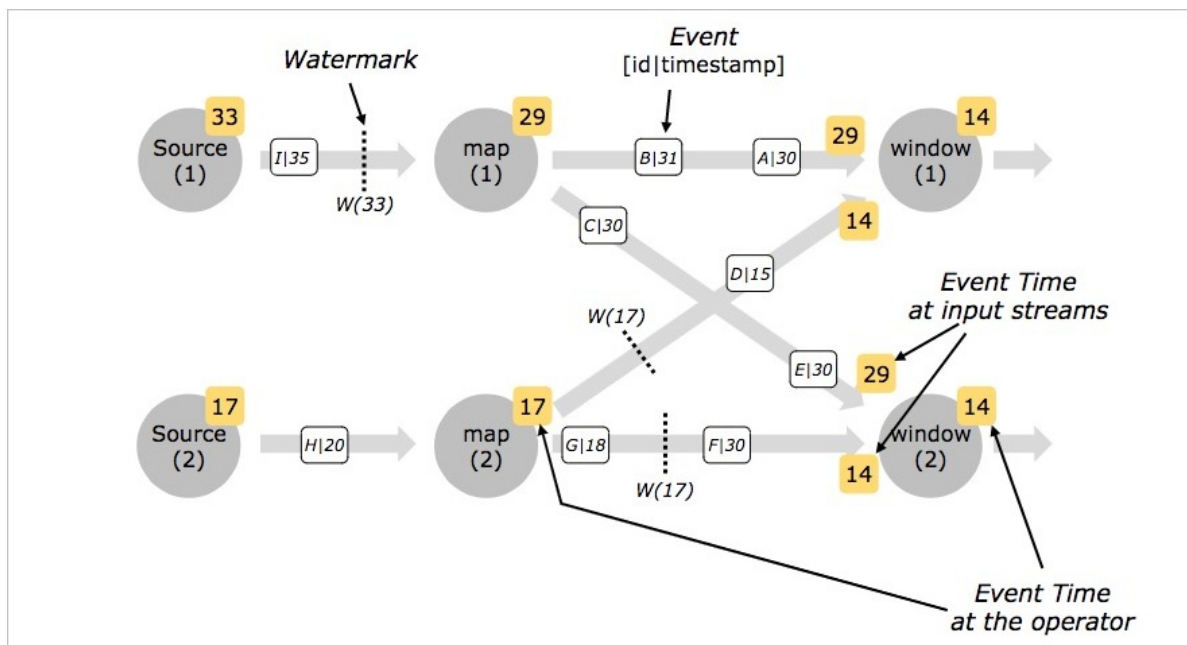
In this example, the watermark time of the data record is $1501750584000 - 4000 = 1501750580000$ (2017-08-03 08:56:20.000). This means that all data whose timestamp is earlier than 1501750580000 (2017-08-03 08:56:20.000) has arrived.

? Note

- If you use an event time-based watermark, the `rowtime` field must be of the `TIMESTAMP` type. Realtime Compute for Apache Flink supports 13-digit UNIX timestamps in milliseconds. If the rowtime field is of another type or the UNIX timestamp is not 13 digits in length, we recommend that you use a computed column to convert the time. For more information, see [Computed column](#).
- The event time and processing time can only be declared in the source table. For more information, see [Event Time](#) and [Processing Time](#).

Summary

- A watermark indicates that all the events whose timestamp t' is earlier than the watermark time t ($t' < t$) have occurred. After the watermark time t takes effect, all subsequently received data records whose event time is earlier than t are discarded. Realtime Compute for Apache Flink will allow you to change the configuration and update the subsequent data.
- Watermarks are important for data streams that arrive out of order because the watermarks help ensure that the computing in a window is correct even if some events arrive late.
- If an operator has multiple input data streams for parallel processing, the event time of the data stream with the shortest time is used as the event time of the operator.



3.1.4.3.3. Computed column

You can calculate a value for a computed column by using data from other columns. If your source table does not have a column of the `TIMESTAMP` type, you can use a computed column to convert a field of another type to the `TIMESTAMP` type.

Concept

A computed column is a virtual column that is not stored in a physical table. You can create computed columns by using expressions, built-in functions, or user-defined extensions (UDXs). In Flink SQL, a computed column can be used the same as columns that are stored in a physical table.

Usage

Currently, the event time (also known as rowtime) column in a **Watermark** must be of the `TIMESTAMP` type. The `LONG` data type will be supported in the future. You can only define a watermark in the DDL statement of a source table. If a source table does not have a column of the `TIMESTAMP` type, you can use a computed column to convert a field of another type to the `TIMESTAMP` type.

Syntax

```
column_name AS computed_column_expression
```

Example

The rowtime column in a watermark must be of the `TIMESTAMP` type. Currently, Realtime Compute only supports 13-bit UNIX timestamps measured in milliseconds. If the `TIME` column in a DataHub source table is defined as a 16-bit UNIX timestamp measured in microseconds, you can use a computed column to convert the 16-bit UNIX timestamp to a 13-bit UNIX timestamp. The sample code is as follows:

```
CREATE TABLE test_stream(  
  a INT,  
  b BIGINT,  
  `TIME` BIGINT,  
  ts AS TO_TIMESTAMP(`TIME`/1000), -- Use a computed column to convert 16-bit timestamps to 13-bit time  
  stamps.  
  WATERMARK FOR ts AS WITHOFFSET(ts, 1000)  
) WITH (  
  type = 'datahub',  
  ...  
);
```

The ``TIME`` field in the source table contains the date and time information. The value of this field is of the `BIGINT` type. A computed column is created to convert the `TIME` column of the `BIGINT` type to the `ts` column of the `TIMESTAMP` type. The `ts` column is used as the rowtime of a watermark.

3.1.4.4. Data types

3.1.4.4.1. Data types

This topic describes data types supported by Realtime Compute.

Supported data types

Data type	Description	Value range
VARCHAR	Character string with a changeable length	Maximum storage capacity: 4 MB
BOOLEAN	Logical value	Valid values: TRUE, FALSE, and UNKNOWN
TINYINT	Tiny integer (one byte)	-128 to 127
SMALLINT	Small integer (two bytes)	-32768 to 32767
INT	Integer (four bytes)	-2147483648 to 2147483647
BIGINT	Big integer (eight bytes)	-9223372036854775808 to 9223372036854775807
FLOAT	Four-byte floating point number	Six digits of precision
DECIMAL	Decimal	An example value for <code>DECIMAL(5,2)</code> is <code>123.45</code> .
DOUBLE	Eight-byte floating point number	15 decimal digits of precision
DATE	Date	Example: <code>DATE'1969-07-20'</code>
TIME	Time	Example: <code>TIME '20:17:40'</code>
TIMESTAMP	Timestamp containing both date and time parts	Example: <code>TIMESTAMP '1969-07-20 20:17:40'</code>
VARBINARY	Binary data	Byte array

Type conversion

Convertible	Data Type										
	VARCHAR	BOOLEAN	TINYINT	SMALLINT	INT	BIGINT	FLOAT	DOUBLE	DATE	TIMESTAMP	TIME
VARCHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
BOOLEAN	Y	Y	N	N	N	N	N	N	N	N	N
TINYINT	Y	N	Y	Y	Y	Y	Y	Y	N	N	N
SMALLINT	Y	N	Y	Y	Y	Y	Y	Y	N	N	N
INT	Y	N	Y	Y	Y	Y	Y	Y	N	N	N
BIGINT	Y	N	Y	Y	Y	Y	Y	Y	N	N	N
FLOAT	Y	N	Y	Y	Y	Y	Y	Y	N	N	N
DOUBLE	Y	N	Y	Y	Y	Y	Y	Y	N	N	N
DATE	Y	N	N	N	N	N	N	N	Y	Y	N
TIMESTAMP	Y	N	N	N	N	N	N	N	Y	Y	Y
TIME	Y	N	N	N	N	N	N	N	N	Y	Y

Type conversion example

Input example

var1 (VARCHAR)	big1 (BIGINT)
1000	323

Sample code

```
CAST(var1 AS BIGINT) AS AA;
CAST(big1 AS VARCHAR) AS BB;
```

Output example

AA (BIGINT)	BB (VARCHAR)
1000	323

3.1.4.4.2. Data types and operators

This topic describes data types and operators in Realtime Compute.

You can perform mathematical and logical operations for different data types. Examples are provided in the following table.

Operator	Description	Data types supported by numeric1 and numeric2	Example
numeric1 + numeric2	Addition	INT, DOUBLE, DECIMAL, BIGINT	2 + 4.2
numeric1 - numeric2	Subtraction		3 - 5.3
numeric1 * numeric2	Multiplication		2 * 4
numeric1 / numeric2	Division		2.4 / 5
numeric1 > numeric2	>		2.4 > 5
numeric1 < numeric2	<		2.4 < 5
numeric1 >= numeric2	>=		2.4 >= 5
numeric1 <= numeric2	<=		2.4 <= 5
numeric1 = numeric2	=	INT, DOUBLE, DECIMAL, BIGINT, VARCHAR	'inhone' = 5
numeric1 <> numeric2	<>		'inhone' <> 5

3.1.4.5. DDL statements

3.1.4.5.1. Overview

Syntax

```
CREATE TABLE tableName
  (columnName dataType [, columnName dataType]*)
  [ WITH (propertyName=propertyValue [, propertyName=propertyValue]*) ];
```

Description

Realtime Compute does not store data from source and result tables. You can only perform operations on source and result tables by using DDL statements to declare references to data stores. Example:

```
create table tt_stream(
  a varchar,
  b varchar,
  c varchar
) with (
  type='DataHub',
  topic='blink_tt_test',
  accessId='yourAccessId',
  accessKey='yourAccessSecret'
);
```

This Flink SQL statement is not used to create a DataHub topic, but to create a table named `tt_stream` in Realtime Compute and use the table to declare a reference to a DataHub topic. You can use `tt_stream` as an alias in subsequent DML statements to perform operations on the table.

- A declaration of a table (`tt_stream` in this example) is valid only for the jobs generated from the current SQL file. A job is generated after an SQL file is published. The table can also be declared for reference in other SQL files under the same project.
- Since Realtime Compute adopts standard SQL, keywords, table names, and field names in DDL statements are not case-sensitive.
- Table and field names must start with a letter or digit and can contain only letters, digits, and underscores (`_`).
- The method of mapping fields between a source table and an upstream data store depends on the feature of the data store. If the data store such as DataHub supports mapping based on key values, the field names in the source table must be the same as those in the data store. However, the number of fields can be different. If a data store does not support mapping based on key values, both the number and sequence of fields defined in the source table must be the same as those in the data store. We recommend that you use the same field names and the same number of fields in DDL statements as those in external data stores. This can prevent data errors caused by inaccurate definitions.

Mappings

Data stores are classified into two types based on whether the data store has a schema.

- Sequence mapping

Systems such as Message Queue (MQ) are non-structured storage systems that do not have a schema. You can customize field names in DDL statements. However, the data types and the number of fields must be the same as those in the data store.

The following example shows an MQ record:

```
asavfa,sddd32,sdfds
```

You must set field names of the `tt_stream` table based on naming conventions.

```
create table tt_stream(  
  a varchar,  
  b varchar,  
  c varchar  
) with (  
  type='mq',  
  topic='blink_mq_test',  
  accessId='yourAccessId',  
  accessKey='yourAccessSecret'  
);
```

- Name mapping

For structured storage systems such as DataHub, field names and data types are defined based on a table schema. We recommend that you define fields by using Flink SQL based on the table schema. The names, number, and sequence of fields must be the same as those in the data store.

 **Note** If field names in the data store (such as Tablestore) are case-sensitive, you must enclose field names in backticks (` `) in Flink SQL statements. DataHub is used in this example. In DDL statements, field names of the declared source table must apply the same casing that DataHub uses.

The following table describes a sample schema of DataHub.

Sample schema of DataHub

Field	Data type
name	varchar
age	bigint
value	varchar

We recommend that you declare references for all fields. Note that you can reference only the fields that are included in the data store. The following code shows a sample of the DDL statement:

```
create table stream_result (
  name varchar,
  age bigint,
  value varchar
) with (
  type='DataHub',
  endpoint='http://dh-cn-hangzhou.aliyuncs.com',
  accessId='yourAccessId',
  accessKey='yourAccessSecret',
  project='project',
  topic='topic'
);
```

Case insensitivity

Realtime Compute adopts standard SQL. Therefore, fields are not case-sensitive. For example, the following two statements produce the same effect. Statement 1:

```
create table stream_result (
  name    varchar,
  value   varchar
);
```

Statement 2:

```
create table STREAM_RESULT (
  NAME     varchar,
  VALUE    varchar
);
```

Realtime Compute references data from multiple data stores. Some data stores, such as Tablestore, are case-sensitive in field names. If a NAME parameter is defined in Tablestore, use the following DDL statement:

```
create table STREAM_RESULT (
  `NAME`    varchar,
  `VALUE`    varchar
);
```

In all subsequent DML statements, the parameter must be enclosed in backticks (``) if it is referenced. The following code shows an example:

```
INSERT INTO Table_A
SELECT
  `NAME`,
  `VALUE`
FROM
  Table_B;
```

3.1.4.5.2. Create a data source table

3.1.4.5.2.1. Overview

Source tables for Realtime Compute store streaming data. Streaming data triggers stream processing jobs in Realtime Compute. To process streaming data, you must create at least one source table that provides streaming data for Realtime Compute jobs.

Syntax

```
CREATE TABLE tableName
  (columnName dataType [, columnName dataType]*)
  [ WITH (propertyName=propertyValue [, propertyName=propertyValue]*) ];
```

Example

```
CREATE TABLE datahub_stream(
  name VARCHAR,
  age BIGINT,
  birthday BIGINT
) WITH (
  type='datahub',
  endPoint='<yourEndpoint>',
  project='<yourProjectName>',
  topic='<yourTopicName>',
  accessId='<yourAccessId>',
  accessKey='<yourAccessKey>',
  startTime='2017-07-21 00:00:00'
);
```

```
CREATE TABLE metaq_stream(
  x VARCHAR,
  y VARCHAR,
  z VARCHAR
) WITH (
  type='mq',
  topic='<yourTopicName>',
  endpoint='<yourEndpointName>',
  pullIntervalMs='1000',
  accessId='<yourAccessId>',
  accessKey='<yourAccessKey>',
  startMessageOffset='1000',
  consumerGroup='<yourConsumerGroup>',
  fieldDelimiter='|'
);
```

Retrieve attribute fields from a source table

- Syntax

Realtime Compute provides the HEADER keyword in the data definition language (DDL) statement of a source table for you to retrieve the attribute fields from the source table.

```
CREATE TABLE sourcetable
(
  `timestamp` VARCHAR HEADER,
  name VARCHAR,
  MsgID VARCHAR
)WITH(
  type='RDS'
);
```

In this example, the 'timestamp' field is defined as **HEADER**. Realtime Compute reads the values of attribute fields from the source table. Then, the 'timestamp' field is used as a common field.

 **Note** The default attribute fields vary based on the source table type, such as DataHub, Log Service, and Message Queue (MQ). You can customize attribute fields for certain types of source tables. For more information, see the topics about source tables of the related type.

- Example

A Log Service source table is used as an example to explain how to retrieve attribute fields. A Log Service source table has three attribute fields listed in the following table.

Field	Description
source	The message source.
topic	The message topic.
timestamp	The time when a data record enters Log Service.

 **Note** To obtain attribute fields of a source table, you must add the **HEADER** keyword to the end of a field declaration.

Example:

- Test data

```
__topic__: ens_altar_flow
result: {"MsgID":"ems0a","Version":"0.0.1"}
```

- Test statements

```
CREATE TABLE sls_log (  
  __topic__ VARCHAR HEADER,  
  result VARCHAR  
)WITH(  
  type='sls'  
);  
CREATE TABLE sls_out (  
  name varchar,  
  MsgID varchar,  
  Version varchar  
)WITH(  
  type='RDS'  
);  
INSERT INTO sls_out  
SELECT  
  __topic__,  
  JSON_VALUE(result,'$.MsgID'),  
  JSON_VALUE(result,'$.Version')  
FROM  
  sls_log
```

- Test results

name(VARCHAR)	MsgID(VARCHAR)	Version(VARCHAR)
ens_altar_flow	ems0a	0.0.1

Source tables for window function-based computing

Realtime Compute can aggregate data by using window functions based on the event time or processing time. If window functions are used for a stream processing job, you must define the watermark and computed column in the DDL statement of a source table. For more information about the watermark and computed column, see [Watermark](#) and [Computed column](#). For more information about data aggregation based on time attributes, see [Time attributes](#).

3.1.4.5.2.2. Create a DataHub source table

This topic describes how to create a DataHub source table in Realtime Compute for Apache Flink. It also describes the attribute fields, parameters in the WITH clause, and data type mapping involved when you create a DataHub source table.

 **Notice** This topic applies only to Blink 1.4.5 and later.

Introduction to DataHub

DataHub is a data distribution platform designed to process streaming data. You can publish and subscribe to applications for streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data. Realtime Compute for Apache Flink uses DataHub to store input and output data for the processing of streaming data.

Syntax

In Realtime Compute for Apache Flink, you can use DataHub to store input data. The following code shows an example:

```
CREATE TABLE datahub_stream(  
  name VARCHAR,  
  age BIGINT,  
  birthday BIGINT  
) WITH (  
  type='datahub',  
  endPoint='http://dh-cn-hangzhou.aliyun-inc.com',  
  project='<yourProjectName>',  
  topic='<yourTopic>',  
  accessId='<yourAccessID>',  
  accessKey='<yourAccessSecret>',  
  startTime='2017-07-21 00:00:00'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to <code>datahub</code> .
endPoint	The endpoint of DataHub.	Yes	Contact Alibaba Cloud O&M engineers.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
project	A project in DataHub.	Yes	None.
topic	A topic of the project.	Yes	None.  Notice Topics support only the data records of the tuple format.
startTime	The start time of DataHub logs.	No	This parameter is required for batch processing. The format of the value is <code>yyyy-MM-dd hh:mm:ss</code> .
endTime	The end time of DataHub logs.	No	This parameter is required for batch processing. The format of the value is <code>yyyy-MM-dd hh:mm:ss</code> .

Parameter	Description	Required	Remarks
maxRetryTimes	The maximum number of retries for reading data from DataHub.	No	The default value of this parameter varies based on Blink versions. - For versions earlier than Blink 2.2.7, the default value is 3. - For Blink 2.2.7 and later, the default value is 20.
retryIntervalMs	The interval of retries.	No	Unit: milliseconds. The default value of this parameter varies based on Blink versions. - For versions earlier than Blink 2.2.7, the default value is 1000. - For Blink 2.2.7 and later, the default value is 50.
batchReadSize	The number of data records that are read at a time.	No	Default value: 10. Maximum value: 1000.
lengthCheck	The policy for checking the number of fields parsed from a row of data.	No	- Default value: NONE - If the number of fields parsed from a row of data is greater than the number of defined fields, data is extracted from left to right based on the order of defined fields. - If the number of fields parsed from a row of data is less than the number of defined fields, this row of data is skipped. - SKIP: If the number of fields parsed from a row of data is different from the number of defined fields, this row of data is skipped. - EXCEPTION: If the number of fields parsed from a row of data does not match the number of defined fields, an exception is reported. - PAD: Data is padded from left to right based on the order of defined fields.  Note - If the number of fields parsed from a row of data is greater than the number of defined fields, data is padded from left to right based on the order of defined fields. - If the number of fields parsed from a row of data is less than the number of defined fields, the values of the missing fields are padded with null from left to right.

Parameter	Description	Required	Remarks
columnErrorDebug	Specifies whether debugging is enabled.	No	- false (default value): Debugging is disabled. - true: Debugging is enabled, and logs that contain parsing exceptions are printed.
isBlob	Specifies whether data of DataHub is of the BLOB type.	No	Default value: false.  Notice - Only Blink 3.4.X and later versions support this parameter. - When you use the BLOB type, you must declare fields as VARBINARY, which is similar to METAQ.

Field type mapping

The following table lists the mapping between DataHub data types and data types of Realtime Compute for Apache Flink. We recommend that you declare the field type mapping in a DDL statement.

DataHub data type	Data type of Realtime Compute for Apache Flink
BIGINT	BIGINT
TIMESTAMP	
STRING	VARCHAR
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DECIMAL	DECIMAL

Attribute field

You can use Flink SQL to retrieve the attribute field `TIMESTAMP` of a DataHub source table. You can obtain the system time at which each data record is written into DataHub after the attribute field is read.

Field	Description
TIMESTAMP	The system time at which each data record is written into DataHub.

Shard ID	System Time	name (STRING)	place (STRING)
0	2019-02-15 11:11:46	/abCQ7	beijing
0	2019-02-15 11:11:46	@*G%T{	guangzhou
0	2019-02-15 11:11:46	XvoedT	guangzhou
0	2019-02-15 11:11:46	%:IHj+O	guangzhou

Sample code

The following sample code demonstrates how to create a DataHub source table in a Realtime Compute for Apache Flink job.

```
create table datahub_input(  
  name VARCHAR  
) with (  
  type='datahub',  
  endPoint='http://dh-cn-hangzhou.aliyun-inc.com',  
  project='test1',  
  topic='topic1',  
  accessId='<yourAccessID>',  
  accessKey='<yourAccessSecret>',  
  startTime='2018-06-01 00:00:00'  
);  
create table test_out(  
  name VARCHAR  
) with (  
  type='print'  
);  
INSERT INTO test_out  
SELECT  
  LOWER(name)  
from datahub_input;
```

FAQ

- Q: How do I convert `TIMESTAMP` values between DataHub and Realtime Compute for Apache Flink?

A: In DataHub, `TIMESTAMP` is a 16-bit UNIX timestamp that is accurate to microseconds. However, in Realtime Compute for Apache Flink, `TIMESTAMP` is 13-bit UNIX timestamp that is accurate to milliseconds. We recommend that you use `BIGINT` for mapping. If you need to use `TIMESTAMP`, we recommend that you use computed columns to perform the conversion. For more information, see [Computed column](#). The following sample code shows the conversion.

```
CREATE TABLE datahub_test (  
  id      VARCHAR,  
  `type`  VARCHAR,  
  `value`  DOUBLE,  
  `time`   BIGINT,  
  ts as TO_TIMESTAMP(`time`/1000),  
  WATERMARK wk for ts as withoffset(ts,2000)  
) WITH (  
  type = 'datahub'  
);
```

- Q: After a DataHub topic was split or scaled in, a Realtime Compute for Apache Flink job failed to run. How do I restore the job?

A: Blink versions earlier than 2.2.0 do not support the DataHub shard scaling feature. If a topic that Realtime Compute for Apache Flink is reading is split or scaled in, the job fails and cannot be automatically restored. To restore the job, you must stop and restart it.

- Q: Can I delete a DataHub topic in use?

A: No, you cannot delete or recreate DataHub topics that are being referenced in Realtime Compute for Apache Flink.

3.1.4.5.2.3. Create a Message Queue for Apache Kafka source table

This topic describes how to create a Message Queue for Apache Kafka source table in Realtime Compute for Apache Flink. It also describes the mappings between the values of the type parameter and Kafka versions, and provides examples on how to parse messages of Message Queue for Apache Kafka.

Notice

- This topic applies only to Blink 2.0 and later.
- This topic applies only to Realtime Compute for Apache Flink that is deployed in exclusive mode.
- You can use Realtime Compute for Apache Flink to read data from a source table of a self-managed Kafka cluster. Before data is read, you must pay attention to the mappings between the values of the type parameter and Kafka versions, and the network configurations of the self-managed Kafka cluster and your Realtime Compute for Apache Flink cluster.

Introduction to the Message Queue for Apache Kafka source table

Message Queue for Apache Kafka is a distributed, high-throughput, and scalable message queue service that is provided by Alibaba Cloud. Message Queue for Apache Kafka is widely used in big data scenarios, such as log collection, monitoring data aggregation, streaming data processing, and online and offline analysis. Realtime Compute for Apache Flink can use Message Queue for Apache Kafka tables as source tables or result tables for the processing of streaming data.

The output data of Message Queue for Apache Kafka is in the serialized VARBINARY format. For each data record obtained from the Message Queue for Apache Kafka source table, you must compile a user-defined table-valued function (UDTF) to parse the data into a data structure before serialization. Realtime Compute for Apache Flink first extracts data from a Message Queue for Apache Kafka source table, compiles a UDTF to parse the data, and then exports the result data to a sink. Flink SQL also allows you to use the CAST function to parse data of the VARBINARY type into data of the VARCHAR type. For more information about UDTFs, see [UDTF](#).

DDL syntax

The DDL definition of a Message Queue for Apache Kafka source table must be the same as the DDL definition in the following SQL statement. You can modify the settings of the parameters in the WITH clause.

```
create table kafka_stream( --The sequence and data types of the following fields must be the same as the five fields in the Message Queue for Apache Kafka source table.
  messageKey VARBINARY,
  `message` VARBINARY,
  topic VARCHAR,
  `partition` INT,
  `offset` BIGINT
) with (
  type='kafka010',
  topic='<yourTopicName>',
  `group.id`='<yourGroupId>',
  ...
);
```

Parameters in the WITH clause

- General configurations

Parameter	Description	Required	Remarks
type	The Kafka version.	Yes	Valid values: Kafka08, Kafka09, Kafka010, and Kafka011. For more information about the mappings between the values of the type parameter and Kafka versions, see Mappings between the values of the type parameter and Kafka versions .
topic	The name of the topic to read.	Yes	None.
topicPattern	The regular expression used to read multiple topics.	No	Topics are separated by vertical bars (), for example, <code>topic1 topic2 topic3</code> . For more information, see Class Pattern .

Parameter	Description	Required	Remarks
startupMode	The start offset for reading data.	No  Note We recommend that you set this parameter based on your business requirements.	Valid values: - GROUP_OFFSETS: reads data by group. This is the default value. - EARLIEST: reads data from the earliest offset in a Kafka cluster. - LATEST: reads data from the latest offset in a Kafka cluster. - TIMESTAMP: reads data from a specified time point. You can specify the startTimeMs or startTime parameter in the WITH clause. The value of the startTimeMs parameter is a timestamp, whereas the value of the startTime parameter is in the <code>yyyy-MM-dd HH:mm:ss</code> format. We recommend that you use the startTimeMs parameter. If neither of the parameters is specified, data is consumed from the start offset specified by Realtime Compute for Apache Flink.  Note - If you do not specify this parameter in plaintext, GROUP_OFFSETS is used by default. - If you set this parameter to TIMESTAMP, you must specify the time zone in job parameters in plaintext. For example, <code>blink.job.timeZone=Asia/Shanghai</code>. - In GROUP_OFFSETS mode, if no offset is configured, data is read from the end of a Kafka partition when a group consumes Kafka data for the first time. - Only Kafka010 and Kafka011 support the TIMESTAMP data type.
partitionDiscoveryIntervalMS	The interval at which new partitions are checked.	No	- Kafka08: By default, new partitions are checked at a specified interval. - Kafka09 or later: The partitionDiscoveryIntervalMS parameter is not supported. You can specify <code>extraConfig='flink.partition-discovery.interval-millis=60000'</code> in the WITH clause to achieve the effect same as the partitionDiscoveryIntervalMS parameter. Unit: milliseconds. Default value: 60000.

Parameter	Description	Required	Remarks
extraConfig	Additional KafkaConsumer configuration items.	No	You want to add configuration items that are required in special scenarios but are not included in the optional configuration items. Example: <code>'fetch.message.max.bytes=104857600'</code> . Multiple configuration items are separated by semicolons (;).

- Configurations for Kafka08
 - Required configurations

Parameter	Description	Required
group.id	The ID of a consumer group.	Yes
zookeeper.connect	The ZooKeeper URL.	Yes

- Optional configurations
 - `consumer.id`
 - `socket.timeout.ms`
 - `fetch.message.max.bytes`
 - `num.consumer.fetchers`
 - `auto.commit.enable`
 - `auto.commit.interval.ms`
 - `queued.max.message.chunks`
 - `rebalance.max.retries`
 - `fetch.min.bytes`
 - `fetch.wait.max.ms`
 - `rebalance.backoff.ms`
 - `refresh.leader.backoff.ms`
 - `auto.offset.reset`
 - `consumer.timeout.ms`
 - `exclude.internal.topics`
 - `partition.assignment.strategy`
 - `client.id`
 - `zookeeper.session.timeout.ms`
 - `zookeeper.connection.timeout.ms`
 - `zookeeper.sync.time.ms`
 - `offsets.storage`
 - `offsets.channel.backoff.ms`
 - `offsets.channel.socket.timeout.ms`
 - `offsets.commit.max.retries`
 - `dual.commit.enabled`
 - `partition.assignment.strategy`
 - `socket.receive.buffer.bytes`
 - `fetch.min.bytes`
- Configurations for Kafka09, Kafka010, and Kafka011
 - Required configurations

Parameter	Description
<code>group.id</code>	The ID of a consumer group.
<code>bootstrap.servers</code>	The endpoint of a Message Queue for Apache Kafka cluster.

- For more information about the optional configurations for Kafka09, Kafka010, and Kafka011, see the following Kafka documentation:
 - [Kafka09](#)
 - [Kafka010](#)
 - [Kafka011](#)

If you want to modify the configurations, you can add parameters to the WITH clause in the DDL statement. For example, if you want to configure Simple Authentication and Security Layer (SASL), add the `security.protocol`, `sasl.mechanism`, and `sasl.jaas.config` parameters.

```
create table kafka_stream(  
  messageKey varbinary,  
  `message` varbinary,  
  topic varchar,  
  `partition` int,  
  `offset` bigint  
) with (  
  type='kafka010',  
  topic='<yourTopicName>',  
  `group.id`='<yourGroupId>',  
  ...,  
  `security.protocol`=SASL_PLAINTEXT,  
  `sasl.mechanism`=PLAIN,  
  `sasl.jaas.config`='org.apache.kafka.common.security.plain.PlainLoginModule required username="<yourUserName>" password="<yourPassword>";'  
);
```

Mappings between the values of the type parameter and Kafka versions

type	Kafka version
Kafka08	0.8.22
Kafka09	0.9.0.1
Kafka010	0.10.2.1
Kafka011	0.11.0.2 and later

Examples of parsing messages of Message Queue for Apache Kafka

- Scenario 1: Realtime Compute for Apache Flink processes the data read from Message Queue for Apache Kafka and exports the result data to ApsaraDB RDS.

Data stored in Message Queue for Apache Kafka is in the JSON format and must be processed by using Realtime Compute for Apache Flink. The following code shows the message format:

```
{
  "name": "Alice",
  "age": 13,
  "grade": "A"
}
```

- Data processing method 1: Realtime Compute for Apache Flink reads and processes data from the Message Queue for Apache Kafka source table and then exports the result data to ApsaraDB RDS.

In Blink 2.2.7 and later, you can use the CAST function to convert the VARBINARY data type into the VARCHAR data type. Then, use the JSON_VALUE function to parse the data of the Message Queue for Apache Kafka source table. The following code shows an example:

```
CREATE TABLE kafka_src (
  messageKey VARBINARY,
  `message` VARBINARY,
  topic VARCHAR,
  `partition` INT,
  `offset` BIGINT
) WITH (
  type = 'kafka010', -- For more information, see Mappings between the values of the type parameter and Kafka versions.
);
CREATE TABLE rds_sink (
  `name` VARCHAR,
  age VARCHAR,
  grade VARCHAR
) WITH (
  type = 'rds'
);
CREATE VIEW input_view AS
SELECT CAST(`message` as VARCHAR) as `message`
FROM kafka_src;
INSERT INTO rds_sink
SELECT
  JSON_VALUE(`message`, '$.name'),
  JSON_VALUE(`message`, '$.age'),
  JSON_VALUE(`message`, '$.grade')
FROM input_view;
```

- Data processing method 2: Realtime Compute for Apache Flink first extracts data from the Message Queue for Apache Kafka source table, compiles a UDTF to parse the data, and then exports the result data to ApsaraDB RDS.

To parse irregular data or complex JSON data, you must compile UDTF code. Examples:

■ SQL

```
-- Define a UDTF to parse messages of Message Queue for Apache Kafka.
CREATE FUNCTION kafkaparser AS 'com.alibaba.kafkaUDTF';
-- Define a Message Queue for Apache Kafka source table. Note that the fields declared in the DDL statement of the Message Queue for Apache Kafka source table must be the same as the fields in the following example. You can modify the settings of the parameters in the WITH clause.
CREATE TABLE kafka_src (
  messageKey VARBINARY,
  `message` VARBINARY
```

```
message VARBINARY,
topic VARCHAR,
`partition` INT,
`offset` BIGINT
) WITH (
type = 'kafka010', -- For more information, see Mappings between the values of the type parameter
and Kafka versions.
topic = 'test_kafka_topic',
`group.id` = 'test_kafka_consumer_group',
bootstrap.servers = 'ip1:port1,ip2:port2,ip3:port3'
);
CREATE TABLE rds_sink (
name VARCHAR,
age INT,
grade VARCHAR,
updateTime TIMESTAMP
) WITH(
type='rds',
url='jdbc:mysql://localhost:3306/test',
tableName='test4',
userName='test',
password='<yourDatabasePassword>'
);
-- Use a UDTF to parse data of the VARBINARY type into formatted data.
CREATE VIEW input_view (
name,
age,
grade,
updateTime
) AS
SELECT
T.name,
T.age,
T.grade,
T.updateTime
FROM
kafka_src as S,
LATERAL TABLE (kafkparser (` message` )) as T (
name,
age,
grade,
updateTime
);
-- Compute the formatted data and export the result data to ApsaraDB RDS.
INSERT INTO rds_sink
SELECT
name,
age,
grade,
updateTime
FROM input_view;
```

■ UDTF

For more information about how to create a UDTF, see [UDTF](#). The following example shows Maven dependencies of Blink 2.2.4.

```
<dependencies>
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-core</artifactId>
    <version>blink-2.2.4-SNAPSHOT</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-streaming-java_2.11</artifactId>
    <version>blink-2.2.4-SNAPSHOT</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-table_2.11</artifactId>
    <version>blink-2.2.4-SNAPSHOT</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.9</version>
  </dependency>
</dependencies>
```

```

package com.alibaba;
import com.alibaba.fastjson.JSONObject;
import org.apache.flink.table.functions.TableFunction;
import org.apache.flink.table.types.DataType;
import org.apache.flink.table.types.DataTypes;
import org.apache.flink.types.Row;
import java.io.UnsupportedEncodingException;
import java.sql.Timestamp;
public class kafkaUDTF extends TableFunction<Row> {
    public void eval(byte[] message) {
        try {
            /* input message :
            {
                "name":"Alice",
                "age":13,
                "grade":"A",
                "updateTime":1544173862
            }
            */
            String msg = new String(message, "UTF-8");
            try {
                JSONObject data = JSON.parseObject(msg);
                if (data != null) {
                    String name = data.getString("name") == null ? "null" : data.getString("name");
                    Integer age = data.getInteger("age") == null ? 0 : data.getInteger("age");
                    String grade = data.getString("grade") == null ? "null" : data.getString("grade");
                    Timestamp updateTime = data.getTimestamp("updateTime");
                    Row row = new Row(4);
                    row.setField(0, name);
                    row.setField(1, age);
                    row.setField(2, grade);
                    row.setField(3, updateTime);
                    System.out.println("Kafka message str ==>" + row.toString());
                    collect(row);
                }
            } catch (ClassCastException e) {
                System.out.println("Input data format error. Input data " + msg + " is not json string");
            }
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
    @Override
    // If the return value is declared as a row, you must reload the UDTF method and specify the types of fields to be returned.
    public DataType getResultType(Object[] arguments, Class[] argTypes) {
        return DataTypes.createRowType(DataTypes.STRING, DataTypes.INT, DataTypes.STRING, DataTypes.TIMESTAMP);
    }
}

```

- Scenario 2: Realtime Compute for Apache Flink reads data from a Message Queue for Apache Kafka source table and then processes the data by using window functions.

Based on the design of Realtime Compute for Apache Flink, you must define watermarks in the DDL statement of a source table for windows such as tumbling and sliding windows. For more information, see [Watermark](#). The method you use to define a watermark in a Message Queue for Apache Kafka source table is different from the method you use for other types of source tables. If you want to perform an event time-based computation by using a window function, you must use a user-defined extension (UDX) to parse the event time in the message field of a source table. Then, you can define a watermark based on the parsed event time. You must use a [Computed column](#) to convert data types for the event time parsed from a Message Queue for Apache Kafka source table. Assume that the data `2018-11-11 00:00:00|1|Anna|female` is written to the Message Queue for Apache Kafka source table. During the computing process, Realtime Compute for Apache Flink first extracts data from the Message Queue for Apache Kafka source table, compiles a UDTF to parse the data, and then exports the result data to ApsaraDB RDS.

- Data processing method 1: Realtime Compute for Apache Flink reads and processes data from the Message Queue for Apache Kafka source table and then exports the result data to ApsaraDB RDS.

In Blink 2.2.7 and later, you can use the CAST function to convert the VARBINARY data type to the VARCHAR data type. Then, use the JSON_VALUE function to parse the data of the Message Queue for Apache Kafka source table. The following code shows an example:

```
CREATE TABLE kafka_src (
  messageKey VARBINARY,
  `message` VARBINARY,
  topic VARCHAR,
  `partition` INT,
  `offset` BIGINT,
  ts as to_timestamp(json_value(cast(`message` as VARCHAR), '$.nodes.time')),
  WATERMARK wk FOR ts as withOffset(ts, 2000)
) WITH (type = 'kafka' -- For more information, see Mappings between the values of the type parameter and Kafka versions.
);
CREATE TABLE rds_sink (
  starttime TIMESTAMP,
  endtime TIMESTAMP,
  `message` BIGINT
) WITH (type = 'rds');
INSERT
  INTO rds_sink
SELECT
  TUMBLE_START(ts, INTERVAL '1' MINUTE),
  TUMBLE_END(ts, INTERVAL '1' MINUTE),
  count(`message`)
FROM
  kafka_src
GROUP BY TUMBLE(ts, INTERVAL '1' MINUTE);
```

- Data processing method 2: Realtime Compute for Apache Flink first extracts data from the Message Queue for Apache Kafka source table, compiles a UDTF to parse the data, and then exports the result data to ApsaraDB RDS.

- SQL

```
-- Define a UDTF to parse messages of Message Queue for Apache Kafka.
CREATE FUNCTION kafkapaser AS 'com.alibaba.kafkaUDTF';
CREATE FUNCTION kafkaUDF AS 'com.alibaba.kafkaUDF';
```

Define a Message Queue for Apache Kafka source table. Note that the fields defined in the DDL statement

```
-- Define a message queue for Apache Kafka source table. Note that the fields defined in the DDL statement must be the same as the fields in the following statement. You can modify the settings of the parameters in the WITH clause.
create table kafka_src (
  messageKey VARBINARY,
  `message` VARBINARY,
  topic VARCHAR,
  `partition` INT,
  `offset` BIGINT,
  ctime AS TO_TIMESTAMP(kafkaUDF(`message`)), -- Define a computed column. A computed column can be considered as a placeholder column that is not stored in a source table. The values in this column are computed. If you want to define a watermark, the data type of the computed column must be TIMESTAMP.
  watermark for `ctime` as withoffset(`ctime`,0) -- Define a watermark in a computed column.
) WITH (
  type = 'kafka010', -- For more information, see Mappings between the values of the type parameter and Kafka versions.
  topic = 'test_kafka_topic',
  `group.id` = 'test_kafka_consumer_group',
  bootstrap.servers = 'ip1:port1,ip2:port2,ip3:port3'
);
create table rds_sink (
  `name` VARCHAR,
  age INT,
  grade VARCHAR,
  updateTime TIMESTAMP
) WITH(
  type='rds',
  url='jdbc:mysql://localhost:3306/test',
  tableName='test4',
  userName='test',
  password='<yourPassword>'
);
-- Use a UDTF to parse data of the VARBINARY type to formatted data.
CREATE VIEW input_view AS
SELECT
  S.ctime,
  T.`order`,
  T.`name`,
  T.sex
from
  kafka_src as S,
  LATERAL TABLE (kafkapaser (`message`)) as T (
    ctime,
    `order`,
    `name`,
    sex
  );
-- Compute the data in input_view.
CREATE VIEW view2 (
  cnt,
  sex
) AS
SELECT
  COUNT(*) as cnt
```



```
SELECT * FROM view1;
T.sex
from
input_view
Group BY sex, TUMBLE(ctime, INTERVAL '1' MINUTE);
-- Compute the formatted data and export the result data to ApsaraDB RDS.
insert into rds_sink
SELECT
cnt,sex
from view2;
```

■ UDF&UDTF

For more information about how to create UDFs and UDTFs, see [UDF](#) and [UDTF](#). The following example shows Maven dependencies of Blink 2.2.4.

```
<dependencies>
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-core</artifactId>
    <version>blink-2.2.4-SNAPSHOT</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-streaming-java_2.11</artifactId>
    <version>blink-2.2.4-SNAPSHOT</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-table_2.11</artifactId>
    <version>blink-2.2.4-SNAPSHOT</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.9</version>
  </dependency>
</dependencies>
```

■ UDTF

```

package com.alibaba;
import com.alibaba.fastjson.JSONObject;
import org.apache.flink.table.functions.TableFunction;
import org.apache.flink.table.types.DataType;
import org.apache.flink.table.types.DataTypes;
import org.apache.flink.types.Row;
import java.io.UnsupportedEncodingException;
/**
 * The following example demonstrates how to parse the JSON strings in a Message Queue for Apache Kafka source table and format the parsed data.
 */
public class kafkaUDTF extends TableFunction<Row> {
    public void eval(byte[] message) {
        try {
            // Read data of the VARBINARY type and convert it to the STRING type.
            String msg = new String(message, "UTF-8");
            // Extract data from JSON strings based on the following fields.
            String ctime = Timestamp.valueOf(data.split("\\|")[0]);
            String order = data.split("\\|")[1];
            String name = data.split("\\|")[2];
            String sex = data.split("\\|")[3];
            // Return rows of data based on the parsed fields.
            Row row = new Row(4);
            row.setField(0, ctime);
            row.setField(1, age);
            row.setField(2, grade);
            row.setField(3, updateTime);
            System.out.println("Kafka message str ==>" + row.toString());
            // Return a row of data.
            collect(row);
        } catch (ClassCastException e) {
            System.out.println("Input data format error. Input data " + msg + "is not json string");
        }
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}

@Override
// If the return value is declared as a row, you must reload the UDTF method and specify the types of fields to be returned.
// Define the data types for objects in output rows.
public DataType getResultType(Object[] arguments, Class[] argTypes) {
    return DataTypes.createRowType(DataTypes.TIMESTAMP,DataTypes.STRING, DataTypes.Integer, DataTypes.STRING,DataTypes.STRING);
}
}

```

■ UDF

```
package com.alibaba;
package com.hjc.test.blink.sql.udx;
import org.apache.flink.table.functions.FunctionContext;
import org.apache.flink.table.functions.ScalarFunction;
public class KafkaUDF extends ScalarFunction {
    // The open method is optional.
    // To implement the open method, you must add "import org.apache.flink.table.functions.FunctionContext;" to the code.
    public String eval(byte[] message) {
        // Read data of the VARBINARY type and convert it to the STRING type.
        String msg = new String(message, "UTF-8");
        return msg.split("\\|")[0];
    }
    public long eval(String b, String c) {
        return eval(b) + eval(c);
    }
    // The close method is optional.
    @Override
    public void close() {
    }
}
```

Create a source table of a self-managed Kafka database

• Example

```
create table kafka_stream(
  messageKey VARBINARY,
  `message` VARBINARY,
  topic varchar,
  `partition` int,
  `offset` bigint
) with (
  type = 'kafka011',
  topic = 'kafka_01',
  `group.id` = 'CID_blink',
  bootstrap.servers = '192.168.0.251:****'
);
```

• Parameters in the WITH clause

For more information, see [Parameters in the WITH clause](#).

Note

- `bootstrap.servers` specifies the address and port number of a self-managed Kafka cluster.
- Only Blink 2.2.6 and later versions support the display of metrics such as transactions per second (TPS) and requests per second (RPS) of Message Queue for Apache Kafka or a self-managed Kafka database.

FAQ

- Q: What do I do if the following error occurs when a job is being started?

```
ERR_ID:
  SQL-00010007
CAUSE:
  Could not create table 'kafka_source' as source table
ACTION:
  Please refer to details section for hint.
  If it doesn't help, please contact customer support
DETAIL:
  java.lang.IllegalArgumentException: Startup time[1566481803000] must be before current time[1566453003356].
```

A: This error is caused by invalid time zone settings. To solve this issue, add the following parameter to job parameters:

```
blink.job.timeZone=Asia/Shanghai
```

- Q: What do I do if a self-managed Kafka cluster cannot consume data?

A:

- Causes

Each broker of the Kafka cluster sends its meta information to ZooKeeper. Then, Kafka consumer accesses the brokers to extract data by using the endpoint in `listener_security_protocol` in the meta information of the brokers. The endpoint is either the IP address or the local server domain name and port number. If the machine where your Realtime Compute for Apache Flink job resides cannot access the endpoint, the consumer of connector cannot extract data. As a result, the data consumption process stops.

- Troubleshooting

- a. View the endpoint for listener `security.protocol` of the broker of ZooKeeper.

```
Command failed: java.lang.IllegalArgumentException: Invalid path string "/ja" caused by invalid character 04
get /brokers/ids/0
{"listener_security_protocol_map":{"PLAINTEXT":"PLAINTEXT"}, "endpoints":{"PLAINTEXT":
  "zxid = 0x248", "jmx_port": 1, "host":
  "ctime = Tue Aug 27 16:38:14 CST 2019", "timestamp": 1566
  "zxid = 0x248", "51", "port":
  "ctime = Tue Aug 27 16:38:14 CST 2019", "version":
  "pzxid = 0x248",
  "cversion = 0",
  "dataVersion = 0",
  "aclVersion = 0",
  "serverIdRange = 0x3f40000000000000"}
```

- Use the network detection feature to check whether you can access the IP address or domain name in the endpoint.
- Log on to your machine to confirm the cause.

- Solutions

- If the endpoint is an IP address, check whether a whitelist is configured for the Kafka server. If no whitelist is configured, configure a whitelist and try again.
- Exclusive clusters of Realtime Compute for Apache Flink cannot resolve domain names. If the endpoint is a domain name and a whitelist has been configured, use one of the following methods:

- If you cannot restart the Kafka service, perform the following operations:

Purchase PrivateZone of Alibaba Cloud DNS and configure domain name resolution for all Kafka brokers. After network detection based on the domain name is successful, restart your Realtime Compute for Apache Flink job.

- If you can restart the Kafka service, perform the following operations:

Configure the IP address and port number (in bootstrap.servers in typical cases) for advertised.listeners for the related broker. Make sure that the network connection is normal. Then, restart your Realtime Compute for Apache Flink job.

- Q: Why does a job terminate immediately after it is started during the consumption of Kafka data?

A: For versions earlier than Blink 3.3.0, if the startupMode parameter is set to TIMESTAMP and no data exists in all partitions of Kafka, the Kafka connector determines that no partition data can be consumed and terminates the job. You can view log information similar to `Consumer subtask {} initially has no partitions to read from.` in the `TaskManager.log` log file that corresponds to the job. We recommend that you upgrade the Blink version to 3.3.0 or later.

- Q: What are the functions of the commit offset mechanism in Realtime Compute for Apache Flink?

A: Realtime Compute for Apache Flink uses `commitOffsetOnCheckpointing` by default. The commit offset policy configured by users does not take effect. After checkpoint generation is enabled, Realtime Compute for Apache Flink commits the currently consumed offset to Kafka each time a checkpoint is generated. This way, data is consumed from the offset committed from the last checkpoint during job restoration. This ensures exactly-once processing of streaming data. If the checkpoint interval exceeds the specified upper limit, Kafka may fail to query the consumed offset.

3.1.4.5.2.4. Create an Oracle database source table

This topic describes how to create an Oracle database source table. It also describes the parameters in the `WITH` clause, data type mapping, sample code, and FAQ involved when you create such a source table.

Notice

- This topic applies only to Blink 3.4.x and later.
- You are allowed to create Oracle database source tables only when Oracle 11g is used.
- You cannot change the number of concurrent jobs for a source node of the Oracle database. By default, only one concurrent job is allowed for a source table.

Syntax

In Realtime Compute for Apache Flink, you can use an Oracle database to store input data. The following code shows an example:

```
create table oracle_source (  
  EMPLOYEE_ID BIGINT,  
  START_DATE TIMESTAMP,  
  END_DATE TIMESTAMP,  
  JOB_ID VARCHAR,  
  DEPARTMENT_ID VARCHAR  
) with (  
  type = 'oracle',  
  url = 'jdbc:oracle:thin:@//127.0.0.1:1521/ORACLE',  
  userName = 'userName',  
  password = 'password',  
  dbName = 'hr',  
  tableName = 'job_history',  
  timeField = 'START_DATE',  
  startTime = '2007-1-1 00:00:00'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to <i>oracle</i> .
url	The database URL.	Yes	Contact Alibaba Cloud O&M engineers.
userName	The username that is used to log on to the database.	Yes	None.
password	The password that is used to log on to the database.	Yes	None.
tableName	The name of the table in the database. Database table names can be in one of the following formats: • <i>Table name 1, Table name 2</i> • <i>Database name. Table name 1, Table name 2</i>  Note Separate table names with commas (,).	Yes	• <i>table1, table2</i> • <i>db1.table1, table2</i>
timeField	The time when the database was updated.	Yes	None.
dbName	The database name.	No	If you have set the <i>tableName</i> parameter, you do not need to set the <i>dbName</i> parameter.

Parameter	Description	Required	Remarks
startTime	The start time of reading data from the source table.	No	<i>2019-5-15 00:00:00</i>
timeZone	The database time zone.	No	<i>Asia/Shanghai", "UTC</i>
queryTimeRangeMs	The time that is taken for data retrieval. Unit: milliseconds.  Note The value of the queryTimeRangeMs parameter must be greater than the value of the queryIntervalMs parameter.	No	Default value: <i>5000</i> .
queryIntervalMs	The interval at which data is queried from the database. Unit: milliseconds.	No	Default value: <i>100</i> .
connectionMaxActive	The maximum number of active connections.	No	Default value: <i>10</i> .
maxRetry	The maximum number of retries upon a connection failure.	No	Default value: <i>3</i> .
escapeFields	Specifies whether to escape field names in the database.	No	Valid values: <i>false</i> (default value): not case-sensitive. <i>true</i>: case-sensitive.

Parameter	Description	Required	Remarks
lengthCheck	The policy for checking the number of fields parsed from a row of data.	No	Valid values: • <i>NONE</i> (default value) ◦ If the number of fields parsed from a row of data is greater than the number of defined fields, data is extracted from left to right based on the order of defined fields. ◦ If the number of fields parsed from a row of data is less than the number of defined fields, the current row is skipped. • <i>SKIP</i>: If the number of fields parsed from a row of data is different from the number of defined fields, the current row is skipped. • <i>EXCEPTION</i>: If the number of fields parsed from a row of data is different from the number of defined fields, an error is returned. • <i>PAD</i>: ◦ If the number of fields parsed from a row of data is greater than the number of defined fields, data is padded from left to right based on the order of defined fields. ◦ If the number of fields parsed from a row of data is less than the number of defined fields, the values of the missing fields are padded with <i>null</i> from left to right.
columnError Debug	Specifies whether debugging is enabled.  Note If debugging is enabled, logs that fail to be parsed are printed.	No	Default value: <i>false</i> .

Field type mapping

Data type of an Oracle database	Data type of Realtime Compute for Apache Flink
• CHAR • VARCHAR • VARCHAR2	VARCHAR

Data type of an Oracle database	Data type of Realtime Compute for Apache Flink
FLOAT	DOUBLE
NUMBER	BIGINT
DECIMAL	DECIMAL

Sample code

The following example demonstrates how to create an Oracle database source table in a Realtime Compute for Apache Flink job.

```
create table oracle_source (  
  EMPLOYEE_ID BIGINT,  
  START_DATE TIMESTAMP,  
  END_DATE TIMESTAMP,  
  JOB_ID VARCHAR,  
  DEPARTMENT_ID VARCHAR  
) with (  
  type = 'oracle',  
  url = 'jdbc:oracle:thin:@//127.0.0.1:1521/ORACLE',  
  userName = 'userName',  
  password = 'password',  
  dbName = 'hr',  
  tableName = 'job_history',  
  timeField = 'START_DATE',  
  startTime = '2007-1-1 00:00:00'  
);  
create table test_out(  
  EMPLOYEE_ID BIGINT,  
  START_DATE TIMESTAMP,  
  END_DATE TIMESTAMP,  
  JOB_ID VARCHAR,  
  DEPARTMENT_ID VARCHAR  
) with (  
  type='print'  
);  
INSERT INTO test_out  
SELECT  
  EMPLOYEE_ID,  
  START_DATE,  
  END_DATE,  
  JOB_ID,  
  DEPARTMENT_ID  
from oracle_source;
```

FAQ

Q: What do I do if no data is found?

A: View the `Round start:{{}}, end:{{}}` and `Round read records` logs on the TaskManager tab. If the logs do not contain data, Blink is faulty.

 Note

- `Round start:[{}],end:[{}]` : displays the start time of the queried data.
- `Round read records` : displays the queried data records.

3.1.4.5.2.5. Create a Log Service source table

This topic describes how to create a Log Service source table in Realtime Compute for Apache Flink. It also describes the attribute fields, parameters in the WITH clause, and data type mapping involved when you create a Log Service source table.

 **Notice** This topic applies only to Blink 1.4.5 and later.

Introduction to Log Service

Log Service is an end-to-end service for log data. The data format of Log Service is similar to JSON. The following code shows an example:

```
{
  "a": 1000,
  "b": 1234,
  "c": "li"
}
```

Log Service is used to store streaming data. Therefore, Realtime Compute for Apache Flink can use the streaming data that is stored in Log Service as input data.

Syntax

The following sample code demonstrates how to create a Log Service source table in a data definition language (DDL) statement. In the code, `sls` indicates Log Service.

```
create table sls_stream(
  a INT,
  b INT,
  c VARCHAR
) with (
  type='sls',
  endPoint='http://cn-hangzhou-share.log.aliyuncs.com',
  accessId='<yourAccessId>',
  accessKey='<yourAccessKey>',
  startTime='2017-07-05 00:00:00',
  project='<yourProjectName>',
  logStore='<yourLogStoreName>',
  consumerGroup='<yourConsumerGroupName>'
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to sls.
endPoint	The endpoint of Log Service.	Yes	Contact Alibaba Cloud O&M engineers.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
project	The name of the Log Service project from which data is read.	Yes	None.
logStore	The name of the Logstore in the Log Service project.	Yes	None.
startTime	The time when a log starts to be consumed.	No	None.
consumerGroup	The name of a consumer group.	No	You can set this parameter as required. The format of the name is not fixed.
heartBeatIntervalMillis	The heartbeat interval of the consumer client.	No	Default value: 10000. Unit: milliseconds.
maxRetryTimes	The maximum number of retries for reading data.	No	Default value: 5.
batchGetSize	The number of log items that are read from a log group at a time.	No	Default value: 100.
columnErrorDebug	Specifies whether to enable debugging.	No	Default value: false. This indicates that debugging is disabled. If you enable debugging, logs that contain parsing exceptions are printed.
			Valid values: • TIMESTAMP: Realtime Compute for Apache Flink starts to consume data from a specified time point in each shard. This value is the default value. • Earliest: Realtime Compute for Apache Flink starts to consume data from the earliest offset in each shard. • Latest: Realtime Compute for Apache Flink starts to consume data from the latest offset in each shard. • Group_Offsets: Realtime Compute for

Parameter	Description	Required	Remarks
startupMode	Specifies whether to enable the consumption mode.	No	Apache Flink preferentially consumes data from the checkpoint that is stored on the server in each shard. You must specify the consumerGroup parameter. The consumption mode varies based on the following scenarios: ◦ If recovery from the Flink state is successful, Realtime Compute for Apache Flink starts to consume data from the checkpoint of the Flink state. ◦ If recovery from the Flink state fails, the consumption mode varies based on the following scenarios: ■ If a checkpoint is available for the specified consumer group, Realtime Compute for Apache Flink attempts to consume data from this checkpoint. ■ If no checkpoint is available for the specified consumer group, the consumption mode varies based on the following scenarios: ■ If you specify startTime, Realtime Compute for Apache Flink starts to consume data at the specified start time. ■ If you do not specify startTime, Realtime Compute for Apache Flink starts to consume data from the earliest offset in each shard.  Notice • Only Blink 3.6.5 and later versions support this parameter. • The preceding configurations take effect only when no checkpoint exists in the state data.

Note

- In Realtime Compute for Apache Flink V1.6.0 and earlier versions, the read performance may be affected if the number of shards in a consumer group is specified. This issue is being rectified.
- Log Service does not support the MAP data type.
- Log Service sets the fields that do not exist to null.
- We recommend that you define the fields in the same order as the fields in the source table. Unordered fields are also supported.
- If input data is in the JSON format, define a separator and use the built-in function `JSON_VALUE` to analyze the data. Otherwise, the following parsing error is returned:

```
2017-12-25 15:24:43,467 WARN [Topology-0 (1/1)] com.alibaba.blink.streaming.connectors.com
mon.source.parse.DefaultSourceCollector - Field missing error, table column number: 3, data co
lumn number: 3, data filed number: 1, data: [{"lg_order_code":"LP00000005","activity_code":"T
EST_CODE1","occur_time":"2017-12-10 00:00:01"}]
```

- The value of the `batchGetSize` parameter must be less than or equal to 1000. Otherwise, an error is returned.
- The `batchGetSize` parameter specifies the number of log items that are read at a time in a log group. If the value of the `batchGetSize` parameter and the size of a single log item that is specified in `LogItem` are large, frequent garbage collections (GCs) may occur. In this case, you must reduce the value of the `batchGetSize` parameter.

Field type mapping

The following table describes the mapping between the field data types of Log Service and Realtime Compute for Apache Flink. We recommend that you declare the mapping in DDL statements.

Data type of Log Service	Data type of Realtime Compute for Apache Flink
STRING	VARCHAR

Attribute fields

By default, Flink SQL supports retrieving three types of Log Service attribute fields. Custom fields are also supported as the input.

Field	Description
<code>__source__</code>	The message source.
<code>__topic__</code>	The message topic.
<code>__timestamp__</code>	The time when a log was generated.

Sample code

```
create table sls_input(  
  a int,  
  b int,  
  c varchar  
) with (  
  type='sls',  
  endPoint='http://cn-hangzhou-share.log.aliyuncs.com',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessKey>',  
  startTime='2017-07-05 00:00:00',  
  project='ali-cloud-streamtest',  
  logStore='stream-test',  
  consumerGroup='consumerGroupTest1'  
);  
create table print_output(  
  a int,  
  b int,  
  c varchar  
) with (  
  type='print'  
);  
INSERT INTO print_output  
SELECT  
  a, b, c  
from sls_input;
```

FAQ

- Q: Why does the overall latency of a job increase, or why is no output generated for the job that has window aggregation?

A: This issue occurs if no new data is written into a partition. To solve this issue, change the parallelism to be the same as the number of read and write partitions.

- Q: How do I set the parallelism?

A: We recommend that you set the parallelism to the number of partitions. Otherwise, if the speeds at which data is read from two partitions vary significantly, data may be filtered out or data latency may occur when you set the start offset for a job to a time prior to the present time.

- Q: How do I troubleshoot the issue that the latency of a Flink job increases?

A: The Log Service source table may be sharded. Shard indexes may not be continuous after sharding, which increases the latency of a Flink job. If you find that the latency of a Flink job increases, check whether the Log Service source table is sharded.

3.1.4.5.2.6. Create a full MaxCompute source table

This topic describes how to create a full MaxCompute source table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause and data type mapping involved when you create a full MaxCompute source table.

Notice

- This topic applies only to Blink 2.2.7 and later.
- Full MaxCompute source tables are typically used as bounded stream tables. This makes MaxCompute different from other data sources such as DataHub and Kafka. In Blink 3.4.4, you can specify a full MaxCompute source table as an unbounded stream table. This way, the source table can continuously listen to the generation of new partitions. If a new partition is generated, Realtime Compute for Apache Flink reads data from the new partition. This feature is deprecated in Blink 3.5.0. To use a MaxCompute source table as an unbounded stream table, see [Create an incremental MaxCompute source table](#).

DDL syntax

In Realtime Compute for Apache Flink, you can use MaxCompute to store input data. The following code shows an example:

```
create table odps_source(  
  id INT,  
  user_name VARCHAR,  
  content VARCHAR  
) with (  
  type = 'odps',  
  endPoint = 'http://service.cn.maxcompute.aliyun-inc.com/api',  
  project = '<projectName>',  
  tableName = '<tableName>',  
  accessId = '<yourAccessKeyId>',  
  accessKey = '<yourAccessKeySecret>',  
  `partition` = 'ds=2018****' --If your MaxCompute source table is a non-partitioned table, you do not need  
  to declare this parameter.  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
endPoint	The endpoint of MaxCompute.	Yes	Contact Alibaba Cloud O&M engineers.
tunnelEndpoint	The endpoint of the MaxCompute Tunnel service.	No	Contact Alibaba Cloud O&M engineers.
project	The name of a MaxCompute project.	Yes	None.
tableName	The name of the MaxCompute source table.	Yes	None.

Parameter	Description	Required	Remarks
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
partition	The name of a partition.	No	- A full MaxCompute source table that contains only one partition For example, if only one partition key column <code>ds</code> exists, <code>`partition`='ds=20180905'</code> indicates that data in the <code>ds=20180905</code> partition is read. - A full MaxCompute source table that contains multiple partitions For example, if two partition key columns <code>ds</code> and <code>hh</code> exist, <code>`partition`='ds=20180905.hh=*'</code> indicates that data in the <code>ds=20180905</code> partition is read. Note You must declare the values of all partitions when you filter partitions. In the preceding example, if you declare only <code>'partition'='ds=20180905'</code>, no partition is read.
subscribeNewPartition	Specifies whether to listen to new partitions that meet specific conditions.	No	Default value: false. This value indicates that the system does not listen to new partitions. Note - If the <code>subscribeNewPartition</code> parameter is set to <code>true</code>, you cannot specify the value of the partition parameter. Otherwise, new partitions cannot be read. - This parameter is provided only in Blink 3.4.4. The parameter is deprecated in Blink 3.5.0. If you need to use this parameter, create an incremental MaxCompute source table. For more information, see Create an incremental MaxCompute source table.

Parameter	Description	Required	Remarks
subscribeIntervalInSec	The interval at which new partitions are listened to.	No	Default value: 30. Unit: seconds.  Note If the value of this parameter is too small, pressure may be caused on the MaxCompute metadata service. This may result in failures to listen to the service.

Field type mapping

MaxCompute data type	Data type of Realtime Compute for Apache Flink
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
DECIMAL	DECIMAL
BINARY	VARBINARY
STRING	VARCHAR

Sample code

The following sample code demonstrates how to create a full MaxCompute source table in a Realtime Compute for Apache Flink job.

```
CREATE TABLE odps_source (  
  cid varchar,  
  rt DOUBLE,  
) with (  
  type = 'odps',  
  endPoint = '<yourEndpointName>',  
  project = '<yourProjectName>',  
  tableName = '<yourTableName>',  
  accessId = '<yourAccessId>',  
  accessKey = '<yourAccessPassword>',  
  partition = 'ds=20190712'  
);  
CREATE TABLE test (  
  cid varchar,  
  invoke_count BIGINT  
) with (  
  type='print'  
);  
INSERT INTO test  
SELECT  
  cid,  
  count(*) as invoke_count  
FROM odps_source GROUP BY cid;
```

3.1.4.5.2.7. Create a Message Queue source table

This topic describes how to create a Message Queue source table in Realtime Compute for Apache Flink. It also describes the comma-separated values (CSV) file format, parameters in the WITH clause, and data type mapping used when you create a Message Queue source table.

 **Note** If you need to use Message Queue with separate namespaces, use Blink 3.X.

Introduction to Message Queue

Message Queue is a professional messaging middleware that is developed by Alibaba Cloud. It is a core product of the enterprise-level Internet architecture. In Realtime Compute for Apache Flink, you can use Message Queue tables as source tables for the processing of streaming data.

Sample code

```
create table mq_stream(  
  x varchar,  
  y varchar,  
  z varchar  
) with (  
  type='mq',  
  topic='<yourTopicName>',  
  endpoint='<yourEndpoint>',  
  pullIntervalMs='1000',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessSecret>',  
  startMessageOffset='1000',  
  consumerGroup='<yourConsumerGroup>',  
  fieldDelimiter='|'  
);
```

 **Note** Message Queue stores unstructured data. You do not need to define schemas for Message Queue source tables. Realtime Compute for Apache Flink supports messages in the CSV and binary formats.

CSV format

Assume that you have a Message Queue message in the following CSV format:

```
1,name,male  
2,name,female
```

 **Note** A Message Queue message can contain any number of data records. Multiple data records are separated by `\n`.

You can use the following DDL statement to declare a Message Queue source table in a Realtime Compute for Apache Flink job:

```
create table mq_stream(  
  id varchar,  
  name varchar,  
  gender varchar  
) with (  
  type='mq',  
  topic='<yourTopicName>',  
  endpoint='<ourEndpoint>',  
  pullIntervalMs='1000',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessSecret>',  
  startMessageOffset='1000',  
  consumerGroup='<yourConsumerGroup>',  
  fieldDelimiter='|'  
);
```

Binary format

For binary-formatted messages, you can use the following sample code to create a Message Queue source table:

```
create table source_table (
  mess varbinary
) with (
  type = 'mq',
  endpoint = '<yourEndpoint>',
  pullIntervalMs='500',
  accessId='<yourAccessId>',
  accessKey='<yourAccessSecret>',
  topic = '<yourTopicName>',
  consumerGroup='<yourConsumerGroup>'
);
create table out_table (
  commodity varchar
) with (
  type='print'
);
INSERT INTO out_table
SELECT
  cast(mess as varchar)
FROM source_table;
```

Note

- The `cast (mess as varbinary)` method is supported only in Realtime Compute for Apache Flink V2.0 and later. If your Realtime Compute for Apache Flink version is earlier than V2.0, upgrade it first.
- Data of the VARBINARY type can be passed in only once.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to mq.
topic	The name of a topic.	Yes	None.
endPoint	The endpoint of Message Queue.	Yes	Contact Alibaba Cloud O&M engineers.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
consumerGroup	The name of a consumer group.	Yes	None.

Parameter	Description	Required	Remarks
pullIntervalMs	The interval at which messages are pulled.	No	Unit: milliseconds.
startTime	The start time of message consumption.	No	None.
startMessageOffset	The offset from the start time.	No	This parameter is optional. If you set this parameter, messages are consumed from the time point specified by this parameter.
tag	A subscription tag.	No	None.
lineDelimiter	A row delimiter that is used to parse message blocks.	No	Default value: <code>\n</code> .
fieldDelimiter	A field delimiter.	No	Default value: <code>\u0001</code> . <code>\u0001</code> is the field delimiter in read-only mode, and <code>^A</code> is the field delimiter in edit mode. <code>\u0001</code> is invisible in read-only mode.
encoding	The encoding format.	No	Default value: <code>utf-8</code> .

Parameter	Description	Required	Remarks
lengthCheck	The policy for checking the number of fields parsed from a row of data.	No	Default value: NONE. • If the number of fields parsed from a row of data is greater than the number of defined fields, data is extracted from left to right based on the order of defined fields. • If the number of fields parsed from a row of data is less than the number of defined fields, this row of data is skipped. Other valid values are SKIP, EXCEPTION, and PAD. • SKIP: If the number of fields parsed from a row of data does not match the number of defined fields, this row of data is skipped. • EXCEPTION: If the number of fields parsed from a row of data does not match the number of defined fields, an exception is reported. • PAD: Data is padded from left to right based on the order of defined fields. ◦ If the number of fields parsed from a row of data is greater than the number of defined fields, data is padded from left to right based on the order of defined fields. ◦ If the number of fields parsed from a row of data is less than the number of defined fields, the values of the missing fields are padded with null.
columnErrorDebug	Specifies whether to enable debugging.	No	Default value: FALSE. If you set this parameter to TRUE, a log entry is printed when a parsing exception occurs.
instanceID	The ID of a Message Queue instance.	No	If the Message Queue instance does not have separate namespaces, you do not need to set this parameter. If the Message Queue instance has separate namespaces, this parameter is required.

Mapping between field data types

Message Queue data type	Data type of Realtime Compute for Apache Flink
STRING	VARCHAR

3.1.4.5.2.8. Create a Tablestore source table

This topic describes how to create a Tablestore source table in Realtime Compute for Apache Flink.

 **Notice** This topic applies only to Realtime Compute for Apache Flink V3.2.2 and later.

Introduction to Tablestore

Tablestore is a distributed NoSQL database service that is built on the Apsara distributed operating system of Alibaba Cloud. Tablestore adopts data sharding and load balancing technologies to scale out and handle concurrent transactions. You can use Tablestore to store and query a large amount of structured data in real time.

Tunnel Service of Tablestore

Tunnel Service is a centralized service that uses the Tablestore API to allow you to consume full and incremental data. You can use the Tunnel Service API and SDKs to create tunnels from which you can consume distributed data in real time. The distributed data is divided into the following types: incremental data, full data, and full and incremental data. You can use the tunnels of the Tunnel Service to consume existing data or added data in tables in streaming processing mode. In Realtime Compute for Apache Flink, the tunnels of the Tunnel Service can serve as the source of streaming data. Each data record uses a JSON-like format. You can run the following code if you need to use Tunnel Service tunnels:

```
{
  "OtsRecordType": "PUT", // The data operation type, such as PUT, UPDATE, and DELETE.
  "OtsRecordTimestamp": 1506416585740836, // The time when the data is written. The time unit is milliseconds. The value 0 indicates that full data is written.
  "PrimaryKey": [
    {
      "ColumnName": "pk_1", // The first primary key column.
      "Value": 1506416585881590900
    },
    {
      "ColumnName": "pk_2", // The second primary key column.
      "Value": "string_pk_value"
    }
  ],
  "Columns": [
    {
      "OtsColumnType": "Put", // The operation type for the column, such as PUT, DELETE_ONE_VERSION, and DELETE_ALL_VERSION.
      "ColumnName": "attr_0",
      "Value": "hello_table_store",
    },
    {
      "OtsColumnType": "DELETE_ONE_VERSION", // No value is specified for the delete operation.
      "ColumnName": "attr_1"
    }
  ]
}
```

DDL syntax

In Realtime Compute for Apache Flink, you can use Tablestore to store input data. The following code shows an example:

```
create table tablestore_stream(  
  pk_1 BIGINT,  
  pk_2 VARCHAR,  
  attr_0 VARCHAR,  
  attr_1 DOUBLE,  
  OtsRecordType VARCHAR HEADER //You must add HEADER to the attribute field.  
) with (  
  type='ots',  
  endPoint='http://blink-demo.cn-hangzhou.vpc.tablestore.aliyuncs.com',  
  instanceName='blink-demo',  
  tableName='demo_table',  
  tunnelName='blink-demo-stream',  
  accessId='<yourAccessID>',  
  accessKey='<yourAccessSecret>',  
  ignoreDelete='false' // Specifies whether to ignore the delete operation.  
);
```

Attribute fields

Field	Description
OtsRecordType	The data operation type.
OtsRecordTimestamp	The data operation time. If you set this parameter to 0, full data is written.
<Column name>_OtsColumnType	The operation type for a column.

Parameters in the WITH clause

Parameter	Description	Remarks
type	The type of the connector.	Set the value to <code>ots</code> .
endPoint	The endpoint of the Tablestore instance.	Contact Alibaba Cloud O&M engineers.
instanceName	The name of the Tablestore instance.	None.

Parameter	Description	Remarks
tableName	The name of the Tablestore table.	Realtime Compute for Apache Flink does not reread data that has been read from a Tablestore source table. If you want Realtime Compute for Apache Flink to reread full data from the source table, you must create a data tunnel.
tunnelName	The tunnel name of the Tablestore source table.	None.
accessId	The AccessKey ID that is used to access Tablestore.	None.
accessKey	The AccessKey secret that is used to access Tablestore.	None.
ignoreDelete	Specifies whether to ignore the delete operation.	Optional. Default value: <code>false</code> .

3.1.4.5.2.9. Create a Hologres source table

This topic describes how to create a Hologres source table. It also describes the data definition language (DDL) syntax, parameters in the WITH clause, data type mapping, and sample code used when you create a Hologres source table.

Notice

- We recommend that you use Hologres 0.7 or later.
- You can use only the Hologres tables that use the row store structure as source tables for Realtime Compute for Apache Flink.
- Hologres source tables support projection pushdown. This allows you to read data from only the required columns in the Hologres source tables.
- Concurrent Blink jobs can read data from one or more Hologres shards. We recommend that the number of concurrent Blink jobs be no more than the number of Hologres shards.
- You can use Hologres source tables to process streaming data and batch data.
- Blink jobs execute snapshot statements to read existing data from Hologres source tables at a high rate. After the read operation is complete, the jobs end. If the jobs fail to read data, they read the data again.

Introduction to Hologres

Hologres is compatible with the PostgreSQL protocol and closely connected to the big data ecosystem. Hologres allows you to analyze and process petabytes of data in high concurrency and low latency scenarios. Hologres provides an easy method for you to use the existing business intelligence (BI) tools to perform multidimensional analysis and explore your business.

DDL syntax

```
create table mysource(  
  name varchar,  
  age BIGINT,  
  birthday BIGINT  
) with (  
  type='hologres',  
  dbname='...',  
  tablename='...',  
  username='...',  
  password='...',  
  endpoint='...',  
  field_delimiter='...' -- This parameter is optional.  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to <i>hologres</i> .
dbname	The name of the database.	Yes	None.
tablename	The name of the table.	Yes	None.
username	The username that is used to log on to the database. You must enter the AccessKey ID of your Alibaba Cloud account.	Yes	None.
password	The password that is used to log on to the database. You must enter the AccessKey secret of your Alibaba Cloud account.	Yes	None.
endpoint	The endpoint of Hologres.	Yes	Contact Alibaba Cloud O&M engineers.
field_delimiter	The delimiter used between rows when data is being exported.  Notice Delimiters cannot be inserted into the data.	Yes	Default value: <code>"\u0002"</code> .

Field type mapping

Hologres	BLINK
INT	INT
INT[]	ARRAY<INT>
BIGINT	BIGINT
BIGINT[]	ARRAY<BIGINT>
REAL	FLOAT
REAL[]	ARRAY<FLOAT>
DOUBLE PRECISION	DOUBLE
DOUBLE PRECISION[]	ARRAY<DOUBLE>
BOOLEAN	BOOLEAN
BOOLEAN[]	ARRAY<BOOLEAN>
TEXT	VARCHAR
TEXT[]	ARRAY<VARCHAR>
NUMERIC	DECIMAL
DATE	DATE
TIMESTAMP WITH TIMEZONE	TIMESTAMP

Sample code

The following sample code demonstrates how to create a Hologres source table in a Realtime Compute for Apache Flink job.

```
create table mysource(  
  name varchar,  
  age BIGINT,  
  birthday BIGINT  
) with (  
  type='hologres',  
  dbname='...',  
  tablename='...',  
  username='...',  
  password='...',  
  endpoint='...',  
  field_delimiter='...' -- This parameter is optional.  
);  
create table print_output(  
  a varchar,  
  b BIGINT,  
  c BIGINT  
) with (  
  type='print'  
);  
INSERT INTO print_output  
SELECT  
  a, b, c  
from mysource;
```

3.1.4.5.2.10. Create an incremental MaxCompute source table

This topic describes how to create an incremental MaxCompute source table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause, data type mapping, and FAQ involved when you create an incremental MaxCompute source table.

Notice

- This topic applies only to Blink 3.5.0-hotfix.
- Incremental MaxCompute source tables cannot be used as dimension tables.
- Incremental MaxCompute source tables support only partitioned tables.

Syntax

```
create table odps_source(
  id int,
  user_name VARCHAR,
  content VARCHAR
) with (
  type = 'continuous-odps',
  endPoint = 'your_end_point_name',
  project = 'your_project_name',
  tableName = 'your_table_name',
  accessId = 'your_access_id',
  accessKey = 'your_access_key',
  startPartition = 'ds=20180905'
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the connector.	Yes	Set the value to continuous-odps .
endPoint	The endpoint of MaxCompute.	Yes	Contact Alibaba Cloud O&M engineers.
tunnelEndpoint	The endpoint of the MaxCompute Tunnel service.	- Yes: This parameter is required if MaxCompute is deployed in a VPC. - No: This parameter is not required if MaxCompute is not deployed in a VPC.	Contact Alibaba Cloud O&M engineers.
project	The name of the project to which the table belongs.	Yes	None.
tableName	The name of the table.	Yes	None.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
	The partition in which data starts to be read. When the system loads partitioned tables, it compares startPartition and all		

Parameter	Description	Required	Remarks
startPartition	partitions in each partitioned table in alphabetical order and then loads the data that meets a specified condition from the partitions. An incremental MaxCompute source table can also continuously listen to incremental MaxCompute partitioned tables. After the source table reads data from all the existing partitions, it continues to listen to the generation of new partitions. If a new partition is generated, the source table reads data from the new partition.  Note - An incremental MaxCompute source table must have level-1 partitions. Level-2 partitions are optional. - If you specify a level-2 partition, make sure that it is placed after a level-1 partition. - If the partition specified by the startPartition parameter does not exist, the system reads data from the next partition.	Yes	For example, if startPartition is set to ds=20191201, data of all partitions that meets the condition of ds >= 20191201 in the incremental MaxCompute partitioned table is loaded. Assume that an incremental MaxCompute partitioned table contains five partitions. Each partition has two partition key columns with a level-1 partition of ds and level-2 partition of type. - ds=20191201,type=a - ds=20191201,type=b - ds=20191202,type=a - ds=20191202,type=b - ds=20191202,type=c If the values of startPartition for the partitions are different, the following partitions meet the preceding condition: - ds=20191202 - ds=20191202,type=a - ds=20191202,type=b - ds=20191202,type=c - ds=20191201,type=c - ds=20191202,type=a - ds=20191202,type=b - ds=20191202,type=c

Field type mapping

MaxCompute data type	Data type of Realtime Compute for Apache Flink
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
DECIMAL	DECIMAL
BINARY	VARBINARY
STRING	VARCHAR



Notice

- Incremental MaxCompute source tables do not support the CHAR, VARCHAR, ARRAY, MAP, or STRUCT data types.
- You can use the STRING data type instead of the VARCHAR data type.

Sample code

One partition is generated in an incremental MaxCompute source table every day. The partition key column is ds. The incremental MaxCompute source table loads data from partitions whose numbers are greater than or equal to 20191201 and continuously listens to the generation of new partitions.

```
-- The incremental MaxCompute source table reads data from partitions in the range of [ds=20191201, ∞).
CREATE TABLE odps_source (
  cid VARCHAR,
  rt DOUBLE,
) with (
  type = 'continuous-odps',
  endPoint = 'your_end_point_name',
  project = 'your_project_name',
  tableName = 'your_table_name',
  accessId = 'xxxx',
  accessKey = 'xxxx',
  startPartition = 'ds=20191201'
);
CREATE TABLE test (
  cid VARCHAR,
  rt DOUBLE,
) with (
  type='print'
);
INSERT INTO test
SELECT
  cid, rt FROM odps_source;
```

3.1.4.5.3. Create a data result table

3.1.4.5.3.1. Overview

Realtime Compute uses the `CREATE TABLE` statement to define the schema of a result table for output data and how data is written into a target data store.

Data can be written by using either of the following methods: append and update.

- Append: If the result table is stored in a relational database with no primary key defined, Log Service, or Message Queue, output data is appended to the result table.
- Update: If the result table is stored in a relational database or a Hadoop database with a primary key defined, output data is written into the result table as follows:
 - If the value for the primary key is not found in the result table, the data record is inserted into the database.
 - If the value for the primary key is found, the existing data record in the database is overwritten.

Result table syntax

```
CREATE TABLE tableName
(columnName dataType [, columnName dataType]*)
[ WITH (propertyName=propertyValue [, propertyName=propertyValue]*) ];
```

Example


```
CREATE TABLE rds_output(  
  id INT,  
  len INT,  
  content VARCHAR,  
  PRIMARY KEY(id)  
) WITH (  
  type='rds',  
  url='yourDatabaseURL',  
  tableName='yourTableName',  
  userName='yourDatabaseUserName',  
  password='yourDatabasePassword'  
);
```

3.1.4.5.3.2. Create a DataHub result table

This topic describes how to create a DataHub result table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause and data type mapping involved when you create a DataHub result table.

 **Notice** This topic applies only to Blink 1.4.5 and later.

Introduction to DataHub

DataHub is a real-time data distribution platform designed to process streaming data. You can publish and subscribe to applications for streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data. Realtime Compute for Apache Flink uses DataHub to store source and result tables for streaming data processing.

DDL syntax

In Realtime Compute for Apache Flink, you can use DataHub to store output data. In the following sample code, the data definition language (DDL) statement creates a DataHub result table:

```
create table datahub_output(  
  name VARCHAR,  
  age BIGINT,  
  birthday BIGINT  
)with(  
  type='datahub',  
  endPoint='<yourEndpoint>',  
  project='<yourProjectName>',  
  topic='<yourTopicName>',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessKey>',  
  batchSize='<yourBatchSize>',  
  batchWriteTimeoutMs='1000'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to <code>datahub</code> .
endPoint	The endpoint of DataHub.	Yes	Contact Alibaba Cloud O&M engineers.
project	The name of a DataHub project.	Yes	None.
topic	The name of a DataHub topic.	Yes	None.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
maxRetryTimes	The maximum number of retries for reading data from DataHub.	No	The default value of this parameter varies based on the Blink versions. - For versions earlier than Blink 2.2.7, the default value is 3. - For Blink 2.2.7 and later, the default value is 20.
batchSize	The number of data records that can be written at a time.	No	The default value of this parameter varies based on the Blink versions. - For versions earlier than Blink 3.3, the default value is 300. - For Blink 3.3 and later, the default value is 100.
batchWriteTimeoutMs	The write timeout period.	No	Optional. Default value: 5000. Unit: milliseconds. This value indicates that if the number of input data records does not reach the data output condition within 5,000 milliseconds, all cached data is written into the result table.
maxBlockMessages	The maximum number of data blocks that are written at a time.	No	Default value: 100.
reserveMilliSecond	Specifies whether to reserve the millisecond component in a value of the <code>TIMESTAMP</code> data type.	No	Default value: false.  Notice This feature is available only in Realtime Compute for Apache Flink V2.2.6 and later.

Parameter	Description	Required	Remarks
partitionBy	The partitioning rule for the result table. Before Realtime Compute for Apache Flink writes data into the result table, Realtime Compute for Apache Flink performs hash partitioning based on the value of this parameter. Then, the data flows to the related partition.	No	This parameter is empty by default. This indicates that data is randomly allocated.  Note The partitionBy parameter determines the Blink subtask to which data flows.
hashFields	The names of columns. After column names are specified, the values of the same column are written into the same shard.	No	Default value: Null. This indicates that data is written randomly. You can specify multiple column values and separate them with commas (,). for example, <code>hashFields='a,b'</code>.  Notice - This feature is available only in Blink 3.3.0 and later. - The hashFields parameter determines the shard into which a data stream is written.

Field type mapping

The following table lists the mappings between the data types of DataHub and Realtime Compute for Apache Flink. We recommend that you declare the mapping in DDL statements.

Data type of DataHub	Data type of Realtime Compute for Apache Flink
BIGINT	BIGINT

Data type of DataHub	Data type of Realtime Compute for Apache Flink
TIMESTAMP	BIGINT  Note In DataHub, TIMESTAMP is a 16-bit UNIX timestamp that is accurate to microseconds. However, in Realtime Compute for Apache Flink, TIMESTAMP is 13-bit UNIX timestamp that is accurate to milliseconds. We recommend that you use BIGINT for mapping. If you need to use TIMESTAMP, we recommend that you use computed columns for conversion. For more information, see Computed column.
STRING	VARCHAR
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DECIMAL	DECIMAL

Sample code

The following sample code demonstrates how to create a DataHub result table in a Realtime Compute for Apache Flink job:

```
create table datahub_input(  
  name VARCHAR  
) with (  
  type='datahub',  
  endPoint='http://dh-cn-hangzhou.aliyun-inc.com',  
  project='test1',  
  topic='topic1',  
  accessId='<yourAccessID>',  
  accessKey='<yourAccessSecret>',  
  startTime='2018-06-01 00:00:00'  
);  
create table datahub_output(  
  name varchar  
)with(  
  type='datahub',  
  endPoint='http://dh-cn-hangzhou.aliyun-inc.com',  
  project='test2',  
  topic='topic2',  
  accessId='<yourAccessID>',  
  accessKey='<yourAccessSecret>',  
  batchSize='1000',  
  batchWriteTimeoutMs='500'  
);  
INSERT INTO datahub_output  
SELECT  
  LOWER(name)  
from datahub_input;
```

3.1.4.5.3.3. Create a Message Queue result table

This topic describes how to create a Message Queue result table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause used when you create a Message Queue result table.



Notice

- This topic applies only to Blink 1.4.5 and later.
- If you need to use Message Queue that has separate namespaces, use Blink 3.X.

Introduction to Message Queue

Message Queue is a professional message middleware that is developed by Alibaba Cloud for commercial use. It is a core product for the enterprise-level Internet architecture. Based on the high-availability distributed cluster technology, Message Queue provides comprehensive cloud messaging services, including message publishing and subscription, message tracing, resource statistics, message scheduling or delaying, monitoring, and alerts.

CSV format

Realtime Compute for Apache Flink writes streaming data to Message Queue result tables for output. In the following sample code, the data definition language (DDL) statement creates a Message Queue result table to store streaming data in the CSV format:

```
CREATE TABLE stream_test_hotline_agent (  
  id INTEGER,  
  len BIGINT,  
  content VARCHAR  
) WITH (  
  type='mq',  
  endpoint='<yourEndpoint>',  
  accessID='<yourAccessId>',  
  accessKey='<yourAccessSecret>',  
  topic='<yourTopicName>',  
  producerGroup='<yourGroupName>',  
  tag='<yourTagName>',  
  encoding='utf-8',  
  fieldDelimiter=',',  
  retryTimes='5',  
  sleepTimeMs='500'  
);
```

Binary format

Realtime Compute for Apache Flink writes streaming data to Message Queue result tables for output. In the following sample code, the DDL statement creates a Message Queue result table to store streaming data in the binary format:

```
CREATE TABLE source_table (  
  commodity VARCHAR  
) WITH (  
  type='random'  
);  
CREATE TABLE result_table (  
  mess VARBINARY  
) WITH (  
  type = 'mq',  
  endpoint='<yourEndpoint>',  
  accessID='<yourAccessId>',  
  accessKey='<yourAccessSecret>',  
  topic='<yourTopicName>',  
  producerGroup='<yourGroupName>'  
);  
INSERT INTO result_table  
SELECT  
  CAST(SUBSTRING(commodity,0,5) AS VARBINARY) AS mess  
FROM source_table;
```

 **Note** Blink 2.0 or later supports `cast(varchar as varbinary)`. If the Blink version is earlier than 2.0, upgrade the Blink version first.

Parameters in the WITH clause

Parameter	Description	Remarks
type	The type of the result table.	Set the value to mq.
topic	The name of the Message Queue topic to which data is written.	None.
endpoint	The endpoint that is used to access Message Queue.	Contact Alibaba Cloud O&M engineers.
accessID	AccessKey ID	None.
accessKey	AccessKey Secret	None.
producerGroup	The name of the producer group.	None.
tag	The message tag.	Optional. This parameter is empty by default.
fieldDelimiter	The field delimiter.	Optional. Default value: <code>\u0001</code> . The delimiter varies based on the following modes: - In read-only mode, the <code>\u0001</code> delimiter is used. <code>\u0001</code> is invisible in read-only mode. - In edit mode, the <code>^A</code> delimiter is used.
encoding	The encoding format.	Optional. Default value: <code>utf-8</code> .
retryTimes	The maximum number of retries for writing data to the table.	Optional. Default value: 10.
sleepTimeMs	The retry interval.	Optional. Default value: 1000. Unit: milliseconds.
instanceID	The ID of a Message Queue instance.	- If the Message Queue instance does not have separate namespaces, you do not need to set this parameter. - If the Message Queue instance has separate namespaces, the instanceID parameter is required.

3.1.4.5.3.4. Create an ApsaraDB RDS result table

This topic describes how to create an ApsaraDB RDS or Distributed Relational Database Service (DRDS) result table in Realtime Compute. It also describes the mapping between data types of Realtime Compute and ApsaraDB RDS or DRDS data types.

Introduction to ApsaraDB RDS

ApsaraDB RDS is a stable, reliable, and scalable online database service. Based on Apsara Distributed File System and high-performance storage, ApsaraDB RDS supports a variety of engines such as MySQL, SQL Server, PostgreSQL, and PostgreSQL Plus Advanced Server (PPAS). PPAS is compatible with Oracle databases. ApsaraDB RDS provides comprehensive solutions for database operations and management, such as data backup, data recovery, monitoring, and data migration.

 **Note** ApsaraDB RDS and DRDS use the same parameters in the WITH clause. When you use ApsaraDB RDS or DRDS to store result tables, make sure that there are real tables.

DDL definition

The following sample code demonstrates how to create an ApsaraDB RDS or DRDS result table. Currently, only ApsaraDB RDS for MySQL is supported.

```
create table rds_output(  
  id int,  
  len int,  
  content varchar,  
  primary key(id,len)  
) with (  
  type='rds',  
  url='<yourDatabaseURL>',  
  tableName='<yourDatabaseTableName>',  
  userName='<yourDatabaseUserName>',  
  password='<yourDatabasePassword>'  
);
```

- In Realtime Compute, each row of output data is converted to a row of SQL statement and then written and executed in the destination database. If you want to write multiple rows of output data into the destination database, you must add ?rewriteBatchedStatements=true to the end of the URL to improve write performance.
- You can define an auto-increment primary key for an ApsaraDB RDS for MySQL database. If you want to use the auto-increment primary key, do not declare the auto-increment field in the DDL statement. For example, if you use ID as an auto-increment field, do not declare the ID field in the DDL statement. When a row of output data is written into the ApsaraDB RDS for MySQL database, a value is automatically filled for the auto-increment field.
- If a DRDS result table is partitioned, the partition key must be declared in primary key() of the DDL statement. Otherwise, you cannot write data into the partitioned table.

Parameters in the WITH clause

Parameter	Description	Remarks
url	The database URL.	None.
tableName	The name of the table in the database.	None.

Parameter	Description	Remarks
userName	The username that is used to log on to the database.	None.
password	The password that is used to log on to the database.	None.
maxRetryTimes	The maximum number of retries for writing data to the table.	Optional. Default value: 3.
batchSize	The number of data records that are written at a time.	Optional. Default value: 50.
bufferSize	The maximum number of data records that can be stored in the buffer before deduplication is triggered. This parameter setting takes effect only after the primary key is specified.	Optional. Default value: 1. This value indicates that deduplication is triggered each time a data record is stored. In versions of Realtime Compute later than 1.4.1, the default value is 1000.
flushIntervalMs	The time interval at which the cache is cleared.	Optional. Default value: 5000. Unit: milliseconds. This value indicates that if no data is written within 5,000 milliseconds, all cached data is written into the result table.
excludeUpdateColumns	The columns that are excluded when you update the result table based on key values.	Optional. By default, this parameter is not specified, which indicates that primary key fields are excluded.
ignoreDelete	Specifies whether to skip delete operations.	Default value: false.
partitionBy	The columns based on which Realtime Compute performs hash partitioning. Before writing data to the sink operator, Realtime Compute performs hash partitioning based on the value of this parameter. The data then flows to the corresponding hash operator.	Optional. This parameter is not specified by default.

Field type mapping

ApsaraDB RDS data type	Realtime Compute data type
TEXT	VARCHAR
BYTE	VARCHAR

ApsaraDB RDS data type	Realtime Compute data type
INTEGER	INT
LONG	BIGINT
DOUBLE	DOUBLE
DATE	VARCHAR
DATETIME	VARCHAR
TIMESTAMP	VARCHAR
TIME	VARCHAR
YEAR	VARCHAR
FLOAT	FLOAT
DECIMAL	DECIMAL
CHAR	VARCHAR

JDBC parameters

Parameter	Description	Default value	Required ApsaraDB RDS for SQL Server version
useUnicode	Specifies whether to use the Unicode character set. If you want to set the characterEncoding parameter to gb2312 or gbk, you must set this parameter to true.	false	1.1g
characterEncoding	The character encoding format. This parameter must be set if the useUnicode parameter is set to true. You can set this parameter to gb2312 or gbk.	false	1.1g
autoReconnect	Specifies whether to automatically reconnect the database when it is disconnected.	false	1.1

Parameter	Description	Default value	Required ApsaraDB RDS for SQL Server version
autoReconnectForPools	Specifies whether to use the reconnection policy for a database connection pool.	false	3.1.3
failOverReadOnly	Specifies whether the database is read-only after it is automatically reconnected.	true	3.0.12
maxReconnects	Specifies the maximum number of reconnection attempts allowed if the autoReconnect parameter is set to true.	3	1.1
initialTimeout	Specifies the interval between two reconnection attempts if the autoReconnect parameter is set to true. Unit: seconds.	2	1.1
connectTimeout	Specifies the timeout period when you use a socket connection to access the database server. Unit: milliseconds. Default value: 0. This value indicates that the connection never times out (only for JDK 1.4 or later).	0	3.0.1
socketTimeout	The timeout period for a socket-based read or write operation. Unit: milliseconds. Default value: 0. This value indicates that the read or write operation never times out.	0	3.0.1

FAQ

- Q: When output data is written to an ApsaraDB RDS result table, is a new data record inserted in the table or is the result table updated based on the primary key value?

A: If a primary key is defined in the DDL statement, the following statement is executed to update the result table: `INSERT INTO tablename(field1,field2, field3, ...) VALUES(value1, value2, value3, ...) ON DUPLICATE KEY UPDATE field1=value1,field2=value2, field3=value3, ...;` . If the primary key value in the output data already exists in the table, the matching record is updated. Otherwise, the output data is inserted as a new record. If no primary key is defined in the DDL statement, the `INSERT INTO` statement is executed to insert the output data.

- Q: How do I perform GROUP BY operations based on the unique index of an ApsaraDB RDS result table?

A: If an ApsaraDB RDS result table has an auto-increment field, this field cannot be declared as the primary key in the DDL statement of Realtime Compute. If you want to perform GROUP BY operations based on the unique index of an ApsaraDB RDS table, declare the unique index in primary key() in the DDL statement.

3.1.4.5.3.5. Create a TSDB result table

This topic describes how to create a Time Series Database (TSDB) result table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause used when you create such a result table.

 **Notice** This topic applies only to Blink 2.0 and later.

Introduction to TSDB

TSDB is a database service that provides efficient read/write, compressed storage, and real-time computing for time series data. It can be widely used in IoT and Internet to monitor devices and business and generate alerts in real time.

DDL syntax

In Realtime Compute for Apache Flink, you can use TSDB to store output data. The following code shows an example:

```
CREATE TABLE stream_test_hitsdb (
  metric VARCHAR,
  `timestamp` INTEGER,
  `value` DOUBLE,
  tagk1 VARCHAR,
  tagk2 VARCHAR,
  tagk3 VARCHAR
) WITH (
  type='hitsdb',
  host='<yourHostName>',
  virtualDomainSwitch = 'false',
  httpConnectionPool = '20',
  batchPutSize = '1000'
);
```

Default format for tables:

- Column 0: metric (VARCHAR)
- Column 1: timestamp (INTEGER), in seconds.

- Column 2: value (DOUBLE)
- Columns 3 to N: tag keys, which are used as the field names in TSDB.

 Note

- You can specify multiple tag columns.
- You must declare metric, timestamp, and value. Their names, sequence, and data types must be the same as those in TSDB.

Parameters in the WITH clause

Parameter	Description	Remarks
type	The type of the result table.	Set the value to <code>hitsdb</code> .
host	The IP address or the domain name mapped to the virtual IP address (VIP) of a TSDB instance.	Enter the hostname of the TSDB instance.
port	The port number used to access the TSDB instance.	Default value: 8242.
virtualDomainSwitch	Specifies whether to use VIPServer.	Default value: false. If you need to use VIPServer, set this parameter to true.
httpConnectionPool	The maximum number of connections in an HTTP connection pool.	Default value: 10.
httpCompress	Specifies whether to use GZIP compression.	Default value: false. This value indicates that GZIP compression is not used.
httpConnectTimeout	The HTTP connection timeout period.	Default value: 0.
ioThreadCount	The number of I/O threads.	Default value: 1.
batchPutBufferSize	The buffer size.	Default value: 10000.

Parameter	Description	Remarks
batchPutRetryCount	The maximum number of retries for writing data to the table.	Default value: 3.
batchPutSize	The maximum number of data records that can be submitted at a time.	Default value: 500.
batchPutTimeLimit	The maximum duration for which a data record can be stored in the buffer.	Default value: 200. Unit: milliseconds.
batchPutConsumerThreadCount	The number of serialized threads.	Default value: 1.

Models used to write data from Realtime Compute for Apache Flink to TSDB

In Blink 3.2 and later, Realtime Compute for Apache Flink can write data to TSDB by using one of the following models:

- Write data points of a single value without tags. This schema contains three fields, whose names cannot be changed. The following example shows the schema format:

```
metric,timestamp,value
```

- Write data points of a single value with tags. The names for the metric, timestamp, and value fields cannot be changed but the tag name can be specified as required. The following example shows the schema format:

```
metric,timestamp,value,tagKey1,.....,tagKeyN
```

- Write data points of a single value with an unknown number of tags. The schema contains four fields, whose names cannot be changed. The following example shows the schema format:

```
metric,timestamp,value,tags
```

The content of tags is a JSON string shown in the following example. Therefore, the limit that the Blink table schema requires a fixed number of tags does not apply.

```
{"tagKey1":"tagValue1","tagKey2":"tagValue2",.....,"tagKeyN":"tagValueN"}
```

- Write data points of multiple values without tags. The following example shows the schema format:

```
metric,timestamp,field_name1,field_name2,.....,field_nameN
```

The names of the metric and timestamp fields cannot be changed. For multi-value fields, a fixed prefix "field_" must be added to each field to identify the field so that fields can be distinguished from tags and single-value data writing is supported. For example, if the field named `field_name1` is written into TSDB, its prefix `field_` is automatically removed and only `name1` is retained. The following example shows the actual format in this model. `name1` and `name2` are the names of multi-value fields.

```
metric,timestamp,name1,name2,……,nameN
```

- Write data points of multiple values with tags. The names of the metric and timestamp fields cannot be changed but the tag name can be specified as required. The following example shows the schema format:

```
metric,timestamp,tagKey1,……,tagKeyN,field_name1,field_name2,……,field_nameN
```

- Write data points of a single value with an unknown number of tags. The following example shows the schema format:

```
metric,timestamp,tags,field_name1,field_name2,……,field_nameN
```

The content of tags is a JSON string shown in the following example. Therefore, the limit that the Blink table schema requires a fixed number of tags does not apply.

```
{"tagKey1":"tagValue1","tagKey2":"tagValue2",……,"tagKeyN":"tagValueN"}
```

FAQ

Q: Why does the following error occur during a failover: "The LONG type cannot be converted to the INT type"?

A: Blink versions earlier than 2.2.5 support only the INT data type. Blink 2.2.5 and later versions support the BIGINT data type.

3.1.4.5.3.6. Create an KVStore for Redis result table

This topic describes how to create an KVStore for Redis result table in Realtime Compute for Apache Flink. This topic also describes the parameters in the WITH clause, the mappings between the field data types of KVStore for Redis and Realtime Compute for Apache Flink, and the attribute fields used when you create an KVStore for Redis result table.

Notice

- This topic applies only to Blink V3.2.0 and later.
- Realtime Compute for Apache Flink allows you to use self-managed Redis databases to store output data in result tables.

Introduction to KVStore for Redis

KVStore for Redis is a database service that is compatible with the protocols of the open source Redis system. It supports a hybrid of memory and hard disks for storage. KVStore for Redis provides a hot standby architecture to ensure high availability. Based on the scalable cluster architecture, KVStore for Redis can meet the business requirements for high throughputs, low-latency operations, and flexible configuration changes. Realtime Compute for Apache Flink allows you to store the output streaming data in KVStore for Redis.

Syntax

You can use five data types when you write data to KVStore for Redis result tables. To create an KVStore for Redis result table, execute the following data definition language (DDL) statements:

- **STRING type**

A DDL statement has two columns. The first column lists keys and the second column lists values. To insert data into an KVStore for Redis result table, run the `set key value` command.

```
create table resik_output (
  a varchar,
  b varchar,
  primary key(a)
) with (
  type = 'redis',
  mode = 'string',
  host = '${redisHost}', -- An example value is '127.0.0.1'.
  port = '${redisPort}', -- An example value is '6379'.
  dbName = '${dbName}', -- The default value is 0.
  ignoreDelete = 'true' -- Specifies whether to delete the previously inserted data when the retraction message is returned. The default value is false.
);
```

- **LIST type**

A DDL statement has two columns. The first column lists keys and the second column lists values. To insert data into an KVStore for Redis result table, run the `lpush key value` command.

```
create table resik_output (
  a varchar,
  b varchar,
  primary key(a)
) with (
  type = 'redis',
  mode = 'list',
  host = '${redisHost}', -- An example value is '127.0.0.1'.
  port = '${redisPort}', -- An example value is '6379'.
  dbName = '${dbName}', -- The default value is 0.
  ignoreDelete = 'true' -- Specifies whether to delete the previously inserted data when the retraction message is returned. The default value is false.
);
```

- **SET type**

A DDL statement has two columns. The first column lists keys and the second column lists values. To insert data into an KVStore for Redis result table, run the `sadd key value` command.


```
create table resik_output (  
  a varchar,  
  b varchar,  
  primary key(a)  
) with (  
  type = 'redis',  
  mode = 'set',  
  host = '${redisHost}', -- An example value is '127.0.0.1'.  
  port = '${redisPort}', -- An example value is '6379'.  
  dbNum = '${dbNum}', -- The default value is 0.  
  ignoreDelete = 'true' -- Specifies whether to delete the previously inserted data when the retraction message is returned. The default value is false.  
);
```

- HASHMAP type

A DDL statement has three columns. The first column lists keys, the second column lists hash keys, and the third column lists hash values. To insert data into an KVStore for Redis result table, run the `hmset key hash_key hash_value` command.

```
create table resik_output (  
  a varchar,  
  b varchar,  
  c varchar,  
  primary key(a)  
) with (  
  type = 'redis',  
  mode = 'hashmap',  
  host = '${redisHost}', -- An example value is '127.0.0.1'.  
  port = '${redisPort}', -- An example value is '6379'.  
  dbNum = '${dbNum}', -- The default value is 0.  
  ignoreDelete = 'true' -- Specifies whether to delete the previously inserted data when the retraction message is returned. The default value is false.  
);
```

- SORTEDSET type

A DDL statement has three columns. The first column lists keys, the second column lists scores, and the third column lists values. To insert data into an KVStore for Redis result table, run the `add key score value` command.

```
create table resik_output (  
  a varchar,  
  b double, -- The data must be of the DOUBLE type.  
  c varchar,  
  primary key(a)  
) with (  
  type = 'redis',  
  mode = 'sortedset',  
  host = '${redisHost}', -- An example value is '127.0.0.1'.  
  port = '${redisPort}', -- An example value is '6379'.  
  dbName = '${dbName}', -- The default value is 0.  
  ignoreDelete = 'true' -- Specifies whether to delete the previously inserted data when the retraction message is returned. The default value is false.  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Valid value
type	The type of the result table.	Yes	Set the value to <code>redis</code> .
mode	The data type of the KVStore for Redis result table.		Valid values: • string • list • set • hashmap • sortedset
host	The endpoint of the KVStore for Redis database.		Example: <code>127.0.0.1</code> .
port	The port of the KVStore for Redis database.	No	Default value: 6379.
dbName	The sequence number of the KVStore for Redis database.		Default value: 0.
ignoreDelete	Specifies whether to ignore the retraction message.		Valid values: true and false. Default value: false. If this parameter is set to false, the inserted data and the keys of the data are deleted when a retraction message is received.

Parameter	Description	Required	Valid value
password	The password that is used to access the KVStore for Redis database.		This parameter is empty by default. This indicates that permission verification is not required.
clusterMode	Specifies whether the KVStore for Redis database is in cluster mode.		Valid values: • true: cluster mode. • false: standalone mode. This is the default value.  Note Only Blink 3.6.X and later support this parameter.

Field type mapping

The following table lists the data type mappings between KVStore for Redis and Realtime Compute for Apache Flink. We recommend that you declare the mappings in DDL statements.

Data type of KVStore for Redis	Data type of Realtime Compute for Apache Flink
STRING	VARCHAR
SCORE	DOUBLE

 **Note** The data of the SCORE type is added to the values of the SORTEDSET data type in KVStore for Redis databases. You must manually set a score of the DOUBLE type for each sorted set value and sort the values based on their scores in ascending order.

Sample code

The following sample code shows how to create an KVStore for Redis result table in a Realtime Compute for Apache Flink job.

```
CREATE TABLE random_stream (  
  v VARCHAR,  
  p VARCHAR) with (  
  type = 'random'  
);  
create table resik_output (  
  a VARCHAR,  
  b VARCHAR,  
  primary key(a)  
) with (  
  type = 'redis',  
  mode = 'string',  
  host = '<yourRedisHost>',  
  password = '<yourRedisPassword>'  
);  
INSERT INTO resik_output  
SELECT v, p  
FROM random_stream;
```

3.1.4.5.3.7. Create an AnalyticDB for MySQL V2.0 result table

This topic describes how to create an AnalyticDB for MySQL V2.0 result table. It also describes the mapping between the field data types of AnalyticDB for MySQL and Realtime Compute for Apache Flink.

Notice

- This topic applies only to Blink 1.4.5 and later.
- You are allowed to define an auto-increment primary key for an AnalyticDB for MySQL V2.0 database. If you want to use the auto-increment primary key, do not declare the auto-increment field in a data definition language (DDL) statement. For example, if you use ID as an auto-increment field, do not declare the ID field in the DDL statement. When a row of output data is written to the ApsaraDB RDS for MySQL database, the value for the auto-increment field is automatically filled.

Introduction to AnalyticDB for MySQL

AnalyticDB for MySQL is a real-time online analytical processing (OLAP) service that is developed by Alibaba Cloud. It is a high concurrency service that has excellent performance in processing large amounts of data. AnalyticDB for MySQL allows you to query and analyze hundreds of billions of data records within milliseconds.

DDL syntax

 **Note** For more information about how to create an AnalyticDB for MySQL V3.0 result table, see [Create an AnalyticDB for MySQL V3.0 result table](#).

In Realtime Compute for Apache Flink, you can use AnalyticDB for MySQL V2.0 to store output data. The following code shows an example:

```
CREATE TABLE stream_test_hotline_agent (  
  id INTEGER,  
  len BIGINT,  
  content VARCHAR,  
  PRIMARY KEY(id)  
) WITH (  
  type='ads',  
  url='yourDatabaseURL',  
  tableName='<yourDatabaseTableName>',  
  userName='<yourDatabaseUserName>',  
  password='<yourDatabasePassword>',  
  batchSize='20'  
);
```

 **Note** The primary key that is declared in an AnalyticDB for MySQL result table must be consistent with that in an AnalyticDB for MySQL database. The primary key is case-sensitive. Inconsistency may cause the "array index out of bounds" error.

Parameters in the WITH clause

Parameter	Description	Remarks
type	The type of the result table.	Set the value to ads.
url	The Java Database Connectivity (JDBC) URL of a database.	Contact Alibaba Cloud O&M engineers.
tableName	The name of the table in the database	None.
username	The username that is used to log on to the database.	None.
password	The password that is used to log on to the database	None.
maxRetryTimes	The maximum number of retries for writing data to the table.	Optional. Default value: 10.

Parameter	Description	Remarks
bufferSize	The maximum number of data records that can be stored in the buffer before data deduplication is triggered.	Optional. Default value: 5000. This value indicates that data deduplication is triggered if the number of input data records in the buffer reaches 5,000.
batchSize	The number of data records that are written at a time.	Optional. Default value: 1000. Note If error code 20015 is returned, the batchSize parameter is set to a large value. For AnalyticDB for MySQL databases, the size of data records that are written at a time cannot exceed 1 MB. If the batchSize parameter is set to 1000, the average size of each data record cannot exceed 1 KB.
batchWriteTimeoutMs	The timeout period for writing data.	Optional. Default value: 5000. Unit: milliseconds. This value indicates that if the number of input data records does not reach the value specified by the batchSize parameter within 5,000 milliseconds, all cached data is written to the result table.
connectionMaxActive	The maximum number of connections in a connection pool.	Optional. Default value: 30.
ignoreDelete	Specifies whether to skip the delete operation.	Default value: false.

Field type mapping

We recommend that you declare the mapping between the field data types of AnalyticDB for MySQL and Realtime Compute for Apache Flink in DDL statements.

Data type of AnalyticDB for MySQL	Data type of Realtime Compute for Apache Flink
BOOLEAN	BOOLEAN
TINYINT	INT
SAMLLINT	
INT	
BIGINT	BIGINT
DOUBEL	DOUBLE

Data type of AnalyticDB for MySQL	Data type of Realtime Compute for Apache Flink
VARCHAR	VARCHAR
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP

3.1.4.5.3.8. Create an AnalyticDB for MySQL V3.0 result table

This topic describes how to create an AnalyticDB for MySQL V3.0 result table in Realtime Compute for Apache Flink. This topic also describes the parameters in the WITH clause used when you create an AnalyticDB for MySQL V3.0 result table.



Notice

- You cannot register AnalyticDB for MySQL V3.0 storage resources in the Realtime Compute for Apache Flink console to store result tables.
- This topic applies only to Blink 3.3.0 and later.
- For more information about how to create an AnalyticDB for MySQL V2.0 result table, see [Create an AnalyticDB for MySQL V2.0 result table](#).
- You are allowed to define an auto-increment primary key for an AnalyticDB for MySQL V3.0 database. If you want to use the auto-increment primary key, do not declare the auto-increment field in a data definition language (DDL) statement. For example, if you use ID as an auto-increment field, do not declare the ID field in the DDL statement. When a row of output data is written to the AnalyticDB for MySQL V3.0 database, the value for the auto-increment field is automatically filled.

DDL syntax

In Realtime Compute for Apache Flink, you can use AnalyticDB for MySQL V3.0 to store output data. The following code shows an example:

```
CREATE TABLE rds_output (
  id INT,
  len INT,
  content VARCHAR,
  PRIMARY KEY(id,len)
) WITH (
  type='ADB30',
  url='jdbc:mysql://<yourNetworkAddress>:<PortId>/<yourDatabaseName>',
  tableName='<yourDatabaseTableName>',
  userName='<yourDatabaseUserName>',
  password='<yourDatabasePassword>'
);
```

Implementation

Realtime Compute for Apache Flink writes data to an AnalyticDB for MySQL V3.0 result table in two steps:

1. Converts each row of output data to a line of SQL statement.
2. Writes and executes the SQL statement in the destination database.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the connector.	Yes	Set the value to ADB30 .
url	The Java Database Connectivity (JDBC) URL of the database.	Yes	The URL of the AnalyticDB for MySQL database.  Note databaseName is the name of the AnalyticDB for MySQL database.
tableName	The name of the table.	Yes	None.
username	The username that is used to access the database.	Yes	None.
password	The password that is used to access the database.	Yes	None.
maxRetryTimes	The maximum number of retries for writing data to the table.	No	Default value: 3.

Parameter	Description	Required	Remarks
bufferSize	The maximum number of data records that can be stored in the buffer before data deduplication is triggered.	No	Default value: 1000. This value indicates that duplicates are removed when the number of input data records reaches 1,000.  Note This parameter is valid only after you specify the primary key.
batchSize	The number of data records that can be written at a time.	No	Default value: 1000.  Note This parameter is valid only after you specify the primary key.
flushIntervalMs	The time interval at which the cache is cleared.	No	Default value: 3000. Unit: milliseconds. This value indicates that all the cached data is written to the result table if the number of input data records does not reach the batchSize value within 3,000 milliseconds.
ignoreDelete	Specifies whether to ignore delete operations.	No	Default value: false. This value indicates that the delete operations are supported.
replaceMode	Specifies whether to use the REPLACE INTO statement to insert data into the table.	No	Valid values: - true: The REPLACE INTO statement is used to insert data into the table. This value is the default value. - false: The INSERT INTO ON DUPLICATE KEY statement is used to insert data into the table.  Note This parameter is valid only when the following conditions are met: - The Blink version is 3.X or later. - The AnalyticDB for MySQL version must be 3.1.3.5 or later.
reserveMillisecond	Specifies whether to reserve the millisecond component in a value of the TIMESTAMP data type.	No	Default value: false. This value indicates that the millisecond component is not reserved.

3.1.4.5.3.9. Create a Log Service result table

This topic describes how to create a Log Service result table in Realtime Compute for Apache Flink.

Notice

- This topic applies only to Blink 1.4.5 and later.
- Log Service result tables support only fields of the VARCHAR type.

Introduction to Log Service

Log Service is an end-to-end service for log data. It helps you quickly collect, consume, ship, query, and analyze data to improve O&M and operations efficiency and build up the capabilities to process a large number of logs. Log Service stores streaming data. Therefore, Realtime Compute for Apache Flink can use Log Service tables as result tables for the processing of streaming data.

DDL syntax

In Realtime Compute for Apache Flink, you can use Log Service to store output data. The following code shows an example:

```
create table sls_stream(  
  `name` VARCHAR,  
  age BIGINT,  
  birthday BIGINT  
)with(  
  type='sls',  
  endPoint='http://cn-hangzhou-corp.sls.aliyuncs.com',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessKey>',  
  project='<yourProjectName>',  
  logstore='<yourLogstoreName>'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
endPoint	The endpoint of Log Service.	Yes	Contact Alibaba Cloud O&M engineers.
project	The name of a project.	Yes	None.
logstore	The name of the table in the database.	Yes	None.
accessId	AccessKey ID	Yes	None.

Parameter	Description	Required	Remarks
accessKey	AccessKey Secret	Yes	None.
topic	An attribute field.	No	This parameter is empty by default. You can use the selected field as the topic attribute field.
timestampColumn	An attribute field.	No	This parameter is empty by default. You can use the selected field as the timestamp attribute field. The data type of this parameter must be INT. If no field is selected, the current time is used as the attribute field.  Note This parameter is available in Realtime Compute for Apache Flink V2.2.2 and later.
source	An attribute field. The source of a log entry. For example, the value of this parameter can be the IP address of the server where the log entry is generated.	No	This parameter is empty by default. You can use the selected field as the source attribute field.
partitionColumn	The partition key column.	No	This parameter is required if the mode parameter is set to partition .
flushIntervalMs	The interval at which data writing is triggered.	No	Default value: 2000. Unit: milliseconds.
reserveMillisecond	Specifies whether to reserve the millisecond component in a value of the TIMESTAMP data type.	No	Default value: false. This value indicates that the millisecond component is not reserved.  Note This parameter is available in Realtime Compute for Apache Flink V2.2.6 and later.

Field type mapping

The following table describes the mapping between the data types of Log Service and Realtime Compute for Apache Flink. We recommend that you declare the mapping in DDL statements.

Data type of Log Service	Data type of Realtime Compute for Apache Flink
STRING	VARCHAR

Sample code

The following sample code demonstrates how to create a Log Service result table in a Realtime Compute for Apache Flink job:

```
CREATE TABLE random_input (  
  a VARCHAR,  
  b VARCHAR) with (  
  type = 'random'  
);  
create table sls_output(  
  a varchar,  
  b varchar  
)with(  
  type='sls',  
  endPoint='http://cn-hangzhou-corp.sls.aliyuncs.com',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessKey>',  
  project='ali-cloud-streamtest',  
  logStore='stream-test2'  
);  
INSERT INTO sls_output  
SELECT a, b  
FROM random_input;
```

FAQ

Q: How do I specify the topic field in a Log Service result table?

A: You can specify the topic field as a field in the result table. For example, specify `topic='age'` in the sample code. After the configuration is completed, the value of the `age` field is written into Log Service but Log Service does not write the `age` field into the downstream storage systems.

3.1.4.5.3.10. Create a Tablestore result table

This topic describes how to create a Tablestore result table in Realtime Compute for Apache Flink. It also describes the mappings between the field data types of Tablestore and Realtime Compute for Apache Flink.

 **Notice** This topic applies only to Blink 1.4.5 and later.

Introduction to Tablestore

Tablestore is a distributed NoSQL database service built on the Apsara distributed operating system of Alibaba Cloud. Tablestore adopts data sharding and load balancing technologies to scale out and handle concurrent transactions. You can use Tablestore to store and query a large amount of structured data in real time.

DDL syntax

In Realtime Compute for Apache Flink, you can use Tablestore to store output data. The following code shows an example:

```
CREATE TABLE stream_test_hotline_agent (  
  name VARCHAR,  
  age BIGINT,  
  birthday BIGINT,  
  PRIMARY KEY (name,age)  
) WITH (  
  type='ots',  
  instanceName='<yourInstanceName>',  
  tableName='<yourTableName>',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessSecret>',  
  endPoint='<yourEndpoint>',  
  valueColumns='birthday'  
);
```

 Note

- The value of the valueColumns parameter cannot be a declared primary key.
- The declared Tablestore result table must contain at least one attribute column and the primary key column.

Parameters in the WITH clause

Parameter	Description	Remarks
type	The type of the result table.	Set the value to ots.
instanceName	The name of a Tablestore instance.	None.
tableName	The name of the table in the database	None.
endPoint	The endpoint of the instance.	Contact Alibaba Cloud O&M engineers.
accessId	AccessKey ID	None.
accessKey	AccessKey Secret	None.
valueColumns	The name of a column to be inserted.	Separate multiple column names with commas (,), for example, 'ID,NAME' .

Parameter	Description	Remarks
bufferSize	The maximum number of data records that can be stored in the buffer before deduplication is triggered.	Optional. Default value: 5000. This value indicates that deduplication is triggered if the number of input data records in the buffer reaches 5,000.  Note Realtime Compute for Apache Flink removes data record duplicates based on the primary key of the Tablestore result table. You can set bufferSize to the number of data record duplicates to be removed. Then, Realtime Compute for Apache Flink writes the data records after duplicates are removed. You can set batchSize to the number of data records to be written at a time.
batchWriteTimeoutMs	The write timeout period.	Optional. Default value: 5000. Unit: milliseconds. This value indicates that if the number of input data records does not reach the value specified by the batchSize parameter within 5,000 milliseconds, all cached data is written into the result table.
batchSize	The number of data records that can be written at a time.	Optional. Default value: 100.
retryIntervalMs	The retry interval.	Optional. Default value: 1000. Unit: milliseconds.
maxRetryTimes	The maximum number of retries for writing data to a table.	Optional. Default value: 100.
ignoreDelete	Specifies whether to ignore DELETE operations.	Default value: false.

Field type mapping

Data type of Tablestore	Data type of Realtime Compute for Apache Flink
INTEGER	BIGINT
STRING	VARCHAR
BOOLEAN	BOOLEAN
DOUBLE	DOUBLE

 **Note** You must define a `primary key` in a Tablestore result table. Output data is appended to the Tablestore result table to update the result.

3.1.4.5.3.11. Create a MaxCompute result table

This topic describes how to create a MaxCompute result table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause, data type mapping, and FAQ involved when you create a MaxCompute result table.

Notice

- Only Blink 1.5.1 and later versions support MaxCompute result tables.
- A clustered table of MaxCompute cannot be used as a result table.

Implementation

The MaxCompute sink works in two phases:

1. Writes data. The MaxCompute sink calls an API operation in the MaxCompute SDK to write data into the buffer. Then, the sink uploads data to the temporary files of MaxCompute at a specified interval or when the data size in the buffer exceeds 64 MB.
2. Commits sessions. When a task runs a checkpoint, the MaxCompute sink calls the COMMIT method of Tunnel to commit sessions, moves temporary files to the data directory of the MaxCompute table, and modifies the metadata.

 **Note** The COMMIT method does not provide atomicity. Therefore, the MaxCompute sink supports at-least-once delivery instead of exactly-once delivery.

Syntax

In Realtime Compute for Apache Flink, you can use MaxCompute to store output data. The following code shows an example:

 **Note** The names, sequence, and types of the fields defined in the data definition language (DDL) statement must be the same as those in the MaxCompute physical table. Otherwise, the queried data in the MaxCompute physical table may be `/n`.

```
create table odps_output(  
  id INT,  
  user_name VARCHAR,  
  content VARCHAR  
) with (  
  type = 'odps',  
  endPoint = '<YourEndPoint>',  
  project = '<YourProjectName>',  
  tableName = '<YourtableName>',  
  accessId = '<yourAccessKeyId>',  
  accessKey = '<yourAccessKeySecret>',  
  `partition` = 'ds=2018****'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the result table.	Yes	Set the value to <code>odps</code> .
endPoint	The endpoint of MaxCompute.	Yes	Contact Alibaba Cloud O&M engineers.
tunnelEndPoint	The endpoint of the MaxCompute Tunnel service.	Yes	Contact Alibaba Cloud O&M engineers.
project	The name of a MaxCompute project.	Yes	None.
tableName	The name of a table.	Yes	None.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.
partition	The name of a partition.	No	This parameter is required if a partitioned table exists. - Fixed partition For example, <code>`partition` = 'ds=20180905'</code> indicates that data is written into partition <code>ds=20180905</code>. - Dynamic partition (available in Blink 3.2.1 and later) If the partition values are not displayed in plaintext mode, data is written into different partitions based on the values of the partition key columns specified in the data. For example, <code>`partition`='ds'</code> indicates that data is written into partitions based on the value of the <code>ds</code> field. If you want to create multi-level dynamic partitions, make sure that the sequence of the partition fields in the WITH clause and DDL statement of the MaxCompute result table are consistent with the field sequence of the MaxCompute physical table. Multiple partition fields are separated by commas (,).

Parameter	Description	Required	Remarks
			Note ◦ In the CREATE TABLE statement, you must explicitly specify the partition key column that you use to create dynamic partitions. ◦ If the partition field for dynamic partitions is left empty and the value of the ds field is <code>null</code> or <code>"</code>, the output varies based on the Blink version: ■ For Blink 3.2.1 and earlier, a NullPointerException (NPE) error is returned. ■ For Blink 3.2.2 and later, a partition with ds=null is created.

Parameter	Description	Required	Remarks
flushIntervalMs	The flush interval for the buffer of a writer in MaxCompute Tunnel. Unit: milliseconds. The MaxCompute sink first inserts data into its buffer. Then, the MaxCompute sink writes the data in the buffer to the destination MaxCompute table at an interval that is specified by the flushIntervalMs parameter. The sink also writes the data to the destination table when the size of the buffer data exceeds the specified threshold.	No	Default value: 30000. Unit: milliseconds.  Note This parameter is available in Blink 3.6.0 and later.
partitionBy	The columns based on which hash shuffling is implemented. Before data is written into a sink node, the system implements hash shuffling based on the value of this parameter. This way, data flows to the destination sink node.	No	The system implements hash shuffling based on multiple columns. The column names are separated by commas (,). This parameter is empty by default.  Note This parameter is available in Blink 3.6.0 and later.

Parameter	Description	Required	Remarks
isOverwrite	Specifies whether to clear the result table before data is written to a sink node.	No	- For versions earlier than Blink 3.2, the default value is true. - For Blink 3.2 and later, the default value is false. When you declare a MaxCompute result table for a streaming job, you must set the isOverwrite parameter to false. Otherwise, an error is returned during compilation.  Note For Blink 3.2 and later, you can change the value of the isOverwrite parameter to true. For Blink versions earlier than 3.2, if you want to change the value of the isOverwrite parameter, you must upgrade the Blink version first.
dynamicPartitionLimit	The number of partitions in the underlying framework.	No	The default value is 100. A map in the memory maintains the mappings between the existing partitions to which data is written and the Tunnel service or the writer. If the map size exceeds the value of the dynamicPartitionLimit parameter, the system reports the following error: Too many dynamic partitions: 100, which exceeds the size limit: 100.

Field type mapping

MaxCompute data type	Data type of Realtime Compute for Apache Flink
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATE TIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
STRING	VARCHAR

MaxCompute data type	Data type of Realtime Compute for Apache Flink
DECIMAL	DECIMAL
BINARY	VARBINARY

Sample code

The following sample code demonstrates how to create a MaxCompute result table in a Realtime Compute for Apache Flink job.

- Write data into a fixed partition

```
CREATE TABLE source (  
  id INT,  
  len INT,  
  content VARCHAR  
) with (  
  type = 'random'  
);  
create table odps_sink (  
  id INT,  
  len INT,  
  content VARCHAR  
) with (  
  type = 'odps',  
  endPoint = '<yourEndpoint>',  
  project = '<yourProjectName>',  
  tableName = '<yourTableName>',  
  accessId = '<yourAccessId>',  
  accessKey = '<yourAccessPassword>',  
  `partition` = 'ds=20180418'  
);  
INSERT INTO odps_sink  
SELECT  
  id, len, content  
FROM source;
```

- Write data into a dynamic partition

```
CREATE TABLE source (  
  id INT,  
  len INT,  
  content VARCHAR,  
  c TIMESTAMP  
) with (  
  type = 'random'  
);  
create table odps_sink (  
  id INT,  
  len INT,  
  content VARCHAR,  
  ds VARCHAR --The partition key column that you use to create dynamic partitions must be explicitly specified in the CREATE TABLE statement.  
) with (  
  type = 'odps',  
  endPoint = '<yourEndpoint>',  
  project = '<yourProjectName>',  
  tableName = '<yourTableName>',  
  accessId = '<yourAccessId>',  
  accessKey = '<yourAccessPassword>',  
  `partition` = 'ds' --The partition value is not provided. This means that data is written into different partitions based on the value of the ds field.  
);  
INSERT INTO odps_sink  
SELECT  
  id,  
  len,  
  content,  
  DATE_FORMAT(c, 'yyMMdd') as ds  
FROM source;
```

FAQ

- What do the `endPoint` and `tunnelEndPoint` parameters specify in the Alibaba Cloud public cloud? What happens if the parameter configurations are invalid?

For more information about the `endPoint` and `tunnelEndPoint` parameters, see [Configure endpoints](#). If the configuration of the two parameters is invalid in a VPC, a task exception may occur.

- If the configuration of the `endPoint` parameter is invalid, the task stops at the progress of 91%.
 - If the configuration of the `tunnelEndPoint` parameter is invalid, the task fails.
- Q: Does Realtime Compute for Apache Flink clear a MaxCompute result table before it writes data into the result table in stream mode when `isOverwrite` is set to `true`?

A: Blink versions earlier than 3.2 support this feature. Blink 3.2 and later versions do not support this feature.

If the `isOverwrite` parameter is set to `true`, Realtime Compute for Apache Flink clears a MaxCompute result table before it writes data into the result table. Realtime Compute for Apache Flink clears data from the existing result table or the result partition each time a job is started or resumed, or before Realtime Compute for Apache Flink writes data into the result table.

- For Blink versions earlier than 3.2, the default value of `isOverwrite` is `true` and cannot be changed. Data may be lost after a streaming job is suspended or resumed.
- For Blink 3.2 and later, the default value of `isOverwrite` is `false`. If you declare a MaxCompute result table for a streaming job, you must set the `isOverwrite` parameter to `false`. Otherwise, an error is returned during compilation. MaxCompute result tables in stream mode support the at-least-once data security mechanism. If a job fails, duplicate data may be generated.

3.1.4.5.3.12. Create an Elasticsearch result table

This topic describes how to create an Elasticsearch result table in Realtime Compute for Apache Flink. It also describes the parameters in the `WITH` clause used when you create an Elasticsearch result table.

 **Notice** This topic applies only to Blink 3.2.2 and later.

DDL syntax

In Realtime Compute for Apache Flink, you can use Elasticsearch to store output data. The following code shows an example:

```
CREATE TABLE es_stream_sink(  
  field1 LONG,  
  field2 VARBINARY,  
  field3 VARCHAR,  
  PRIMARY KEY(field1)  
)WITH(  
  type='elasticsearch',  
  endPoint='http://es-cn-mp****.public.elasticsearch.aliyuncs.com:****',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessSecret>',  
  index='<yourIndex>',  
  typeName='<yourTypeName>'  
);
```

Note

- Elasticsearch supports data updates based on the PRIMARY KEY field. You can specify only one field for the PRIMARY KEY field.
- After you specify the PRIMARY KEY field, values in the field are used as document IDs.
- If you do not specify the PRIMARY KEY field, document IDs are randomly generated. For more information, see [Index API](#).
- In full update mode, later documents overwrite earlier documents.
- In incremental update mode, the system updates the fields based on the entered field values.
- All updates use the UPSERT syntax by default, which means to insert or update data.

Parameters in the WITH clause (general configurations)

Parameter	Description	Default value	Required
type	The type of the connector.	elasticsearch	Yes
endPoint	The server address of an Elasticsearch cluster, for example, <i>http://127.0.0.1:9211</i> .	None	Yes
accessId	The AccessKey ID that is used to access an Elasticsearch cluster.  Note If you use the Kibana plug-in to connect to Elasticsearch, enter the Kibana logon ID.	None	Yes
accessKey	The AccessKey secret that is used to access an Elasticsearch cluster.  Note If you use the Kibana plug-in to connect to Elasticsearch, enter the Kibana logon password.	None	Yes
index	The index name, which is similar to the database name.	None	Yes
typeName	The type name, which is similar to the database table name.	None	Yes
bufferSize	The maximum number of data records that can be stored in the buffer before deduplication is triggered.	1000	No
maxRetryTimes	The maximum number of retries for writing data into a table.	30	No
timeout	The read timeout period. Unit: milliseconds.	600000	No
discovery	Specifies whether node discovery is enabled. If this feature is enabled, the client refreshes the server list every five minutes.	false	No
compression	Specifies whether to compress request bodies in the GZIP format.	true	No
multiThread	Specifies whether to enable multithreading for JestClient.	true	No

Parameter	Description	Default value	Required
ignoreWriteError	Specifies whether to ignore write exceptions.	false	No
settings	The settings used to create indexes.	None	No
updateMode	The update mode used after the primary key is specified. - full: Full data is overwritten. - inc: Incremental data is updated.	full	No

Parameters in the WITH clause (dynamic indexing)

Parameter	Description	Default value	Required
dynamicIndex	Specifies whether to enable dynamic indexing. - true: Dynamic indexing is enabled. - false: Dynamic indexing is disabled.	false	No.
indexField	The field name of the index.	None	This parameter is required when dynamicIndex is set to true. This parameter supports only the TIMESTAMP (in seconds), DATE, and LONG data types.
indexInterval	The interval at which an index is changed.	d	This parameter is required when dynamicIndex is set to true. Valid values: - d: day - m: month - w: week

Parameter	Description	Default value	Required
mapping	The mapping between the types and formats of fields in a document configured when dynamic indexing is enabled.	Empty string	No. Blink 3.7.0 and later versions support this parameter. The following example shows how to set the mapping between the type and format of the sendTime field: <pre> { "properties": { "sendTime": { "type": "date", "format": "yyyy-MM-dd HH:mm:ss" } } }</pre>

Note

- Only Blink 2.2.7 and later versions support the dynamic indexing feature.
- After dynamic indexing is enabled, the `index` name in the basic configuration is used as the unified alias for indexes created subsequently. An alias can correspond to multiple indexes.
- Actual index names that correspond to different values of `indexInterval` :
 - d -> Alias "yyyyMMdd"
 - m -> Alias "yyyyMM"
 - w -> Alias "yyyyMMW"
- You can use Index API to change a single actual index, but the alias supports only the `GET` method. If you want to change the alias, see [Index Aliases](#).

3.1.4.5.3.13. Create a Message Queue for Apache Kafka result table

This topic describes how to create a Message Queue for Apache Kafka result table in Realtime Compute for Apache Flink. It also describes the mapping between the values of the type parameter and Kafka versions.

Notice

- This topic applies only to Realtime Compute for Apache Flink V2.0 and later.
- This topic applies only to Realtime Compute for Apache Flink in exclusive mode.
- Data of a Message Queue for Apache Kafka result table can be written into a self-managed Kafka cluster. Before data is written, you must pay attention to the mapping between the values of the type parameter and Kafka versions, and the network configurations of the self-managed Kafka cluster and the Realtime Compute for Apache Flink cluster.

Introduction to the Message Queue for Apache Kafka result table

Message Queue for Apache Kafka is a distributed, high-throughput, and scalable message queue service provided by Alibaba Cloud. This service is widely used in big data fields, such as log collection, monitoring data aggregation, streaming data processing, and online and offline data analysis. Realtime Compute for Apache Flink allows you to create source tables and result tables of Message Queue for Apache Kafka for the processing of streaming data.

DDL syntax

The following example demonstrates how to create a Message Queue for Apache Kafka result table in a data definition language (DDL) statement.

```
create table sink_kafka (  
  messageKey VARBINARY,  
  `message` VARBINARY,  
  PRIMARY KEY (messageKey)  
) with (  
  type = 'kafka010',  
  topic = '<yourTopicName>',  
  bootstrap.servers = '<yourServerAddress>'  
);
```

Note

- When you create a Message Queue for Apache Kafka result table, you must specify **PRIMARY KEY (messageKey)** in plaintext mode.
- Only Blink 2.2.6 and later versions support the display of metrics such as transactions per second (TPS) and requests per second (RPS) of Alibaba Cloud Message Queue for Apache Kafka or a self-managed Kafka database.

Parameters in the WITH clause

- General configurations

Parameter	Description	Required	Remarks
type	The Kafka version.	Yes	Valid values: Kafka08, Kafka09, Kafka010, and Kafka011. For more information, see Mapping between the values of the type parameter and Kafka versions .
topic	The name of the topic.	Yes	None.

- Required configurations

- Required configurations for Kafka08

Parameter	Description
zookeeper.connect	The ZooKeeper URL.

- Required configurations for Kafka09, Kafka010, and Kafka011

Parameter	Description
bootstrap.servers	The Kafka cluster address.

- Optional configurations

- consumer.id
- socket.timeout.ms
- fetch.message.max.bytes
- num.consumer.fetchers
- auto.commit.enable
- auto.commit.interval.ms
- queued.max.message.chunks
- rebalance.max.retries
- fetch.min.bytes
- fetch.wait.max.ms
- rebalance.backoff.ms
- refresh.leader.backoff.ms
- auto.offset.reset
- consumer.timeout.ms
- exclude.internal.topics
- partition.assignment.strategy
- client.id
- zookeeper.session.timeout.ms
- zookeeper.connection.timeout.ms
- zookeeper.sync.time.ms
- offsets.storage
- offsets.channel.backoff.ms
- offsets.channel.socket.timeout.ms
- offsets.commit.max.retries
- dual.commit.enabled
- partition.assignment.strategy
- socket.receive.buffer.bytes
- fetch.min.bytes

 **Note** For more information about optional configuration items, see the following official Kafka documentation:

- [Kafka09](#)
- [Kafka010](#)
- [Kafka011](#)

Mapping between the values of the type parameter and Kafka versions

type	Kafka version
Kafka08	0.8.22
Kafka09	0.9.0.1
Kafka010	0.10.2.1
Kafka011	0.11.0.2 and later

Example

```
create table datahub_input (  
  id VARCHAR,  
  nm VARCHAR  
) with (  
  type = 'datahub'  
);  
create table sink_kafka (  
  messageKey VARBINARY,  
  `message` VARBINARY,  
  PRIMARY KEY (messageKey)  
) with (  
  type = 'kafka010',  
  topic = '<yourTopicName>',  
  bootstrap.servers = '<yourServerAddress>'  
);  
INSERT INTO  
  sink_kafka  
SELECT  
  cast(id as VARBINARY) as messageKey,  
  cast(nm as VARBINARY) as `message`  
FROM  
  datahub_input;
```

3.1.4.5.3.14. Create an ApsaraDB RDS for SQL Server result table

This topic describes how to create an ApsaraDB RDS for SQL Server result table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause, data type mapping, and JDBC parameters used when you create such a result table.



Notice

- This topic applies only to Blink 3.2.0 and later.
- Realtime Compute for Apache Flink does not use ApsaraDB RDS for SQL Server as a data store.

Syntax

In Realtime Compute for Apache Flink, you can use ApsaraDB RDS for SQL Server to store output data. The following code shows an example:

```
create table ss_output(  
  id INT,  
  len INT,  
  content VARCHAR,  
  primary key(id,len)  
) with (  
  type='jdbc',  
  url='jdbc:sqlserver://ip:port;database=****',  
  tableName='<yourDatabaseTableName>',  
  userName='<yourDatabaseUserName>',  
  password='<yourDatabasePassword>'  
);
```



Note

- In Realtime Compute for Apache Flink, each row of output data is converted to a line of SQL statement and then written and executed in the destination database. If you want to write multiple rows of output data at a time, you must add ?rewriteBatchedStatements=true to the end of the URL. This improves system efficiency.
- You can define an auto-increment primary key for an ApsaraDB RDS for SQL Server database. If you want to use the auto-increment primary key, do not declare the auto-increment field in the DDL statement. For example, if you use the ID field as an auto-increment field, do not declare the ID field in the DDL statement. When a row of output data is written into an ApsaraDB RDS for SQL Server database, a value is automatically filled for the auto-increment field.
- If a DRDS result table has partitions, the shard key must be declared in primary key() of the DDL statement. Otherwise, you cannot write data into the partitioned table.
- The fields declared in a DDL statement must include at least one non-primary key field. Otherwise, an error is returned.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the result table.	Yes	Set the value to jdbc.

Parameter	Description	Required	Remarks
url	The Java Database Connectivity (JDBC) URL of the database.	Yes	None.
tableName	The name of the table.	Yes	None.
username	The username that is used to access the database.	Yes	None.
password	The password that is used to access the database.	Yes	None.
maxRetryTimes	The maximum number of retries for writing data into the table.	No	Default value: 10.
bufferSize	The maximum number of data records that can be stored in the buffer before deduplication is triggered.	No	Default value: 10000. This value indicates that deduplication is triggered after the number of input data records reaches 10,000.
flushIntervalMs	The time interval at which the buffer is cleared.	No	Unit: milliseconds. Default value: 2000. This value indicates that if the number of input data records does not reach the value specified by the bufferSize parameter within 2,000 milliseconds, all cached data is written into the result table.
excludeUpdateColumns	Specifies whether to ignore the update of a specific field.	No	This parameter is optional. It is empty by default, which indicates that the primary key field is ignored by default. When data with the same primary key is updated, the specified columns are not updated.
ignoreDelete	Specifies whether to skip delete operations.	No	Default value: False.

Data type mapping

Data type of ApsaraDB RDS for SQL Server	Data type of Realtime Compute for Apache Flink
BOOLEAN	BOOLEAN
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DECIMAL	DECIMAL
DOUBLE	DOUBLE
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
VARBINARY	VARBINARY

JDBC parameters

Parameter	Description	Default value	Required JDBC version
useUnicode	Specifies whether to use the Unicode character set. This parameter must be set to True if you set the characterEncoding parameter to GB2312 or GBK.	False	1.1g
characterEncoding	The character encoding format, such as GB2312 or GBK. If useUnicode is set to true, you must specify a character encoding format.	False	1.1g

Parameter	Description	Default value	Required JDBC version
autoReconnect	Specifies whether to automatically re-establish a connection if the connection to the database is unexpectedly interrupted.	False	1.1
autoReconnectForPools	Specifies whether to apply the reconnection policy to a database connection pool.	False	3.1.3
failOverReadOnly	Specifies whether to set the connection to read-only after the database is automatically reconnected.	True	3.0.12
maxReconnects	The maximum number of reconnection attempts allowed. This parameter must be set if the autoReconnect parameter is set to True.	3	1.1
initialTimeout	The interval between two reconnection attempts. Unit: seconds. This parameter must be set if the autoReconnect parameter is set to True.	2	1.1
connectTimeout	The timeout period when you use a socket connection to access the database server. Unit: milliseconds.	Default value: 0. This value indicates that the connection never times out. This parameter is provided in JDK V1.4 and later.	3.0.1
socketTimeout	The timeout period for read and write operations on a socket connection. Unit: milliseconds.	The value 0 indicates that the read or write operation never times out.	3.0.1

Sample code

The following example demonstrates how to create an ApsaraDB RDS for SQL Server result table in a Realtime Compute for Apache Flink job.


```
CREATE TABLE source (  
  id INT,  
  len INT,  
  content VARCHAR  
) with (  
  type = 'random'  
);  
CREATE TABLE rds_output(  
  id INT,  
  len INT,  
  content VARCHAR,  
  PRIMARY KEY (id,len)  
) WITH (  
  type='jdbc',  
  url='<yourDatabaseURL>',  
  tableName='<yourDatabaseTable>',  
  userName='<yourDatabaseUserName>',  
  password='<yourDatabasePassword>'  
);  
INSERT INTO rds_output  
SELECT id, len, content FROM source;
```

FAQ

- Q: When the output data of Realtime Compute for Apache Flink is written into an ApsaraDB RDS for SQL Server table, is the result table updated based on the primary key or is a new data record generated in the table?

A: The processing method depends on whether the primary key is defined in the DDL statement.

- If a primary key is defined in the DDL statement, the result table is updated by using `insert into on duplicate key update`. For a data record, if the primary key does not exist, the record is inserted into the table as a new row. If the value of the primary key field exists, the original row in the table is updated.
 - If no primary key is defined in the DDL statement, new data records are appended to the table by using `insert into`.
- Q: What do I need to pay attention to when I perform GROUP BY operations based on the unique index of an ApsaraDB RDS for SQL Server table?

A: Pay attention to the following items:

- Declare the unique index in the primary key of your job.
- An ApsaraDB RDS for SQL Server table has only one auto-increment primary key. Therefore, this auto-increment primary key cannot be declared as the primary key in a Realtime Compute for Apache Flink job.

3.1.4.5.3.15. Create an ApsaraDB for MongoDB result table

This topic describes how to create an ApsaraDB for MongoDB result table in Realtime Compute for Apache Flink. This parameter also describes the parameters in the WITH clause that is used when you create an ApsaraDB for MongoDB result table.

Notice

- This topic applies to only Blink V3.2.2 and later.
- You cannot update the primary key in an ApsaraDB for MongoDB result table. As a result, data that has the same primary key is inserted repeatedly into the table.

Syntax

In Realtime Compute for Apache Flink, you can use ApsaraDB for MongoDB to store output data. To create an ApsaraDB for MongoDB result table, you can use the following sample code:

```
CREATE TABLE mongodb_sink (  
  `a`      VARCHAR  
) WITH (  
  type = 'mongodb',  
  database = '<yourDatabaseName>',  
  collection = '<yourCollectionName>',  
  uri = 'mongodb://{<databaseAccount>}:{<atabasePassword>}@{host}:****? replicaSet=mgset-1224****',  
  keepAlive = 'true',  
  maxConnectionIdleTime = '20000',  
  batchSize = '2000'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the connector.	Yes	Set the value to mongodb.
database	The name of the ApsaraDB for MongoDB database.	Yes	None.
collection	The set of result table data.	Yes	None.
uri	The connection string of the ApsaraDB for MongoDB database.	Yes	None.
keepAlive	Specifies whether to maintain the persistent connection.	No	Default value: true.

Parameter	Description	Required	Remarks
maxConnectionIdleTime	The time-out duration of the connection.	No	The integer value. Unit: milliseconds. This parameter cannot be set to a negative value. Default value: 60000. If this parameter is set to 0, the connection does not time out.
batchSize	The number of data records that can be written at a time.	No	The integer value. Default value: 1024. The system sets the maximum number of data records that can be stored in the buffer. When the number of data records reaches the specified value of batchSize, the system triggers the data output.  Note When the checkpoint time arrives, the data output is triggered even if the number of data records in the buffer does not reach the specified value of batchSize.

3.1.4.5.3.16. Create a result table in an Oracle database

This topic describes how to create a result table in an Oracle database. This topic also describes the parameters in the WITH clause, the mappings between the field data types of Oracle and Realtime Compute for Apache Flink, and the considerations. You may need to use these items when you create the table.



Notice

- This topic applies to only Blink V3.6.0 and later.
- You can create result tables in an Oracle database only when you use Oracle 19c.

Notes

- In Realtime Compute for Apache Flink, an SQL statement is executed to write each row of result data to the result table in the destination database.
- Realtime Compute for Apache Flink performs the following operations based on whether the required table exists in the Oracle database:
 - If the required table does not exist, the underlying layer of Blink creates a table in the Oracle database to store the result data.
 - If the required table exists, Realtime Compute for Apache Flink writes or updates data to the result table.
- The primary keys of logical and physical tables can be different. The primary keys of logical tables must include the primary keys of physical tables.
- Realtime Compute for Apache Flink uses the APPEND function or the UPSERT function to insert data into an Oracle database result table.

- If the primary key of the table is not defined, use the APPEND function.
- If the primary key is defined, use the UPSERT function. If the primary key does not exist, insert the primary key. If the primary key exists, update the primary key.

Syntax

In Realtime Compute for Apache Flink, you can use an Oracle database to store output data. To use an Oracle database to store a result table, you can use the following sample code:

```
CREATE TABLE oracle_sink(  
  employee_id BIGINT,  
  employee_name VARCHAR,  
  employee_age INT,  
  PRIMARY KEY(employee_id)  
) WITH (  
  type = 'oracle',  
  url = '<yourUrl>',  
  userName = '<yourUserName>',  
  password = '<yourPassword>',  
  tableName = '<yourTableName>'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Example
type	The type of the result table.	Yes	Set the value to <i>oracle</i> .
url	The connection string of the database.	Yes	Contact Alibaba Cloud O&M engineers.
userName	The username that is used to access the database.	Yes	None.
password	The password that is used to access the database.	Yes	None.
tableName	The name of the table in the database.	Yes	None.
maxRetryTimes	The maximum number of attempts to insert data into the result table.	No	Default value: 10.

Parameter	Description	Required	Example
batchSize	The number of data records that are written at a time.  Note - The batchSize parameter takes effect only after the primary key is defined. - Write operations are triggered when the value of the batchSize or bufferSize parameter reaches the specified threshold.	No	Default value: 50.
bufferSize	The number of data records in the buffer after duplicates are removed.  Note - The batchSize parameter takes effect only after the primary key is defined. - Write operations are triggered when the value of the batchSize or bufferSize parameter reaches the specified threshold.	No	Default value: 500.

Parameter	Description	Required	Example
flushIntervalMs	The time-out period of the write operation. Unit: milliseconds.  Note If no data is written to the database within the period that is specified by this parameter, the system writes the cached data to the result table again.	No	Default value: 500.
excludeUpdateColumns	Specifies whether to skip the specified columns when the data of a row in the table is updated.  Note By default, the primary key is not updated when the data of a row in the table is updated.	No	By default, this parameter is empty.
ignoreDelete	Specifies whether to ignore delete operations.	No	Default value: false.

Field type mappings

Oracle field type	Realtime Compute for Apache Flink field type
- CHAR - VARCHAR - VARCHAR2	VARCHAR
FLOAT	DOUBLE
NUMBER	BIGINT
DECIMAL	DECIMAL

Sample code

For a Realtime Compute for Apache Flink job, you can use the following sample code to create a result table in an Oracle database.

```
CREATE TABLE oracle_source (  
  employee_id BIGINT,  
  employee_name VARCHAR,  
  employ_age INT  
) WITH (  
  type = 'random'  
);  
CREATE TABLE oracle_sink(  
  employee_id BIGINT,  
  employee_name VARCHAR,  
  employ_age INT,  
  primary key(employee_id)  
)with(  
  type = 'oracle',  
  url = 'jdbc:oracle:thin:@192.168.171.62:1521:sit0',  
  userName = 'blink_test',  
  password = 'blink_test',  
  tableName = 'oracle_sink'  
);  
INSERT INTO oracle_sink  
SELECT * FROM oracle_source;
```

3.1.4.5.3.17. Create an InfluxDB result table

This topic describes how to create an InfluxDB result table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH clause and data type mapping involved when you create an InfluxDB result table.



Notice

- InfluxDB does not support the storage registration feature.
- This topic applies only to Blink 3.5.0-hotfix and later.

DDL syntax

In Realtime Compute for Apache Flink, you can use InfluxDB to store output data. The following code shows an example:

```
create table stream_test_influxdb(  
  `metric` varchar,  
  `timestamp` BIGINT,  
  `tag_value1` varchar,  
  `field_fieldValue1` Double  
)with(  
  type = 'influxdb',  
  endpoint = 'http://service.cn.influxdb.aliyuncs.com:****',  
  database = '<yourDatabaseName>',  
  batchPutsize = '1',  
  username = '<yourDatabaseUserName>',  
  password = '<yourDatabasePassword>'  
);
```

Default format for the created table:

- Column 0: metric (VARCHAR). This column is required.
- Column 1: timestamp (BIGINT). This column is required. Unit: milliseconds.
- Column 2: tag_value1 (VARCHAR). This column is required. You must enter at least one value in this column.
- Column 3: field_fieldValue1 (DOUBLE). This column is required. You must enter at least one value in this column.

To specify multiple field_fieldValue values, use the following format:

```
field_fieldValue1 <Data type>,
field_fieldValue2 <Data type>,
...
field_fieldValueN <Data type>
```

The following code shows an example:

```
field_fieldValue1 Double,
field_fieldValue2 INTEGER,
...
field_fieldValueN INTEGER
```

 **Note** An InfluxDB result table can contain only metric, timestamp, tag_*, and field_.*

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the result table.	Yes	Set the value to InfluxDB.
endpoint	The endpoint of the InfluxDB database.	Yes	Contact Alibaba Cloud O&M engineers.
database	The name of the InfluxDB database.	Yes	For example, you can set this parameter to db-blink or blink.
batchPutSize	The number of data records that are submitted at a time.	No	Default value: 500.
username	The username that is used to access the InfluxDB database.	Yes	You must have the write permission on the InfluxDB database.
password	The password that is used to access the InfluxDB database.	Yes	Default value: 0.
retentionPolicy	The retention policy.	No	If this parameter is empty, the default retention policy is used for each database.

Field type mapping

InfluxDB data type	Data type of Realtime Compute for Apache Flink
BOOLEAN	BOOLEAN
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DECIMAL	DECIMAL
DOUBLE	DOUBLE
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR

3.1.4.5.3.18. Create a Phoenix5 result table

This topic describes how to create a Phoenix5 result table in Realtime Compute for Apache Flink.



Notice

- This topic applies only to Realtime Compute for Apache Flink in exclusive mode.
- This topic applies only to Blink 3.4.0 and later.
- Only Phoenix 5.X is supported.
- Phoenix is an HBase SQL service deployed on an ApsaraDB for HBase instance. You can use Phoenix only after you activate this service in ApsaraDB for HBase instances.

DDL syntax

In Realtime Compute for Apache Flink, you can use Phoenix5 to store output data. The following code shows an example:

```
create table US_POPULATION_SINK (  
  `STATE` varchar,  
  CITY varchar,  
  POPULATION BIGINT,  
  PRIMARY KEY (`STATE`, CITY)--- The primary key. This field is required.  
) WITH (  
  type = 'PHOENIX5',  
  serverUrl = '<yourserverUrl>',  
  tableName = '<yourTableName>'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the result table.	Yes	Set the value to PHOENIX5 .
serverUrl	The address of the Phoenix5 Query Server. - If Phoenix5 is created in a cluster, the URL of Server Load Balancer (SLB) is used. - If Phoenix5 is created on a server, the URL of the server is used.	Yes	You must activate the HBase SQL service in an ApsaraDB for HBase instance..
tableName	The name of the Phoenix5 table.	Yes	None.

Sample code

The following sample code demonstrates how to create a Phoenix5 result table in a Realtime Compute for Apache Flink job.

```
create table `source` (
  `id` varchar,
  `name` varchar,
  `age` varchar,
  `birthday` varchar
) WITH (
  type = 'random'
);
create table sink (
  `id` varchar,
  `name` varchar,
  `age` varchar,
  `birthday` varchar,
  primary key (id)
) WITH (
  type = 'PHOENIX5',
  serverUrl = '<yourserverUrl>',
  tableName = '<yourTableName>'
);
INSERT INTO sink
SELECT `id`, `name`, `age`, `birthday`
FROM `source`;
```

3.1.4.5.3.19. Create a Hologres result table

This topic describes how to create a Hologres result table. It also describes the parameters in the WITH clause, streaming semantics, and data type mapping involved when you create a Hologres result table.



Notice

- This topic applies only to Blink 3.6.0 and later.
- We recommend that you use Hologres 0.7 or later.
- Hologres writes data asynchronously. Therefore, you must add `blink.checkpoint.fail_on_checkpoint_error=true` to the code so that a failover is triggered only when a job exception occurs.

Introduction to Hologres

Hologres is compatible with the PostgreSQL protocol and integrates seamlessly with the big data ecosystem. Hologres supports real-time analysis and the processing of petabytes of data with high concurrency and low latency. This allows you to use existing Business Intelligence (BI) tools to easily perform multidimensional analysis and business exploration.

Syntax

In Realtime Compute for Apache Flink, you can use Hologres to store output data. The following code shows an example:

```
create table Hologres_sink(  
  name varchar,  
  age BIGINT,  
  birthday BIGINT  
) with (  
  type='hologres',  
  dbname='<yourDbname>',  
  tablename='<yourTablename>',  
  username='<yourUsername>',  
  password='<yourPassword>',  
  endpoint='<yourEndpoint>',  
  field_delimiter='|' -- This parameter is optional.  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the result table.	Yes	Set the value to <i>hologres</i> .
dbname	The name of the database.	Yes	None.
tablename	The name of the table.	Yes	None.
username	The username that is used to access the database.	Yes	None.

Parameter	Description	Required	Remarks
password	The password that is used to access the database.	Yes	None.
endpoint	The VPC endpoint of Hologres.	Yes	Contact Alibaba Cloud O&M engineers.
field_delimiter	The delimiter used between rows when data is being exported.  Notice Do not insert delimiters in data. This parameter takes effect only when the bulkload semantics is used.	No	Default value: <code>\u0002</code> .
mutateType	The streaming write semantics. For more information, see Streaming semantics .	No	Default value: <i>insertorignore</i> .
partitionrouter	Specifies whether to write data into a partitioned table.	No	Default value: <i>false</i>.  Note For Blink 3.6.X, Hologres streaming sink cannot automatically create partitioned tables. Before Realtime Compute for Apache Flink writes data to a partitioned table, you must manually create a sub-table in Hologres.
ignoredelete	Specifies whether to skip retract messages.	No	Default value: <i>false</i>.  Note This parameter takes effect only when the streaming semantics is used.

Parameter	Description	Required	Remarks
createPartTable	Specifies whether a non-existent partitioned table can be automatically created based on partition values when Realtime Compute for Apache Flink writes data into a partitioned table. Note If a partition value contains a hyphen (-), partitioned tables cannot be automatically created.	No	- false (default value): Partitioned tables cannot be automatically created. - true: Partitioned tables can be automatically created. Note Only Blink versions later than 3.7 support this parameter.

Streaming semantics

Stream processing, also known as streaming data or event processing, involves the processing of a series of unbounded data or events. A system that processes streaming data or events usually allows you to specify a reliability mode or processing semantics to ensure data accuracy because network or device faults may cause data loss.

Based on the configurations of Hologres streaming sink and Hologres table attributes, the following two types of semantics are supported:

- Exactly-once: The system processes data or events only once even multiple faults occur.
- At-least-once: If data or an event is lost before stream processing is completed, the system transfers all data and events again. In this case, data or events may be processed multiple times. If the first retry succeeds, no further retries are required.

When you use streaming semantics in a Hologres result table, take note of the following items:

- If no primary keys have been configured in the Hologres physical table, Hologres streaming sink uses the at-least-once semantics.
- If primary keys have been configured in the Hologres physical table, Hologres streaming sink uses the exactly-once semantics based on the primary keys. If multiple records with the same primary key are written to the table, you must set the mutationType parameter to determine how the result table is updated. This parameter has the following valid values:
 - insertorignore (default value): Hologres keeps the first record and discards the subsequent records.
 - insertorreplace: Hologres completely replaces the existing record with the one that arrives later.
 - insertorupdate: Hologres partially replaces the existing record with the one that arrives later.

? Note

- If the `mutationType` parameter is set to `insertorupdate` or `insertorreplace`, the system updates data based on the primary key.
 - The number of columns in the result table defined by Blink can be different from the number of columns in the Hologres physical table. Make sure that the value null can be used to fill the missing columns. Otherwise, an error is returned.
- By default, Hologres streaming sink can import data to only one non-partitioned table. If the sink imports data to the parent table of a partitioned table, data queries fail even if data import is successful. To enable data to be automatically written into a partitioned table, you can set the `partitionRouter` parameter to `true`. Take note of the following items:
 - You must set `tablename` to the name of the parent table.
 - Blink connectors do not automatically create partitioned tables. We recommend that you create a partitioned table before you import data. Otherwise, the data fails to be imported.

Field type mapping

Hologres	BLINK
INT	INT
INT []	ARRAY<INT>
BIGINT	BIGINT
BIGINT []	ARRAY<BIGINT>
REAL	FLOAT
REAL[]	ARRAY<FLOAT>
DOUBLE PRECISION	DOUBLE
DOUBLE PRECISION[]	ARRAY<DOUBLE>
BOOLEAN	BOOLEAN
BOOLEAN[]	ARRAY<BOOLEAN>
TEXT	VARCHAR
TEXT[]	ARRAY<VARCHAR>
NUMERIC	DECIMAL
DATE	DATE
TIMESTAMP WITH TIMEZONE	TIMESTAMP

3.1.4.5.3.20. Create an AnalyticDB for PostgreSQL result table

This topic describes how to create an AnalyticDB for PostgreSQL result table. This topic also describes the parameters in the WITH clause and data type mappings used when you create an AnalyticDB for PostgreSQL result table.

 **Notice** This topic applies to only Blink 3.6.0 and later.

Principles

Realtime Compute for Apache Flink writes data to an AnalyticDB for PostgreSQL result table in two steps:

1. Converts each row of output data to a line of SQL statement.
2. Writes and executes the SQL statement in the destination database.

DDL syntax

In Realtime Compute for Apache Flink, you can use AnalyticDB for PostgreSQL to store output data. The following sample code shows how to create an AnalyticDB for PostgreSQL result table.

```
create table rds_output(  
  id INT,  
  len INT,  
  content VARCHAR,  
  PRIMARY KEY(id)  
) with (  
  type='adbpg',  
  url='jdbc:postgresql://<yourNetworkAddress>:<PortId>/<yourDatabaseName>',  
  tableName='<yourDatabaseTableName>',  
  userName='<yourDatabaseUserName>',  
  password='<yourDatabasePassword>'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the source table.	Yes	Set the value to <code>adbpg</code> .
url	The Java Database Connectivity (JDBC) URL of the database.	Yes	Contact Alibaba Cloud O&M engineers to obtain the required information.
tableName	The name of the table.	Yes	N/A

Parameter	Description	Required	Remarks
username	The account that is used to access the AnalyticDB for PostgreSQL database.	Yes	N/A
password	The password that is used to access the AnalyticDB for PostgreSQL database.	Yes	N/A
maxRetryTimes	The maximum number of retries for writing data to the table.	No	Default value: 3.
useCopy	Specifies whether to use the copy API to write data.	No	Valid values: 0: indicates that the INSERT statement is executed to write data to the AnalyticDB for PostgreSQL database. This is the default value. 1: indicates that the copy API is used to write data to the AnalyticDB for PostgreSQL database.
batchSize	The number of data records that can be written at a time.	No	Default value: 5000.
exceptionMode	The policy that is used to handle exceptions during data writing.	No	Valid values: <i>ignore</i>: The system ignores the data that is written when exceptions occur. This is the default value. <i>strict</i>: If an exception occurs during data writing, an error message appears on the Failover tab.
conflictMode	The policy that is used to handle primary key conflicts or unique index conflicts.	No	Valid values: <i>ignore</i>: Primary key conflicts are ignored and the existing data is retained. This is the default value. <i>strict</i>: If a primary key conflict occurs, an error message appears on the Failover tab. <i>update</i>: If a primary key conflict occurs, data is updated.
targetSchema	The name of the schema.	No	Default value: public.

Parameter	Description	Required	Remarks
connection MaxActive	The maximum number of connections allowed for a single task.	No	Configure this parameter based on the actual number of parallel tasks and the maximum number of connections allowed to the destination database.

Data type mapping

The following table lists the mappings between the field data types of AnalyticDB for PostgreSQL and Realtime Compute for Apache Flink.

Data type of AnalyticDB for PostgreSQL	Data type of Realtime Compute for Apache Flink
BOOLEAN	BOOLEAN
SMALLINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
DOUBLE PRECISION	DOUBLE
TEXT	VARCHAR
TIMESTAMP	DATETIME
DATE	DATE
REAL	FLOAT
DOUBLE PRECISION	DECIMAL
TIME	TIME
TIMESTAMP	TIMESTAMP

3.1.4.5.3.21. Create an ApsaraDB for HBase result table

This topic describes how to create an ApsaraDB for HBase result table in Realtime Compute for Apache Flink.

 Notice

- This topic applies only to Realtime Compute for Apache Flink in exclusive mode.
- Blink versions earlier than 3.3.0 support only HBase Standard Edition.
- Blink 3.3.0 and later versions support both HBase Standard Edition and HBase Enhanced Edition.
- Blink 3.5.0 and later versions support switchover between primary and secondary ApsaraDB for HBase databases for data writing.
- ApsaraDB for HBase result tables in Realtime Compute for Apache Flink do not support self-managed open source HBase.

DDL syntax

In Realtime Compute for Apache Flink, you can use ApsaraDB for HBase to store output data.

- The following sample code demonstrates how to create an ApsaraDB for HBase result table of HBase Standard Edition:

```
create table liuxd_user_behavior_test_front (
  row_key varchar,
  from_topic varchar,
  origin_data varchar,
  record_create_time varchar,
  primary key (row_key)
) with (
  type = 'cloudhbase',
  zkQuorum = '2',
  columnFamily = '<yourColumnFamily>',
  tableName = '<yourTableName>',
  batchSize = '500'
);
```

- The following sample code demonstrates how to create an ApsaraDB for HBase result table of HBase Enhanced Edition:

```
create table liuxd_user_behavior_test_front (
  row_key varchar,
  from_topic varchar,
  origin_data varchar,
  record_create_time varchar,
  primary key (row_key)
) with (
  type = 'cloudhbase',
  endPoint = '<host:port>', ----The Java API URL that is used to access the Enhanced Edition of an ApsaraDB for HBase database.
  userName = 'root', -- The username that is used to access the ApsaraDB for HBase database.
  password = 'root', --The password that is used to access the ApsaraDB for HBase database.
  columnFamily = '<yourColumnFamily>',
  tableName = '<yourTableName>',
  batchSize = '500'
);
```

- The following sample code demonstrates how to create an ApsaraDB for HBase dimension table of

HBase Enhanced Edition in Blink 3.5.0 or later:

```
create table liuxd_user_behavior_test_front (
  row_key varchar,
  from_topic varchar,
  origin_data varchar,
  record_create_time varchar,
  primary key (row_key)
) with (
  type = 'cloudhbase',
  zkQuorum = '<host:port>', ----The Java API URL that is used to access the Enhanced Edition of an Apsara
DB for HBase database.
  userName = 'root', --The username that is used to access the ApsaraDB for HBase database.
  password = 'root', --The password that is used to access the ApsaraDB for HBase database.
  columnFamily = '<yourColumnFamily>',
  tableName = '<yourTableName>',
  batchSize = '500'
);
```

- The following sample code demonstrates how to create an ApsaraDB for HBase dimension table in Blink 3.5.0 or later that supports switchover between primary and secondary ApsaraDB for HBase databases for data writing:

```
create table liuxd_user_behavior_test_front (
  row_key varchar,
  from_topic varchar,
  origin_data varchar,
  record_create_time varchar,
  primary key (row_key)
) with (
  type = 'cloudhbase',
  zkQuorum = '<host:port>', ---- The URL that is used to access ApsaraDB for HBase databases in high avail
ability (HA) mode.
  haClusterID = 'ha-xxx', ---- The instance ID of ApsaraDB for HBase databases in HA mode.
  userName = 'root', --The username that is used to access the ApsaraDB for HBase database.
  password = 'root', --The password that is used to access the ApsaraDB for HBase database.
  columnFamily = '<yourColumnFamily>',
  tableName = '<yourTableName>',
  batchSize = '500'
);
```

Note

- You can define multiple fields for the primary key. Multiple fields are separated with the value of `rowkeyDelimiter`. The default value is a colon (:).
- When you undo the deletion operation in ApsaraDB for HBase, if a column stores multiple versions of a value, all the versions of the value are deleted.
- The connection parameters are different between ApsaraDB for HBase Standard Edition and ApsaraDB for HBase Enhanced Edition.
 - In ApsaraDB for HBase Standard Edition, the connection parameter is `zkQuorum`.
 - In ApsaraDB for HBase Enhanced Edition, the connection parameter is `endPoint`.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the result table.	Yes	Set the value to cloudhbase.
zkQuorum	The ZooKeeper address that is configured for the ApsaraDB for HBase cluster.	Yes	You can view the configuration related to hbase.zookeeper.quorum in the <i>hbase-site.xml</i> file.  Note This parameter takes effect in only ApsaraDB for HBase Standard Edition.
zkNodeParent	The path of the cluster configured on the ZooKeeper servers.	No	You can view the relevant configurations of hbase.zookeeper.quorum in the <i>hbase-site.xml</i> file.  Note This parameter takes effect only in ApsaraDB for HBase Standard Edition.
endPoint	The name of the region where your ApsaraDB for HBase instance is deployed.	Yes	Contact Alibaba Cloud O&M engineers.
userName	The username that is used to access an ApsaraDB for HBase database.	No	This parameter takes effect only in ApsaraDB for HBase Enhanced Edition.
password	The password that is used to access an ApsaraDB for HBase database.	No	This parameter takes effect only in ApsaraDB for HBase Enhanced Edition.
tableName	The name of the ApsaraDB for HBase table.	Yes	None.
columnFamily	The name of the column family.	Yes	Only the same column family can be inserted.
maxRetryTimes	The maximum number of retries.	No	Default value: 10.  Note This parameter is available only in Realtime Compute for Apache Flink V3.2.3 and later.

Parameter	Description	Required	Remarks
bufferSize	The maximum number of data records that can be stored in the buffer before deduplication is triggered.	No	Default value: 5000.
batchSize	The number of data records that can be written at a time.	No	Default value: 100.  Note We recommend that you set this parameter to a value that ranges from 200 to 300. A large value of batchSize may cause an out-of-memory (OOM) error for the task.
flushIntervalMs	The interval at which the buffer is cleared to reduce the latency of writing data into ApsaraDB for HBase.	No	Default value: 2000. Unit: milliseconds.
writePkValue	Specifies whether to write the primary key value.	No	Default value: false.
stringWriteMode	Specifies whether to insert data as the STRING type.	No	Default value: false.
rowkeyDelimiter	The delimiter of rowKey.	No	The default delimiter is a colon (:).
isDynamicTable	Specifies whether the table is a dynamic table.	No	Default value: false.
haClusterID	The ID of an ApsaraDB for HBase instance in HA mode.	No	This parameter is required only when you access zone-disaster recovery instances.

Dynamic table

Some result data of Realtime Compute for Apache Flink is used as a dynamic column based on the value of a column and written to ApsaraDB for HBase. In the following example, the turnover per hour is used as a dynamic column data in ApsaraDB for HBase.

rowkey	cf:0	cf:1	cf:2
20170707	100	cf:1	300

If `isDynamicTable` is set to `true`, the table is an ApsaraDB for HBase table that supports dynamic columns.

A dynamic table can contain only three columns, such as `ROW_KEY`, `COLUMN`, and `VALUE`. The second column that is `COLUMN` in this example is a dynamic column. Other parameters in the dynamic table are the same as those in the `WITH` clause of ApsaraDB for HBase.

 **Note** When a dynamic table is used, all data types must be converted to the `STRING` type before data is written into ApsaraDB for HBase.

```
CREATE TABLE stream_test_hotline_agent (
  name varchar,
  age varchar,
  birthday varchar,
  primary key (name)
) WITH (
  type = 'cloudhbase',
  ...
  columnFamily = 'cf',
  isDynamicTable = 'true'
);
```

 **Note**

- In the preceding declaration, `birthday` is inserted into the `cf:age` column with `ROW_KEY` of `name`. For example, `(wang.18,2016-12-12)` is inserted into the row for which the `ROW_KEY` value is `wang` and the `cf:18` column.
- In the DDL statement, you must declare `ROW_KEY` as the primary key and declare the following fields in the descending order: `ROW_KEY`, `COLUMN`, and `VALUE`. In this example, `ROW_KEY` is `name`, `COLUMN` is `age`, and `VALUE` is `birthday`.

Sample code

The following sample code demonstrates how to create an ApsaraDB for HBase result table in a Realtime Compute for Apache Flink job:

```
create table source (  
  id TINYINT,  
  name BIGINT  
) with (  
  type = 'random'  
);  
create table sink (  
  id TINYINT,  
  name BIGINT,  
  primary key (id)  
) with (  
  type = 'cloudhbase',  
  zkQuorum = '<yourZkQuorum>',  
  columnFamily = '<yourColumnFamily>',  
  tableName = '<yourTableName>'  
);  
INSERT INTO sink  
SELECT id, name FROM source;
```

FAQ

Why is the error `cloudHbase update error, No columns to insert for #10 item` reported when a job for an ApsaraDB for HBase result table is running?

The column data for a single record that is written into the ApsaraDB for HBase result table cannot all be null. The column data excludes ROW_KEY. Before you use Realtime Compute for Apache Flink to write data into the ApsaraDB for HBase result table, filter out all null data.

3.1.4.5.4. Create a data dimension table

3.1.4.5.4.1. Overview

You must add `PERIOD FOR SYSTEM TIME` in the CREATE TABLE statement that is used to create a dimension table. `PERIOD FOR SYSTEM_TIME` defines the period of time when a data record is valid. This means that the table is frequently updated.

```
CREATE TABLE white_list (  
  id varchar,  
  name varchar,  
  age int,  
  PRIMARY KEY (id), -- You must define a primary key for the dimension table.  
  PERIOD FOR SYSTEM_TIME -- Defines the period of time when a data record is valid.  
) with (  
  type = 'rds',  
  ...  
)
```

 Note

- When you declare a dimension table, you must specify the primary key.
- When you join a dimension table with another table, the ON condition must contain an equivalence condition that includes the primary key of either table.

3.1.4.5.4.2. Create a Tablestore dimension table

This topic describes how to create a Tablestore dimension table in Realtime Compute for Apache Flink.

 **Notice** This topic applies only to Blink 1.4.5 and later.

Introduction to Tablestore

Tablestore is a distributed NoSQL database service that is built on the Apsara distributed operating system of Alibaba Cloud. Tablestore adopts sharding and load balancing technologies to scale out services and handle concurrent transactions. You can use Tablestore to store and query large amounts of structured data in real time.

Example

In Realtime Compute for Apache Flink, you can use a Tablestore table as a dimension table. The following code shows an example:

```
CREATE TABLE ots_dim_table (  
  id int,  
  len int,  
  content VARCHAR,  
  PRIMARY KEY (id),  
  PERIOD FOR SYSTEM_TIME--Define the change period of the dimension table.  
) WITH (  
  type='ots',  
  endPoint='<yourEndpoint>',  
  instanceName='<yourInstanceName>',  
  tableName='<yourTableName>',  
  accessId='<yourAccessId>',  
  accessKey='<yourAccessKey>'  
);
```

 Note

- When you declare a dimension table, you must specify a primary key.
- When you join a dimension table with another table, the ON clause must contain the equivalent (=) conditions for all the primary key fields.
- The primary key of a Tablestore table is the row key of the table.

Parameters in the WITH clause

Parameter	Description	Remarks
type	The type of the dimension table.	Set the value to <code>ots</code> .
endPoint	The endpoint of the Tablestore instance.	Contact Alibaba Cloud O&M engineers.
instanceName	The name of the Tablestore instance.	None.
tableName	The name of the Tablestore dimension table.	None.
accessId	The AccessKey ID that is used to access Tablestore.	None.
accessKey	The AccessKey secret that is used to access Tablestore.	None.

Parameters in the CACHE clause

Parameter	Description	Remarks
cache	The cache policy.	Valid values: <i>None</i>: indicates that data is not cached. This is the default cache policy. <i>LRU</i>: indicates that only the specified data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system does not find the record in the cache, it searches for the data record in the physical dimension table. If you use this cache policy, you must specify the <code>cacheSize</code> and <code>cacheTTLms</code> parameters.
cacheSize	The maximum number of rows that can be cached.	You can set this parameter when the cache parameter is set to <i>LRU</i> . Default value: 10000.
cacheTTLms	The cache timeout period. Unit: milliseconds.	You can set this parameter when the cache parameter is set to <i>LRU</i> .

Sample code

```
CREATE TABLE datahub_input1 (  
  id    BIGINT,  
  name  VARCHAR,  
  age   BIGINT  
) WITH (  
  type='datahub'  
);  
CREATE TABLE phoneNumber(  
  name VARCHAR,  
  phoneNumber bigint,  
  primary key(name),  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
)with(  
  type='ots'  
);  
CREATE TABLE result_infor(  
  id bigint,  
  phoneNumber bigint,  
  name VARCHAR  
)with(  
  type='rds'  
);  
INSERT INTO result_infor  
SELECT  
  t.id,  
  w.phoneNumber,  
  t.name  
FROM datahub_input1 as t  
JOIN phoneNumber FOR SYSTEM_TIME AS OF PROCTIME() as w --You must include this clause when you perform a JOIN operation on the dimension table.  
ON t.name = w.name;
```

For more information about the syntax for dimension tables, see [JOIN statements for dimension tables](#).

3.1.4.5.4.3. Create an ApsaraDB for HBase dimension table

This topic describes how to create an ApsaraDB for HBase dimension table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH and CACHE clauses used when you create such a dimension table.

Notice

- Blink versions earlier than Blink 3.3.0 support only HBase Standard Edition.
- Blink 3.3.0 and later versions support HBase Standard Edition and HBase Enhanced Edition.
- Blink 3.5.0 and later versions support switchover between primary and secondary ApsaraDB for HBase databases for data writing.
- For more information about the JOIN syntax of an ApsaraDB for HBase dimension table, see [JOIN statements for dimension tables](#).
- ApsaraDB for HBase dimension tables in Realtime Compute for Apache Flink do not support user-created open source HBase.
- Only one primary key is allowed in an ApsaraDB for HBase dimension table.

DDL syntax

- HBase Standard Edition

```
CREATE TABLE hbase (  
  `key` varchar,  
  `name` varchar,  
  PRIMARY KEY (`key`), -- The rowkey field of the ApsaraDB for HBase dimension table.  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) with (  
  TYPE = 'cloudhbase',  
  zkQuorum = '<yourzkQuorum>',  
  columnFamily = '<yourColumnFamilyName>',  
  tableName = '<yourTableName>'  
);
```

- HBase Enhanced Edition

```
CREATE TABLE hbase (  
  `key` varchar,  
  `name` varchar,  
  PRIMARY KEY (`key`), -- The rowkey field of the ApsaraDB for HBase dimension table.  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) with (  
  TYPE = 'cloudhbase',  
  endPoint = '<host:port>', -- The Java API URL that is used to access the Enhanced Edition of an ApsaraDB for HBase database.  
  userName = 'root', -- The username that is used to access an ApsaraDB for HBase database.  
  password = 'root', -- The password that is used to access an ApsaraDB for HBase database.  
  columnFamily = '<yourColumnFamilyName>',  
  tableName = '<yourTableName>'  
);
```

- HBase Enhanced Edition for Blink 3.5.0 and later

```
create table liuxd_user_behavior_test_front (
  row_key varchar,
  from_topic varchar,
  origin_data varchar,
  record_create_time varchar,
  primary key (row_key)
) with (
  type = 'cloudhbase',
  zkQuorum = '<host:port>', -- The Java API URL that is used to access the Enhanced Edition of an ApsaraDB for HBase database.
  userName = 'root', -- The username that is used to access an ApsaraDB for HBase database.
  password = 'root', -- The password that is used to access an ApsaraDB for HBase database.
  columnFamily = '<yourColumnFamily>',
  tableName = '<yourTableName>',
  batchSize = '500'
);
```

- Blink 3.5.0 and later versions support switchover between primary and secondary ApsaraDB for HBase databases for data writing.

```
create table liuxd_user_behavior_test_front (
  row_key varchar,
  from_topic varchar,
  origin_data varchar,
  record_create_time varchar,
  primary key (row_key)
) with (
  type = 'cloudhbase',
  zkQuorum = '<host:port>', -- The URL that is used to access ApsaraDB for HBase databases in high availability (HA) mode.
  haClusterID = 'ha-xxx', -- The instance ID of ApsaraDB for HBase databases in HA mode.
  userName = 'root', -- The username that is used to access an ApsaraDB for HBase database.
  password = 'root', -- The password that is used to access an ApsaraDB for HBase database.
  columnFamily = '<yourColumnFamily>',
  tableName = '<yourTableName>',
  batchSize = '500'
);
```

Note

- When you declare a dimension table, you must specify a primary key.
- When you join a dimension table with another dimension table, the ON condition must contain equivalent conditions that include all primary keys. The primary key in the preceding examples is `row_key`.
- The connection parameters in HBase Standard Edition and HBase Enhanced Edition are different.
 - HBase Standard Edition: `zkQuorum`.
 - HBase Enhanced Edition: `endPoint`.
 - HBase Standard Edition and HBase Enhanced Edition for Blink 3.5.0 and later: `zkQuorum`.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the dimension table.	Yes	Set the value to <code>cloudhbase</code> .
zkQuorum	The ZooKeeper address configured for the ApsaraDB for HBase cluster. The address is a list of hosts separated by commas (,).	Yes	You can view the configuration related to <code>hbase.zookeeper.quorum</code> in the <code>hbase-site.xml</code> file.  Note This parameter takes effect only in HBase Standard Edition.
zkNodeParent	The path of the cluster configured on the ZooKeeper servers.	No	You can view the configuration related to <code>hbase.zookeeper.quorum</code> in the <code>hbase-site.xml</code> file.  Note This parameter takes effect only in HBase Standard Edition.
endPoint	The name of the region where your ApsaraDB for HBase instance is deployed.	Yes	You can obtain the value of this parameter from the console of your ApsaraDB for HBase instance.  Note This parameter takes effect only in HBase Enhanced Edition.
userName	The username that is used to log on to the ApsaraDB for HBase database.	No	 Note This parameter takes effect only in HBase Enhanced Edition.
password	The password that is used to log on to the ApsaraDB for HBase database.	No	 Note This parameter takes effect only in HBase Enhanced Edition.
tableName	The name of the ApsaraDB for HBase dimension table.	Yes	None.
columnFamily	The column family name.	Yes	Only the same column family can be inserted.
maxRetryTimes	The maximum number of retries for writing data into the table.	No	Default value: 10.

Parameter	Description	Required	Remarks
partitionedJoin	Specifies whether to use joinKey for partitioning.	No	Default value: False. If you set this parameter to True, joinKey is used for partitioning. Data is delivered to each node for joining, which increases the cache hit ratio.
shuffleEmptyKey	Specifies whether to randomly send upstream empty keys to downstream nodes.	No	Default value: True. Valid values: - True: If multiple empty keys exist in the upstream, Realtime Compute for Apache Flink randomly sends all the empty keys to each JOIN node. - False: If multiple empty keys exist in the upstream, Realtime Compute for Apache Flink sends all the empty keys to a single JOIN node.  Note This parameter can be used only after the partitionedJoin parameter takes effect.

Parameters in the CACHE clause

Parameter	Description	Required	Remarks

Parameter	Description	Required	Remarks
cache	The policy for caching data.	No	Valid values: <i>None</i> (default value): indicates that no data is cached. <i>LRU</i>: indicates that partial data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system does not find the record in the cache, it searches for the data record in the physical dimension table. If this cache policy is used, you must configure the <code>cacheSize</code> and <code>cacheTTLms</code> parameters. <i>ALL</i>: indicates that all data in the dimension table is cached. Before a Realtime Compute for Apache Flink job starts to run, Realtime Compute for Apache Flink loads all data in the dimension table to the cache, and then searches the cache for all subsequent queries in the dimension table. If the system does not find the data record in the cache, the join key does not exist. The system reloads all data in the cache after cache entries expire. If the amount of data of a remote table is small and a large number of missing keys exist, we recommend that you set this parameter to <i>ALL</i>. (The source table and dimension table cannot be associated based on the <i>ON</i> clause.) If this cache policy is used, you must configure the <code>cacheTTLms</code> and <code>cacheReloadTimeBlackList</code> parameters.  Note If the cache parameter is set to <i>ALL</i>, the memory of the node for joining tables must be increased because Realtime Compute for Apache Flink asynchronously loads data from the dimension table. The increased memory size is two times that of the remote table.
cacheSize	The maximum number of data records that can be cached. Unit: lines.	No	You can set this parameter when the cache parameter is set to <i>LRU</i>. Default value: 10000.

Parameter	Description	Required	Remarks
cacheTTLms	The cache timeout period. Unit: milliseconds.	No	If the cache parameter is set to LRU , the cacheTTLms parameter specifies the time before cache entries expire. Cache entries do not expire by default. If the cache parameter is set to ALL , the cacheTTLms parameter specifies the interval at which the cache is loaded. The cache is not reloaded by default.
cacheReloadTimeBlackList	The time periods during which the cache is not refreshed. This parameter is used when the cache parameter is set to ALL . The cache is not refreshed during the time periods that you specify for this parameter. This parameter is useful for massive online promotional events such as Double 11.	No	This parameter is empty by default. Custom input format: <pre>2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00</pre> Separate multiple time periods with commas (,). Separate the start time and end time of each time period with a hyphen and a greater-than sign (->).
cacheScanLimit	The maximum number of rows returned by the server to the client for each remote procedure call (RPC) when the server reads full ApsaraDB for HBase data. This parameter is used when the cache parameter is set to ALL .	No	Default value: 100.

Sample code

The following sample code demonstrates how to create an ApsaraDB for HBase dimension table in a Realtime Compute for Apache Flink job.


```
create table source (
  id TINYINT,
  name BIGINT
) with (
  type = 'random'
);
create table dim (
  id TINYINT,
  score BIGINT
  primary key(id),
  PERIOD FOR SYSTEM_TIME
)with(
  type = 'cloudhbase',
  zkQuorum = '<yourzkQuorum>',
  columnFamily = '<yourColumnFamilyName>',
  tableName = '<yourTableName>'
);
CREATE table result_infor(
  id BIGINT,
  score BIGINT
)with(
  type='rds'
);
INSERT INTO result_infor
SELECT
  t.id,
  w.score
FROM source as t
JOIN dim FOR SYSTEM_TIME AS OF PROCTIME() as w
ON t.id = w.id;
```

3.1.4.5.4.4. Create a KVStore for Redis dimension table

This topic describes how to create a KVStore for Redis dimension table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH and CACHE clauses, data type mapping, and sample code used when you create such a dimension table.

Notice

- This topic applies only to Blink 3.2.2 and later.
- KVStore for Redis dimension tables in Realtime Compute for Apache Flink can only reference data of the STRING type in KVStore for Redis databases.
- KVStore for Redis dimension tables in Realtime Compute for Apache Flink support user-created Redis databases.

Syntax

In Realtime Compute for Apache Flink, you can create a KVStore for Redis dimension table. The following code shows an example:

```
CREATE TABLE white_list (  
  id VARCHAR,  
  name VARCHAR,  
  PRIMARY KEY (id), -- The Row Key field in a KVStore for Redis database.  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) WITH (  
  type = 'redis',  
  host = '<yourHostName>',  
  port = '<yourPort>',  
  password = '<yourPassword>',  
  dbName = '<yourDatabaseNumber>'  
);
```

Note

- Only one primary key can be declared for a KVStore for Redis dimension table.
- When you join a dimension table with another dimension table, the ON condition must contain equivalent conditions of all the primary keys.
- You can declare only two fields for a KVStore for Redis dimension table, and the fields must be of the VARCHAR type.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the dimension table.	Yes	Set the value to <code>redis</code> .
host	The endpoint of a KVStore for Redis instance.	Yes	None.
port	The port of the KVStore for Redis database.	No	Default value: 6379.
dbName	The sequence number of the KVStore for Redis database.	No	Default value: 0.
password	The password that is used to access the KVStore for Redis database.	No	This parameter is empty by default, which indicates that permission verification is not required.

Parameter	Description	Required	Remarks
hashName	The hash key name in hash mode.	No	This parameter is empty by default, which indicates that Realtime Compute for Apache Flink reads data of the STRING type from the KVStore for Redis database. In typical cases, the data type in the Redis dimension table is STRING, which is represented as key-value pairs. If you set the hashName parameter, data in the KVStore for Redis dimension table is of the HASHMAP type, which is presented as key-{field-value} pairs. - key is the value of the hashName parameter. - field is the value of the key parameter that you specify in the CREATE TABLE statement. - value is the value assigned to key, which has the same meaning as value in key-value of the STRING type.

Parameters in the CACHE clause

Parameter	Description	Required	Remarks
cache	The policy for caching data.	No	Valid values: <i>None</i> (default value): indicates that no data is cached. <i>LRU</i>: indicates that partial data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system does not find the record in the cache, it searches for the data record in the physical dimension table. If this cache policy is used, you must configure the cacheSize and cacheTTLms parameters.
cacheSize	The maximum number of data records that can be cached. Unit: lines.	No	If you set the cache parameter to LRU , you can set this parameter to specify the maximum number of data records that can be cached. Default value: 10000.
cacheTTLms	The cache timeout period.	No	The cache does not expire by default. Unit: milliseconds. If the cache policy is set to LRU, this parameter specifies the time before the cache expires.

Parameter	Description	Required	Remarks
cacheEmpty	Specifies whether to clear the cache.	No	Default value: true.

Field type mapping

The following table describes the mapping between KVStore for Redis data types and data types of Realtime Compute for Apache Flink. We recommend that you declare the mapping in a DDL statement.

Data type of KVStore for Redis	Data type of Realtime Compute for Apache Flink
STRING	VARCHAR

Sample code

The following sample code demonstrates how to create a KVStore for Redis dimension table in a Realtime Compute for Apache Flink job.

```
CREATE TABLE event (  
  id VARCHAR,  
  data VARCHAR) with (  
  type = 'random'  
);  
CREATE TABLE white_list (  
  id VARCHAR,  
  name VARCHAR,  
  PRIMARY KEY (id), -- The Row Key field in a KVStore for Redis database.  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) WITH (  
  type = 'redis',  
  host = '<yourRedisHost>',  
  password = '<yourRedisPassword>'  
);  
SELECT e.*, w.*  
FROM event AS e  
JOIN white_list FOR SYSTEM_TIME AS OF PROCTIME() AS w  
ON e.id = w.id;
```

3.1.4.5.4.5. Create an ApsaraDB RDS dimension table

This topic describes how to create an ApsaraDB RDS dimension table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH and CACHE clauses and data type mapping used when you create such a dimension table.

 **Notice** Realtime Compute for Apache Flink cannot use ApsaraDB RDS for SQL Server V8.0 as a data store.

Introduction to ApsaraDB RDS

Alibaba Cloud ApsaraDB RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage services of Alibaba Cloud, ApsaraDB RDS supports a variety of engines such as MySQL, SQL Server, PostgreSQL, and PostgreSQL Plus Advanced Server (PPAS). ApsaraDB RDS provides comprehensive solutions for database operations and management, such as disaster recovery, data backup, data recovery, monitoring, and data migration.

Note You can use the same parameters in the WITH clause for ApsaraDB RDS, Distributed Relational Database Service (DRDS), and PolarDB. When you use an ApsaraDB RDS, DRDS, or PolarDB table as a dimension table, make sure that a real table exists.

Syntax

The following sample code demonstrates how to create an ApsaraDB RDS or DRDS dimension table. Only ApsaraDB RDS for MySQL is supported.

```
CREATE TABLE rds_dim_table(  
  id INT,  
  len INT,  
  content VARCHAR,  
  PRIMARY KEY (id),  
  PERIOD FOR SYSTEM_TIME--The identifier of a dimension table.  
) with (  
  type='rds',  
  url='<yourDatabaseURL>',  
  tableName='<yourDatabaseTableName>',  
  userName='<yourDatabaseUserName>',  
  password='<yourDatabasePassword>'  
);
```

Note When you declare a dimension table, you must specify the primary key. When you join a dimension table with another table, the ON condition must contain equivalent conditions that include all primary keys. The primary key in ApsaraDB RDS or DRDS is the primary key or unique index column of an ApsaraDB RDS or DRDS dimension table.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the dimension table.	Yes	Set the value to <code>rds</code> .
url	The Java Database Connectivity (JDBC) URL of a database.	Yes	Contact Alibaba Cloud O&M engineers.
tableName	The name of a table.	Yes	None.

Parameter	Description	Required	Remarks
userName	The username of the database user.	Yes	None.
password	The password of the database user.	Yes	None.
maxRetryTimes	The maximum number of connection retries.	No	Default value: 10.

Parameters in the CACHE clause

Parameter	Description	Required	Remarks
cache	The policy for caching data.	No	ApsaraDB RDS or DRDS dimension tables support the following cache policies: <i>None</i> (default value): indicates that no data is cached. <i>LRL</i>: indicates that partial data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system does not find the record in the cache, the system searches for the data record in the physical dimension table. If you use this cache policy, you must configure the <code>cacheSize</code> and <code>cacheTTLs</code> parameters. <i>ALL</i>: indicates that all data in the dimension table is cached. Before a Realtime Compute for Apache Flink job starts to run, Realtime Compute for Apache Flink loads all data in the dimension table to the cache, and then searches the cache for all subsequent queries in the dimension table. If the system does not find the data record in the cache, the join key does not exist. The system reloads all data in the cache after cache entries expire. If the amount of data of a remote table is small and a large number of missing keys exist, we recommend that you set this parameter to ALL. (The source table and dimension table cannot be associated based on the ON clause.) If you use this cache policy, you must configure the <code>cacheTTLs</code> and <code>cacheReloadTimeBlackList</code> parameters.  Note If you set the cache parameter to ALL, you must increase the memory of the node for joining tables because Realtime Compute for Apache Flink asynchronously loads data from the dimension table. The increased memory size is twice that of the remote table.

Parameter	Description	Required	Remarks
cacheSize	The maximum number of data records that can be cached. Unit: lines.	No	You can set this parameter when cache is set to LRU . Default value: 10000.
cacheTTLms	The cache timeout period. Unit: milliseconds.	No	If you set the cache parameter to LRU , the cacheTTLms parameter specifies the time before cache entries expire. Cache entries do not expire by default. If you set the cache parameter to ALL , the cacheTTLms parameter specifies the cache refresh interval. The cache is not refreshed by default.
cacheReloadTimeBlackList	The time periods during which the cache is not refreshed. You can set this parameter when cache is set to ALL . The cache is not refreshed during the time periods that you specify for this parameter. This parameter is useful for massive online promotional events such as Double 11.	No	This parameter is empty by default. Example: 2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00 . Separate multiple time periods with commas (,). Separate the start time and end time of each time period with a hyphen and a greater-than sign (->).

Sample code

The following sample code demonstrates how to create an ApsaraDB RDS dimension table in a Realtime Compute for Apache Flink job.

```
CREATE TABLE datahub_input1 (  
  id    BIGINT,  
  name  VARCHAR,  
  age   BIGINT  
) WITH (  
  type='datahub'  
);  
create table phoneNumber(  
  name VARCHAR,  
  phoneNumber BIGINT,  
  primary key(name),  
  PERIOD FOR SYSTEM_TIME-- The identifier of a dimension table.  
)with(  
  type='rds'  
);  
CREATE table result_infor(  
  id BIGINT,  
  phoneNumber BIGINT,  
  name VARCHAR  
)with(  
  type='rds'  
);  
INSERT INTO result_infor  
SELECT  
  t.id,  
  w.phoneNumber,  
  t.name  
FROM datahub_input1 as t  
JOIN phoneNumber FOR SYSTEM_TIME AS OF PROCTIME() as w -- You must include this clause in the INSERT  
INTO statement when you join a dimension table with another table.  
ON t.name = w.name;
```

For more information about the detailed syntax of a dimension table, see [JOIN statements for dimension tables](#).

Field type mapping

Data type of ApsaraDB RDS	Data type of Realtime Compute for Apache Flink
BOOLEAN	BOOLEAN
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DECIMAL	DECIMAL

Data type of ApsaraDB RDS	Data type of Realtime Compute for Apache Flink
DOUBLE	DOUBLE
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
VARBINARY	VARBINARY

3.1.4.5.4.6. Create a MaxCompute dimension table

This topic describes how to create a MaxCompute dimension table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH and CACHE clauses and data type mapping used when you create a MaxCompute dimension table.



Notice

- Blink 2.1.1 and later versions support MaxCompute dimension tables.
- For more information about the query syntax of a dimension table, see [JOIN statements for dimension tables](#).
- To use a MaxCompute dimension table, you must grant the read permissions to the account used to access MaxCompute.

DDL syntax

```
CREATE TABLE white_list (  
  id varchar,  
  name varchar,  
  age int,  
  PRIMARY KEY (id),  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) WITH (  
  type = 'odps',  
  endPoint = '<YourEndPoint>',  
  project = '<YourProjectName>',  
  tableName = '<YourtableName>',  
  accessId = '<yourAccessKeyId>',  
  accessKey = '<yourAccessKeySecret>',  
  `partition` = 'ds=2018****',  
  cache = 'ALL'  
);
```

 Note

- When you declare a dimension table, you must specify a primary key. The primary key of a MaxCompute dimension table must be unique. Duplicate primary keys are deleted.
- When you join a dimension table with another table, the ON condition must contain equivalence conditions that include all primary keys.
- partition is a keyword and must be commented with backticks (`), for example, `'partition'`.
- If the dimension table is a partitioned table, Realtime Compute for Apache Flink does not write partition key columns to the DDL statement.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the dimension table.	Yes	Set the value to <code>odps</code> .
endPoint	The endpoint of MaxCompute.	Yes	Contact Alibaba Cloud O&M engineers.
tunnelEndPoint	The endpoint of the MaxCompute Tunnel service.	Yes	Contact Alibaba Cloud O&M engineers.
project	The name of a MaxCompute project.	Yes	None.
tableName	The name of the table.	Yes	None.
accessId	AccessKey ID	Yes	None.
accessKey	AccessKey Secret	Yes	None.

Parameter	Description	Required	Remarks
partition	The name of a partition.	No	- Fixed partition - A MaxCompute table with only one partition For example, if only one partition key column <code>ds</code> exists, <code>`partition`='ds=20180905'</code> indicates that the data in the <code>ds=20180905</code> partition is read. - A MaxCompute table with multiple partitions For example, if two partition key columns <code>ds</code> and <code>hh</code> exist, <code>`partition`='ds=20180905.hh=*</code> indicates that the data in the <code>ds=20180905</code> partition is read.  Note You must declare the values of all partitions when you filter partitions. In the preceding example, if you declare only <code>`partition`='ds=20180905'</code>, no partition is read. - Non-fixed partition - Blink 2.2.0 and later versions support <code>'partition'='max_pt()'</code>. This setting indicates that the partition ranked first in alphabetical order among all partitions is loaded each time the system loads data of partitions. - Blink 3.2.2 and later versions support <code>'partition'='max_pt_with_done()'</code>. This setting indicates that the partition with the file name extension <code>.done</code> and ranked first in alphabetical order among all partitions is loaded each time the system loads data of partitions.
maxRowCount	The maximum number of rows that Realtime Compute for Apache Flink can load from a table.	No	Default value: 100000.  Note If your data contains more than 100,000 rows, you must set this parameter. We recommend that you set this parameter to a greater value than the actual number of rows to be loaded.

Parameters in the CACHE clause

Parameter	Description	Remarks

Parameter	Description	Remarks
cache	The cache policy.	You must set the cache parameter to ALL for a MaxCompute dimension table and explicitly declare this setting in the DDL statement. <i>ALL</i>: indicates that all data in the dimension table is cached. Before a Realtime Compute for Apache Flink job runs, Realtime Compute for Apache Flink loads all the data in the dimension table to the cache. Then, Realtime Compute for Apache Flink searches among the cache if requests are sent to retrieve data from the dimension table. If the system does not find the data record in the cache, the join key does not exist. The system reloads all data in the cache after cache entries expire. If the data amount of a remote table is small and a large number of missing keys exist, we recommend that you set this parameter to ALL. (The source table and dimension table cannot be associated based on the ON clause.) If you set this parameter to ALL, you must specify the following parameters: cacheTTLms and cacheReloadTimeBlackList.  Note - If the cache parameter is set to ALL, you must increase the memory of the join node because the system asynchronously loads data of the dimension table. We recommend that you increase the size of the memory at least four times the data amount of the remote table. The value is related to the MaxCompute storage compression algorithm. - If a job exception occurs due to frequent garbage collection when you use a super-large MaxCompute dimension table, and this issue persists even if you increase the memory of the join node, we recommend that: - For Blink 3.6.0 and later versions, set the partitionedJoin parameter to true to enable partitionedJoin optimization. - Use a dimension table with key-value pairs for which you can set the cache parameter to LRU, for example, an ApsaraDB for HBase dimension table.
cacheSize	The maximum number of data records that can be cached.	You can specify the cacheSize parameter based on your business requirements. By default, Realtime Compute for Apache Flink can cache 100,000 rows that are stored in a MaxCompute dimension table.

Parameter	Description	Remarks
cacheTTLms	The cache timeout period.	Unit: milliseconds. If you set the cache parameter to ALL , the timeout period specifies the interval at which Realtime Compute for Apache Flink refreshes the cache. The cache is not refreshed by default.
cacheReloadTimeBlackList	The time periods during which the cache is not refreshed. This parameter takes effect when the cache parameter is set to ALL. The cache is not refreshed during the time periods that you specify for this parameter. This parameter is useful for large-scale online promotional events such as Double 11.	This parameter is empty by default. The following example shows the format of the values: 2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00. The following list describes the delimiters that you use for the parameter values: • Separate multiple time periods with commas (,). • Separate the start time and end time of each time period with a hyphen and a greater-than sign (->).
partitionedJoin	Specifies whether to cache full data of a dimension table in the memory of each concurrent job.	This parameter is optional. Default value: false. This value indicates that full data of the dimension table is cached in the memory of each concurrent job. If you set this parameter to true, partial data of the dimension table is cached in the memory of each concurrent task.

Sample code

The following sample code demonstrates how to create a MaxCompute dimension table in a Realtime Compute for Apache Flink job.

```
CREATE TABLE datahub_input1 (
  id BIGINT,
  name VARCHAR,
  age BIGINT
) with (
  type='datahub'
);
CREATE TABLE odps_dim (
  name VARCHAR,
  phoneNumber BIGINT,
  PRIMARY KEY (name),
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.
) with (
  type = 'odps',
  endPoint = '<yourEndpointName>',
  project = '<yourProjectName>',
  tableName = '<yourTableName>',
  accessId = '<yourAccessId>',
  accessKey = '<yourAccessPassword>',
  `partition` = 'ds=20180905',--The fixed partition. For more information about dynamic partitions and fixed
partitions, see the description of parameters in the WITH clause.
  cache = 'ALL'
);
CREATE table result_infor(
  id BIGINT,
  phoneNumber BIGINT,
  name VARCHAR
)with(
  type='print'
);
INSERT INTO result_infor
SELECT
  t.id,
  w.phoneNumber,
  t.name
FROM datahub_input1 as t
JOIN odps_dim FOR SYSTEM_TIME AS OF PROCTIME() as w --You must include this clause when you perform
a JOIN operation on a dimension table.
ON t.name = w.name;
```

Field type mapping

The following table lists the mapping between the data types of MaxCompute and Realtime Compute for Apache Flink (fully-managed Flink). We recommend that you declare the mapping in a DDL statement.

MaxCompute	BLINK
TINYINT	TINYINT
SMALLINT	SMALLINT

MaxCompute	BLINK
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
DECIMAL	DECIMAL
BINARY	VARBINARY
STRING	VARCHAR

 **Note** Realtime Compute for Apache Flink supports only the preceding data types in MaxCompute dimension tables.

3.1.4.5.4.7. Create an AnalyticDB for MySQL V3.0 dimension table

This topic describes how to create an AnalyticDB for MySQL V3.0 dimension table. It also describes parameters in the WITH and CACHE clauses used when you create such a dimension table.

 **Notice** This topic applies only to Blink-3.5.0-hotfix and later.

Syntax

```
CREATE TABLE dim_ads(
  `name` VARCHAR,
  id VARCHAR,
  PRIMARY KEY (`name`),
  PERIOD FOR SYSTEM_TIME
)with(
  type='ADB30',
  url='jdbc:mysql://<Internal network address>/<databaseName>',
  tableName='xxx',
  userName='xxx',
  password='xxx'
);
```

Note

- You must specify a primary key when you declare a dimension table.
- When you join a dimension table with another dimension table, the ON condition must contain equivalent conditions that include all primary keys.
- In AnalyticDB for MySQL, you can use the primary key or unique index of a dimension table as the primary key.

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the dimension table.	Yes	Set the value to ADB30.
url	The URL of the AnalyticDB for MySQL database.	Yes	Contact Alibaba Cloud O&M engineers.
tableName	The name of the table.	Yes	None.
userName	The username of the database user.	Yes	None.
password	The password of the database user.	Yes	None.
maxRetryTimes	The maximum number of retries for writing data to the table.	No	Default value: 3.

Parameters in the CACHE clause

Parameter	Description	Remarks
		AnalyticDB for MySQL V3.0 supports the following cache policies: • <i>None</i> (default value): indicates that no data is cached. • <i>LRU</i>: indicates that partial data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system does not find the record in the cache, it searches for the data record in the physical dimension table.

Parameter	Description	Remarks
cache	The cache policy.	If you use this cache policy, you must configure the <code>cacheSize</code> and <code>cacheTTLs</code> parameters. <code>ALL</code>: indicates that all data in the dimension table is cached. Before a Realtime Compute for Apache Flink job starts to run, the system loads all data in the dimension table to the cache, and then searches the cache for all subsequent queries in the dimension table. If the system does not find the data record in the cache, the join key does not exist. The system reloads all data in the cache after cache entries expire. If the amount of data of a remote table is small and a large number of missing keys exist, we recommend that you set this parameter to <code>ALL</code>. (The source table and dimension table cannot be associated based on the <code>ON</code> clause.) If you use this cache policy, you must configure the <code>cacheTTLs</code> and <code>cacheReloadTimeBlackList</code> parameters.  Note - If you set the cache parameter to <code>ALL</code>, you must increase the memory of the node for joining tables because Realtime Compute for Apache Flink asynchronously loads data from the dimension table. The increased memory size is twice the memory size of the remote table. - If the amount of data in a dimension table is large and the cache parameter is set to <code>ALL</code>, an out of memory (OOM) error may occur or a full garbage collection may be time-consuming. To address this issue, you can use one of the following methods: - If the cache parameter can be set to <code>ALL</code> for a dimension table, enable the <code>partitionedJoin</code> function. If you use a Blink version earlier than 3.6.0, full data of the dimension table is loaded for each concurrent job by default. If you use a Blink version later than 3.6.0, you can enable the <code>partitionedJoin</code> function when the cache parameter is set to <code>ALL</code>. After you enable the <code>partitionedJoin</code> function, only the required data is cached for each concurrent job. - Use an HBase or ApsaraDB RDS dimension table that uses key-value pairs to store data.

Parameter	Description	Remarks
cacheSize	The cache size. Unit: lines.	You can set this parameter when the cache parameter is set to <i>LRU</i> . Default value: 10000.
cacheTTLms	The cache refresh interval. The system reloads the latest data in the dimension table based on the value of this parameter to ensure that the source table can be joined with the latest data of the dimension table.	Unit: milliseconds. This parameter is empty by default, which indicates that the latest data in the dimension table is not reloaded.
cacheReloadTimeBlackList	The time periods during which cache is not refreshed. This parameter takes effect when the cache parameter is set to ALL. The cache is not refreshed during the time periods that you specify for this parameter. This parameter is useful for massive online promotional events such as Double 11.	This parameter is optional. It is empty by default. Example: '2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00'. Make sure that the parameter conforms to the following rules: • Separate multiple time periods with commas (,). • Separate the start time and end time of each time period with a hyphen and a greater-than sign (->).

Parameter	Description	Remarks
partitionedJoin	Specifies whether to enable partitionedJoin. If you enable the partitionedJoin function, the shuffle operation is performed before the source table is associated with the dimension table based on join keys. - If you set the cache parameter to <i>LRU</i>, the cache hit rate increases. - If you set the cache parameter to <i>ALL</i>, you can save memory resources because only the required data is cached for each concurrent job.	This parameter is set to false by default, which indicates that the partitionedJoin function is disabled.  Note Before you enable the partitionedJoin function, set partitionedJoin to true.

Sample code

```
CREATE TABLE datahub_input1 (  
  id BIGINT,  
  name VARCHAR,  
  age BIGINT  
) WITH (  
  type='datahub'  
);  
create table phoneNumber (  
  name VARCHAR,  
  phoneNumber BIGINT,  
  primary key(name),  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) with (  
  type='ADB30'  
);  
CREATE table result_infor (  
  id BIGINT,  
  phoneNumber BIGINT,  
  name VARCHAR  
) with (  
  type='rds'  
);  
INSERT INTO result_infor  
SELECT  
  t.id,  
  w.phoneNumber,  
  t.name  
FROM datahub_input1 as t  
JOIN phoneNumber FOR SYSTEM_TIME AS OF PROCTIME() as w -- You must include this clause in the INSERT  
INTO statement when you join a dimension table with another table.  
ON t.name = w.name;
```

3.1.4.5.4.8. Create a Hologres dimension table

This topic describes how to create a Hologres dimension table. It also describes the parameters in the WITH clause, parameters in the CACHE clause, and data type mapping used when you create such a dimension table.

Notice

- This topic applies only to Blink 3.6.0 and later.
- We recommend that you use Hologres 0.7 or later.

Introduction to Hologres

Hologres is compatible with the PostgreSQL protocol and closely connected to the big data ecosystem. Hologres supports real-time analysis and processing of petabytes of data with high concurrency and low latency. This allows you to use existing Business Intelligence (BI) tools to easily perform multidimensional analysis and business exploration.

Syntax

In Realtime Compute for Apache Flink, you can use a Hologres table as a dimension table. The following code shows an example:

```
CREATE TABLE rds_dim_table(  
  id INT,  
  len INT,  
  content VARCHAR,  
  PRIMARY KEY (id),  
  PERIOD FOR SYSTEM_TIME -- The identifier of a dimension table.  
) with (  
  type='hologres',  
  endpoint='...',  
  dbname='...',  
  tablename='...',  
  username='...',  
  password='...'  
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the database.	Yes	Set the value to <i>hologres</i> .
endpoint	The endpoint of Hologres.	Yes	None.
tablename	The table from which data is read.	Yes	None.
dbname	The database from which data is read.	Yes	None.
username	The username that is used to log on to the database.	Yes	None.
password	The password that is used to log on to the database.	Yes	None.

Parameters in the CACHE clause

Parameter	Description	Required	Example
			Hologres dimension tables support the following cache policies: <i>None</i> (default value): indicates that no data is cached. <i>LRU</i>: indicates that partial data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system

Parameter	Description	Required	Example
cache	The policy for caching data.	No	does not find the record in the cache, it searches for the data record in the physical dimension table. If this cache policy is used, you must configure the cacheSize and cacheTTLs parameters. • <i>ALL</i>: indicates that all data in the dimension table is cached. Before a Realtime Compute for Apache Flink job starts to run, Realtime Compute for Apache Flink loads all data in the dimension table to the cache, and then searches the cache for all subsequent queries in the dimension table. If the system does not find the data record in the cache, the join key does not exist. The system reloads all data in the cache after cache entries expire. If the amount of data of a remote table is small and a large number of missing keys exist, we recommend that you set this parameter to <i>ALL</i>. (The source table and dimension table cannot be associated based on the ON clause.) If this cache policy is used, you must configure the cacheTTLs and cacheReloadTimeBlackList parameters.

Parameter	Description	Required	Example	Note
				- If the cache parameter is set to ALL, the memory of the node for joining tables must be increased because Realtime Compute for Apache Flink asynchronously loads data from the dimension table. The increased memory size is two times that of the remote table. - If the amount of data in a dimension table is large and the cache parameter is set to ALL, an out of memory (OOM) error may occur or a full garbage collection may be time-consuming. To address this issue, you can use one of the following methods: - If the cache parameter can be set to ALL for a dimension table, enable the partitionedJoin function. For Blink versions earlier than 3.6.0, full data of the dimension table is loaded in each concurrent job by default. For Blink versions later than 3.6.0, the partitionedJoin function is supported when you set the cache parameter to ALL. After you enable the partitionedJoin function, only the required data is cached for each concurrent job.
cacheSize	The maximum number of data records that can be cached. Unit: lines.	No		You can specify this parameter only when you set the cache parameter to <i>LRU</i>. Default value: <i>10000</i>. Use an HBase or ApsaraDB RDS dimension table that uses key-value pairs to store data.
cacheTTLms	The cache timeout period. Unit: milliseconds.	No		- You can specify this parameter only when you set the cache parameter to <i>LRU</i>. The cache will not expire by default. - If the cache parameter is set to <i>ALL</i>, the cacheTTLms parameter specifies the interval at which the cache is reloaded. The cache is not reloaded by default.
partitionedJoin	Specifies whether to search for data by partition.	No		Valid values: <i>false</i> (default value) <i>true</i>
async	Specifies whether to read data asynchronously.	No		Valid values: <i>false</i> (default value) <i>true</i>

Field type mapping

Hologres data type	Data type of Realtime Compute for Apache Flink
INT	INT
INT []	ARRAY<INT>
BIGINT	BIGINT
BIGINT []	ARRAY<BIGINT>
REAL	FLOAT
REAL []	ARRAY<FLOAT>
DOUBLE PRECISION	DOUBLE
DOUBLE PRECISION []	ARRAY<DOUBLE>
BOOLEAN	BOOLEAN
BOOLEAN []	ARRAY<BOOLEAN>
TEXT	VARCHAR
TEXT []	ARRAY<VARCHAR>
NUMERIC	DECIMAL
DATE	DATE
TIMESTAMP WITH TIMEZONE	TIMESTAMP

Sample code

The following sample code demonstrates how to create a Hologres dimension table in a Realtime Compute for Apache Flink job:


```
create table randomSource (a int, b VARCHAR, c VARCHAR) with (type = 'random');
create table test (
  a int,
  b VARCHAR,
  c VARCHAR,
  PRIMARY KEY (a, b), PERIOD FOR SYSTEM_TIME
) with (
  type = 'hologres',
  ...
);
create table print_sink (
  a int,
  b VARCHAR
) with (
  type = 'print',
  `ignoreWrite` = 'false'
);
insert into print_sink
select randomSource.a, test.b from randomSource
LEFT JOIN test FOR SYSTEM_TIME AS OF PROCTIME()
on randomSource.a = test.a and randomSource.b = test.b;
```

3.1.4.5.4.9. Create a Phoenix5 dimension table

This topic describes how to create a Phoenix5 dimension table in Realtime Compute for Apache Flink. It also describes the parameters in the WITH and CACHE clauses used when you create such a dimension table.

 **Notice** Only Blink versions later than Blink 3.4.0 support Phoenix dimension tables.

Syntax

```
create table US_POPULATION_DIM (
  `STATE` varchar,
  CITY varchar,
  POPULATION BIGINT,
  PRIMARY KEY (`STATE`, CITY),
  PERIOD FOR SYSTEM_TIME
) WITH (
  type = 'PHOENIX5',
  serverUrl = '<YourServerUrl>',
  tableName = '<YourTableName>'
);
```

Parameters in the WITH clause

Parameter	Description	Required	Remarks
type	The type of the dimension table.	Yes	Set the value to PHOENIX5 .

Parameter	Description	Required	Remarks
serverUrl	The address of the Phoenix5 Query Server. • If Phoenix5 is created in a cluster, the URL of Server Load Balancer (SLB) is used. • If Phoenix5 is created on a server, the URL of the server is used.	Yes	You must enable the HBase SQL service in an ApsaraDB for HBase instance.
tableName	The name of the Phoenix5 table.	Yes	None.

Parameters in the CACHE clause

Parameter	Description	Required	Remarks

Parameter	Description	Required	Remarks
cache	The policy for caching data.	No	Valid values: <i>None</i> (default value): indicates that no data is cached. <i>LRU</i>: indicates that partial data in the dimension table is cached. The system searches the cache each time it receives a data record. If the system does not find the record in the cache, it searches for the data record in the physical dimension table. If this cache policy is used, you must configure the <code>cacheSize</code> and <code>cacheTTLs</code> parameters. <i>ALL</i>: indicates that all data in the dimension table is cached. Before a Realtime Compute for Apache Flink job starts to run, the system loads all data in the dimension table to the cache, and then searches the cache for all subsequent queries in the dimension table. If the system does not find the data record in the cache, the join key does not exist. The system reloads all data in the cache after cache entries expire. If the amount of data of a remote table is small and a large number of missing keys exist, we recommend that you set this parameter to <i>ALL</i>. (The source table and dimension table cannot be associated based on the <code>ON</code> clause.) If this cache policy is used, you must configure the <code>cacheTTLs</code> and <code>cacheReloadTimeBlackList</code> parameters.  Note If the cache parameter is set to <i>ALL</i>, the memory of the node for joining tables must be increased because Realtime Compute for Apache Flink asynchronously loads data from the dimension table. The increased memory size is two times that of the remote table.
cacheSize	The maximum number of data records that can be cached. Unit: lines.	No	If you set the cache parameter to <i>LRU</i> , you can set this parameter to specify the maximum number of data records that can be cached. Default value: 10000.

Parameter	Description	Required	Remarks
cacheTTL Ms	The cache timeout period. Unit: milliseconds.	No	If the cache parameter is set to <i>LRU</i> , the cacheTTL parameter specifies the time allowed before cache entries expire. Cache entries do not expire by default. If the cache parameter is set to <i>ALL</i> , the cacheTTL parameter specifies the interval at which the cache is loaded. The cache is not reloaded by default.
cacheReloadTimeBlackList	The time periods during which the cache is not refreshed. This parameter takes effect when the cache parameter is set to <i>ALL</i> . The cache is not refreshed during the time periods that you specify for this parameter. This parameter is useful for massive online promotional events such as Double 11.	No	This parameter is optional. It is empty by default. Example: '2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00'. Separate multiple time periods with commas (,). Separate the start time and end time of each time period with a hyphen and a greater-than sign (->).

Sample code

```
CREATE TABLE datahub_input1 (  
  id BIGINT,  
  name VARCHAR,  
  age BIGINT  
) WITH (  
  type='datahub'  
);  
create table phoneNumber(  
  name VARCHAR,  
  phoneNumber BIGINT,  
  primary key(name),  
  PERIOD FOR SYSTEM_TIME-- The identifier of a dimension table.  
)with(  
  type='PHOENIX5'  
);  
CREATE table result_infor(  
  id BIGINT,  
  phoneNumber BIGINT,  
  name VARCHAR  
)with(  
  type='rds'  
);  
INSERT INTO result_infor  
SELECT  
  t.id,  
  w.phoneNumber,  
  t.name  
FROM datahub_input1 as t  
JOIN phoneNumber FOR SYSTEM_TIME AS OF PROCTIME() as w -- You must include this clause in the INSERT  
INTO statement when you join a dimension table with another table.  
ON t.name = w.name;
```

3.1.4.5.4.10. Create an Elasticsearch dimension table

This topic describes how to create an Elasticsearch dimension table in Realtime Compute for Apache Flink. This topic also describes the parameters in the WITH clause and CACHE clauses used when you create an Elasticsearch dimension table.

 **Notice** This topic applies only to Blink 3.2.2 and later.

DDL syntax

In Realtime Compute for Apache Flink, you can use an Elasticsearch table as a dimension table. The following code shows an example:

```
CREATE TABLE es_stream_sink(  
  field1 LONG,  
  field2 VARBINARY,  
  field3 VARCHAR,  
  PRIMARY KEY(field1),  
  PERIOD FOR SYSTEM_TIME  
) WITH (  
  type = 'elasticsearch',  
  endPoint = '<yourEndPoint>',  
  accessId = '<yourUsername>',  
  accessKey = '<yourPassword>',  
  index = '<yourIndex>',  
  typeName = '<yourTypeName>'  
);
```

 **Note** An Elasticsearch dimension table supports data updates based on the primary key of an Elasticsearch cluster. You can specify only one field for the primary key field.

Parameters in the WITH clause

Parameter	Description	Default value	Required
type	The type of the dimension table.	elasticsearch	Yes
endPoint	The endpoint of the Elasticsearch cluster, for example, <i>http://127.0.0.1:9211</i> .	No default value	Yes
accessId	The AccessKey ID that is used to access the Elasticsearch cluster.	No default value	Yes
accessKey	The AccessKey secret that is used to access the Elasticsearch cluster.	No default value	Yes
index	The index name, which is similar to the database name.	No default value	Yes
typeName	The type name, which is similar to the database table name.	No default value	Yes
maxRetryTimes	The maximum number of retries for writing data to the table.	30	No
timeout	The read timeout period. Unit: milliseconds.	600000	No
discovery	Specifies whether node discovery is enabled. If this feature is enabled, the client refreshes the server list every 5 minutes.	false	No

Parameter	Description	Default value	Required
compression	Specifies whether to compress request bodies in the GZIP format.	true	No
multiThread	Specifies whether to enable multithreading for JestClient.	true	No

Parameters in the CACHE clause

Parameter	Description	Remarks
cache	The cache policy.	<i>None</i>: indicates that no data is cached. This value is the default value. <i>LRU</i>: indicates that only the specified data in the dimension table is cached. The system searches the cache each time it receives a data record from the source table. If the system does not find the record in the cache, the system searches for the data record in the physical dimension table. If you use this cache policy, you must configure the cacheSize and cacheTTLms parameters. <i>ALL</i>: indicates that all the data in the dimension table is cached. Before Realtime Compute for Apache Flink runs a job, Realtime Compute for Apache Flink loads all the data in the dimension table to the cache and then searches the cache for all subsequent queries in the dimension table. If the system does not find the data record in the cache, the key does not exist. The system reloads all data in the cache after cache entries expire.
cacheSize	The cache size.	You can specify this parameter only if you set the cache parameter to LRU . Default value: 10000.
cacheTTLms	The interval at which the cache is refreshed.	The cache does not time out by default. The purpose of setting this parameter varies based on the cache policy. - If the cache parameter is set to LRU, the cacheTTLms parameter specifies the timeout period of the cache. - If the cache parameter is set to ALL, the cacheTTLms parameter specifies the interval at which the cache is refreshed. The cache is not refreshed by default.

3.1.4.6. DML statements

3.1.4.6.1. INSERT INTO statement

This topic describes how to use the INSERT INTO statement in Realtime Compute.

Syntax

```
INSERT INTO tableName  
  [ (columnName[, columnName]*) ]  
  queryStatement;
```

Example

```
INSERT INTO LargeOrders  
SELECT * FROM Orders WHERE units > 1000;  
INSERT INTO Orders(z, v)  
SELECT c,d FROM OO;
```

- A Realtime Compute job created from an SQL file can include multiple DML operations. It also supports multiple data stores for input and output data, and supports multiple dimension tables. For example, a job can contain two paragraphs of SQL statements written for different businesses. You can use SQL statements to write output data into different data stores.
- In Realtime Compute, you cannot use separate SELECT statements. SELECT statements must be used after CREATE VIEW or INSERT INTO statements.
- You can update a table by using the INSERT INTO statement. For example, you can use an INSERT INTO statement to write a data record with a key value into an RDS table. If the key value is found in the table, the existing data record in the table is overwritten. If the key value is not found, the data record is added to the table.

Table types and statements

Table type	Statement
Source table	You can use the FROM clause. The INSERT INTO statement is not supported.
Dimension table	You can use the JOIN statement. The INSERT INTO statement is not supported.
Result table	You can only use the INSERT INTO statement.
View	You can only use the FROM clause.

3.1.4.7. Query statements

3.1.4.7.1. SELECT statements

You can execute SELECT statements to retrieve data from tables.

Syntax


```
SELECT [ DISTINCT ]  
{ * | projectItem [, projectItem ]* }  
FROM tableExpression;
```

Test data

a (VARCHAR)	b (INT)	c (DATE)
a1	211	1990-02-20
b1	120	2018-05-12
c1	89	2010-06-14
a1	46	2016-04-05

Simple queries

- Test statement

```
SELECT * FROM <Table name>;
```

- Test result

a (VARCHAR)	b (INT)	c (DATE)
a1	211	1990-02-20
b1	120	2018-05-12
c1	89	2010-06-14
a1	46	2016-04-05

Rename objects

- Test statement

```
SELECT a, c AS d FROM <Table name>;
```

- Test result

a (VARCHAR)	d (DATE)
a1	1990-02-20
b1	2018-05-12
c1	2010-06-14
a1	2016-04-05

Deduplication queries

- Test statement

```
SELECT DISTINCT a FROM Table name;
```

- Test result

a (VARCHAR)
a1
b1
c1

Subqueries

In most cases, SELECT statements read data from tables, for example, `SELECT column_1, column_2 ... FROM table_name`. SELECT statements can also read data from the results of other SELECT statements. This is known as subqueries.

 **Note** You must specify aliases in subqueries.

- Test statement

```
INSERT INTO result_table
SELECT * FROM
    (SELECT t.a,
        sum(t.b) AS sum_b
    FROM t1 t
    GROUP BY t.a
    ) t1
WHERE t1.sum_b > 100;
```

- Test result

a (VARCHAR)	b (INT)
a1	211
b1	120
a1	257

 **Note** The preceding test result is a debugging result. In the result, you can view the computing process. If your job is published and the result table is stored in DataHub, Alibaba Cloud Message Queue for Apache Kafka, or Alibaba Cloud Message Queue, the computing process is displayed. If your job is published and the result table is stored in a relational database such as ApsaraDB RDS, the records that have the same primary key values are combined into one record.

3.1.4.7.2. WHERE

A WHERE clause filters data returned by a SELECT statement.

Syntax

```
SELECT [ ALL | DISTINCT ]  
{ * | projectItem [, projectItem ]* }  
FROM tableExpression  
[ WHERE booleanExpression ];
```

The following table describes the operators that can be used in a WHERE clause.

Operator	Description
=	Equal to
<>	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Example

- Test data

Address	City
Oxford Street	Beijing
Fifth Avenue	Beijing
Changan Street	Shanghai

- Test statements

```
SELECT * FROM XXXX WHERE City='Beijing';
```

- Test results

Address	City
Oxford Street	Beijing
Fifth Avenue	Beijing

3.1.4.7.3. HAVING

When you use an aggregate function, you must add a HAVING clause to achieve the same filter effect as a WHERE clause.

Syntax

```
SELECT [ ALL | DISTINCT ] { * | projectItem [, projectItem ]* }  
FROM tableExpression  
[ WHERE booleanExpression ]  
[ GROUP BY { groupItem [, groupItem ]* } ]  
[ HAVING booleanExpression ];
```

Example

- Test data

Customer	OrderPrice
Bush	1000
Carter	1600
Bush	700
Bush	300
Adams	2000
Carter	100

- Test statements

```
SELECT Customer,SUM(OrderPrice) FROM XXX  
GROUP BY Customer  
HAVING SUM(OrderPrice)<2000;
```

- Test results

Customer	SUM (OrderPrice)
Carter	1700

3.1.4.7.4. GROUP BY

A GROUP BY clause groups a result set by one or more columns.

Syntax

```
SELECT [ DISTINCT ]  
{ * | projectItem [, projectItem ]* }  
FROM tableExpression  
[ GROUP BY { groupItem [, groupItem ]* } ];
```

Example

- Test data (Table T1)

Customer	OrderPrice
Bush	1000
Carter	1600
Bush	700
Bush	300
Adams	2000
Carter	100

- Test statements

```
SELECT Customer,SUM(OrderPrice) FROM T1  
GROUP BY Customer;
```

- Test results

Customer	SUM(OrderPrice)
Bush	2000
Carter	1700
Adams	2000

3.1.4.7.5. JOIN statements

JOIN statements in Realtime Compute for Apache Flink have the same semantic meanings as those for batch processing. The two types of JOIN statements allow you to join two tables. The difference is that each JOIN statement in Realtime Compute for Apache Flink joins two dynamic tables. The join results are dynamically updated to ensure that the final results are the same as the corresponding results of batch processing.

Syntax

```
tableReference [, tableReference ]* | tableexpression  
[ LEFT ] JOIN tableexpression [ joinCondition ];
```

- tableReference: specifies the table name.
- tableexpression: specifies the expression.
- joinCondition: specifies the join condition.

**Notice**

- Only EQUI JOIN operations are supported.
- Only INNER JOIN and LEFT OUTER JOIN operations are supported.

Example 1: Join the Orders table and the Products table

- Test data Orders

rowtime	productId	orderId	units
10:17:00	30	5	4
10:17:05	10	6	1
10:18:05	20	7	2
10:18:07	30	8	20
11:02:00	10	9	6
11:04:00	10	10	1
11:09:30	40	11	12
11:24:11	10	12	4

Products

productId	name	unitPrice
30	Cheese	17
10	Beer	0.25
20	Wine	6
30	Cheese	17
10	Beer	0.25
10	Beer	0.25
40	Bread	100
10	Beer	0.25

- Test statement

```
SELECT o.rowtime, o.productId, o.orderId, o.units, p.name, p.unitPrice
FROM Orders AS o
JOIN Products AS p
ON o.productId = p.productId;
```

- Test result

o.rowtime	o.productId	o.orderId	o.units	p.name	p.unitPrice
10:17:00	30	5	4	Cheese	17
10:17:05	10	6	1	Beer	0.25
10:18:05	20	7	2	Wine	6
10:18:07	30	8	20	Cheese	17
11:02:00	10	9	6	Beer	0.25
11:04:00	10	10	1	Beer	0.25
11:09:30	40	11	12	Bread	100
11:24:11	10	12	4	Beer	0.25

Example 2: Join the datahub_stream1 table and the datahub_stream2 table

- Test data datahub_stream1

a (BIGINT)	b (BIGINT)	c (VARCHAR)
0	10	test11
1	10	test21

datahub_stream2

a (BIGINT)	b (BIGINT)	c (VARCHAR)
0	10	test11
1	10	test21
0	10	test31
1	10	test41

- Test statement

```
SELECT s1.c,s2.c
FROM datahub_stream1 AS s1
JOIN datahub_stream2 AS s2
ON s1.a=s2.a
WHERE s1.a = 0;
```

- Test result

s1.c (VARCHAR)	s2.c (VARCHAR)
test11	test11
test11	test31

3.1.4.7.6. JOIN statements for dimension tables

In Realtime Compute for Apache Flink, each data stream can be associated with an dimension table that is stored in an external data source. This allows you to perform associated queries in Realtime Compute for Apache Flink.

 **Note** A dimension table constantly changes. Therefore, when you associate a data stream with a dimension table, you must specify the time of the dimension table snapshot with which the data stream is associated. A data stream can be associated with only the dimension table snapshot that is taken at the current time. In the future, Realtime Compute for Apache Flink will allow you to associate data streams with dimension table snapshots that are taken at different points in time. The points in time are specified by the rowtime field in the left table.

Syntax

```
SELECT column-names
FROM table1 [AS <alias1>]
[LEFT] JOIN table2 FOR SYSTEM_TIME AS OF PROCTIME() [AS <alias2>]
ON table1.column-name1 = table2.key-name1;
```

In the following example, an event stream is joined with a whitelist dimension table.

```
SELECT e.*, w.*
FROM event AS e
JOIN white_list FOR SYSTEM_TIME AS OF PROCTIME() AS w
ON e.id = w.id;
```


 Note

- Dimension tables support `INNER JOIN` and `LEFT JOIN` operations, and do not support `RIGHT JOIN` or `FULL JOIN` operations.
- You must append `FOR SYSTEM_TIME AS OF PROCTIME()` to the end of the dimension table. This way, each data record in the dimension table that can be viewed at the current time is associated with the source data.
- The subsequent input data in the source table is associated with only the latest records that are stored in the dimension table at the current time. This means that the JOIN operation is performed only at the processing time. Therefore, if the data in the dimension table is added, updated, or deleted after the JOIN operation is performed, the associated data remains unchanged.
- The ON clause must contain the equivalent (=) conditions for all the primary key fields of the dimension table. The primary key fields in the conditions must be the same as those in the physical tables that are referenced in the SQL statement. The ON clause can also contain other equivalent (=) conditions.
- If you want to perform one-to-many table joins, you must specify the join keys in the data definition language (DDL) INDEX syntax for dimension tables.
- Two dimension tables cannot be joined.
- In the join conditions that are specified in the ON clause, the fields in the dimension table cannot use type conversion functions such as CAST. If you need to convert data types, perform the conversion on the fields in the source table.

Examples

- Test data datahub_input1

id (BIGINT)	name (VARCHAR)	age (BIGINT)
1	lilei	22
2	hanmeimei	20
3	libai	28

phoneNumber

name (VARCHAR)	phoneNumber (BIGINT)
dufu	18867889855
baijuyi	18867889856
libai	18867889857
lilei	18867889858

- Test statement

```
CREATE TABLE datahub_input1 (  
  id    BIGINT,  
  name  VARCHAR,  
  age   BIGINT  
) WITH (  
  type='datahub'  
);  
create table phoneNumber(  
  name VARCHAR,  
  phoneNumber bigint,  
  primary key(name),  
  PERIOD FOR SYSTEM_TIME  
)with(  
  type='rds'  
);  
CREATE table result_infor(  
  id bigint,  
  phoneNumber bigint,  
  name VARCHAR  
)with(  
  type='rds'  
);  
INSERT INTO result_infor  
SELECT  
  t.id,  
  w.phoneNumber,  
  t.name  
FROM datahub_input1 as t  
JOIN phoneNumber FOR SYSTEM_TIME AS OF PROCTIME() as w  
ON t.name = w.name;
```

- Test result

id (BIGINT)	phoneNumber (BIGINT)	name (VARCHAR)
1	18867889858	lilei
3	18867889857	libai

3.1.4.7.7. UNION ALL

A UNION ALL clause is used to combine two data streams. The field types and sequences of the two data streams must be the same.

Syntax

```
select_statement  
UNION ALL  
select_statement;
```

Note Realtime Compute for Apache Flink also supports the `UNION` function. `UNION ALL` allows duplicate values and `UNION` does not allow duplicate values. In Realtime Compute for Apache Flink, `UNION` is equivalent to the combination of `UNION ALL` and `DISTINCT`. We recommend that you do not use `UNION` because its operating efficiency is low.

Example

- Test data test_source_union1

a (VARCHAR)	b (BIGINT)	c (BIGINT)
test1	1	10

test_source_union2

a (VARCHAR)	b (BIGINT)	c (BIGINT)
test1	1	10
test2	2	20

test_source_union3

a (VARCHAR)	b (BIGINT)	c (BIGINT)
test1	1	10
test2	2	20
test1	1	10

- Sample code

```
SELECT
  a,
  sum(b) as d,
  sum(c) as e
FROM
  (SELECT * from test_source_union1
   UNION ALL
   SELECT * from test_source_union2
   UNION ALL
   SELECT * from test_source_union3
  )t
GROUP BY a;
```

- Test results

a (VARCHAR)	d (BIGINT)	e (BIGINT)
test1	1	10

a (VARCHAR)	d (BIGINT)	e (BIGINT)
test2	2	20
test1	2	20
test1	3	30
test2	4	40
test1	4	40

 **Note** The preceding test results are debugging results. In these results, you can view the computing process. If your job is published and the result table is stored in DataHub, Alibaba Cloud Message Queue for Apache Kafka, or Alibaba Cloud Message Queue, the result data contains data about the computing process. If your job is published and the result table is stored in a relational database such as ApsaraDB RDS, the records that have the same primary key values are combined into one record.

3.1.4.7.8. TopN

A TopN clause is used to compute the largest or smallest N data records of a metric in real-time data. Flink SQL uses an OVER window function to flexibly implement TopN computing.

Syntax

```
SELECT *
FROM (
  SELECT *,
    ROW_NUMBER() OVER ([PARTITION BY col1[, col2..]]
    ORDER BY col1 [asc|desc][, col2 [asc|desc]...]) AS rownum
  FROM table_name)
WHERE rownum <= N [AND conditions]
```

 Note

- **ROW_NUMBER()** : specifies an **OVER** window function to compute the row number. The value starts from 1.
- **PARTITION BY col1[, col2..]** : specifies one or more partitioning columns. This parameter is optional.
- **ORDER BY col1 [asc|desc][, col2 [asc|desc]...]** : specifies the columns based on which you want to sort data and the sorting order of each column.

As shown in the preceding syntax, a TopN clause requires two levels of queries.

- In the subquery, the **ROW_NUMBER()** window function is used to sort data based on the specified columns and mark the data with rankings.
- In the outer query, only the first N data records in the ranking list are obtained. For example, if N is set to 10, the first 10 data records are obtained.

During the execution, Flink SQL sorts an input data stream based on the sort key. If the first N data records in a partition are changed, the updated data is sent downstream as an update stream.

Note Therefore, if you want to export the TopN data to external storage, the target result table must contain a primary key.

WHERE

To enable Flink SQL to identify a TopN query, use `rownum <= N` in the outer query to specify the first N records. Do not place `rownum` in an expression, for example, `rownum - 5 <= N`. If you specify multiple conditions in the WHERE clause, you must use `AND` to join the conditions.

Example 1

You can use the following example to return the top 100 keywords that are queried the most in each city within a specific hour. The hour, city, and ranking columns in the output table identify a unique record. Therefore, you must declare the three columns as a composite key. The key must also be configured in the external storage.

```
CREATE TABLE rds_output (
  rownum BIGINT,
  start_time BIGINT,
  city VARCHAR,
  keyword VARCHAR,
  pv BIGINT,
  PRIMARY KEY (rownum, start_time, city)
) WITH (
  type = 'rds',
  ...
)
INSERT INTO rds_output
SELECT rownum, start_time, city, keyword, pv
FROM (
  SELECT *,
    ROW_NUMBER() OVER (PARTITION BY start_time, city ORDER BY pv desc) AS rownum
  FROM (
    SELECT SUBSTRING(time_str,1,12) AS start_time,
      keyword,
      count(1) AS pv,
      city
    FROM tmp_search
    GROUP BY SUBSTRING(time_str,1,12), keyword, city
  ) a
) t
WHERE rownum <= 100
```

Example 2

- Test data

ip (VARCHAR)	time (VARCHAR)
192.168.1.1	100,000,000

ip (VARCHAR)	time (VARCHAR)
192.168.1.2	100,000,000
192.168.1.2	100,000,000
192.168.1.3	100,030,000
192.168.1.3	100,000,000
192.168.1.3	100,000,000

- Test statements

```
CREATE TABLE source_table (
  IP VARCHAR,
  `TIME` VARCHAR
)WITH(
  type='datahub',
  endPoint='<yourEndpoint>',
  project='<yourProjectName>',
  topic='<yourTopicName>',
  accessId='<yourAccessId>',
  accessKey='<yourAccessSecret>'
);
CREATE TABLE result_table (
  rownum BIGINT,
  start_time VARCHAR,
  IP VARCHAR,
  cc BIGINT,
  PRIMARY KEY (start_time, IP)
) WITH (
  type = 'rds',
  url='<yourDatabaseAddress>',
  tableName='blink_rds_test',
  userName='<yourDatabaseUserName>',
  password='<yourDatabasePassword>'
);
INSERT INTO result_table
SELECT rownum,start_time,IP,cc
FROM (
  SELECT *,
    ROW_NUMBER() OVER (PARTITION BY start_time ORDER BY cc DESC) AS rownum
  FROM (
    SELECT SUBSTRING(`TIME`,1,2) AS start_time,-- You can specify a value based on the actual time. The
data specified here is an example.
    COUNT(IP) AS cc,
    IP
    FROM source_table
    GROUP BY SUBSTRING(`TIME`,1,2), IP
  )a
) t
WHERE rownum <= 3 -- You can specify a value based on the number of data records you want to obtain. T
he data specified here is an example.
```

● Test results

rownum (BIGINT)	start_time (VARCHAR)	ip (VARCHAR)	cc (BIGINT)
1	10	192.168.1.3	3
2	10	192.168.1.2	2
3	10	192.168.1.1	1

No ranking number optimization

- You can use no ranking number optimization to solve the data bloat issue.

- Data bloat

Based on the TopN syntax, the `rownum` field is written into a result table as one of its primary keys. This may lead to data bloat. For example, if the ranking of a data record rises from the ninth to the first after a data update, the rankings of the records from the first to the ninth all change. The changes must be updated in the result table. This results in data bloat. The data update in the result table may slow down because an excessive volume of data is received.

- Method of no ranking number optimization

To avoid data bloat, remove `rownum` from the result table and compute `rownum` at the front end. The volume of TopN data records is not large, so the top 100 data records can be obtained quickly at the front end. In this case, if the ranking of a data record rises from the ninth to the first after an update, only this record needs to be delivered to the result table. This accelerates data update in the result table.

- Syntax

```
SELECT col1, col2, col3
FROM (
  SELECT col1, col2, col3
  ROW_NUMBER() OVER ([PARTITION BY col1[, col2..]]
  ORDER BY col1 [asc|desc][, col2 [asc|desc]...]) AS rownum
FROM table_name)
WHERE rownum <= N [AND conditions]
```

The syntax is similar to the original TopN syntax. You only need to remove the `rownum` field from the outer query.

 **Note** If the `rownum` field is removed, pay attention to the definition of the primary keys in the result table. If the definition is incorrect, the TopN query result is incorrect. The primary keys must be defined in the key list at the GROUP BY node before the TopN clause.

- Example

This example is a case from a customer in the video industry. Heavy traffic is generated each time a video is distributed. You can identify the most popular videos based on the video traffic. The following example identifies the top 5 videos that generate the most traffic per minute.

- Test statements


```
-- Read the original data storage table from Log Service.
CREATE TABLE sls_cdnlog_stream (
  vid VARCHAR, -- The video ID.
  rowtime TIMESTAMP, -- The time when the video was watched.
  response_size BIGINT, -- The traffic generated for watching the video.
  WATERMARK FOR rowtime as withOffset(rowtime, 0)
) WITH (
  type='sls',
  ...
);
-- Compute the consumed bandwidth by video ID in a 1-minute window.
CREATE VIEW cdnvid_group_view AS
SELECT vid,
  TUMBLE_START(rowtime, INTERVAL '1' MINUTE) AS start_time,
  SUM(response_size) AS rss
FROM sls_cdnlog_stream
GROUP BY vid, TUMBLE(rowtime, INTERVAL '1' MINUTE);
-- Create a result table.
CREATE TABLE hbase_out_cdnvidtoplog (
  vid VARCHAR,
  rss BIGINT,
  start_time VARCHAR,
  -- Do not store the rownum field in the result table.
  -- Pay attention to the definition of the primary keys. The primary keys must be defined in the key list
  at the GROUP BY node before the TopN clause.
  PRIMARY KEY(start_time, vid)
) WITH (
  type='RDS',
  ...
);
-- Identify and export the IDs of the top 5 videos that generate the most traffic per minute.
INSERT INTO hbase_out_cdnvidtoplog
-- Do not include the rownum field in the outer query.
SELECT vid, rss, start_time FROM
(
  SELECT
  vid, start_time, rss,
  ROW_NUMBER() OVER (PARTITION BY start_time ORDER BY rss DESC) as rownum
FROM
  cdnvid_group_view
)
WHERE rownum <= 5;
```

- Test data

vid (VARCHAR)	rowtime (TIMESTAMP)	response_size (BIGINT)
10000	2017-12-18 15:00:10	2000
10000	2017-12-18 15:00:15	4000
10000	2017-12-18 15:00:20	3000
10001	2017-12-18 15:00:20	3000
10002	2017-12-18 15:00:20	4000
10003	2017-12-18 15:00:20	1000
10004	2017-12-18 15:00:30	1000
10005	2017-12-18 15:00:30	5000
10006	2017-12-18 15:00:40	6000
10007	2017-12-18 15:00:50	8000

- Test results

start_time (VARCHAR)	vid (VARCHAR)	rss (BIGINT)
2017-12-18 15:00:00	10000	9000
2017-12-18 15:00:00	10007	8000
2017-12-18 15:00:00	10006	6000
2017-12-18 15:00:00	10005	5000
2017-12-18 15:00:00	10002	4000

3.1.4.7.9. CEP statements

MATCH_RECOGNIZE is a complex event processing (CEP) statement that identifies events from input data streams based on the specified rules and generates output events based on the specified method.

Syntax

```
SELECT [ ALL | DISTINCT ]
{ * | projectItem [, projectItem ]* }
FROM tableExpression
[MATCH_RECOGNIZE (
[PARTITION BY {partitionItem [, partitionItem]*}]
[ORDER BY {orderItem [, orderItem]*}]
[MEASURES {measureItem AS col [, measureItem AS col]*}]
[ONE ROW PER MATCH|ALL ROWS PER MATCH|ONE ROW PER MATCH WITH TIMEOUT ROWS|ALL ROWS PER MATCH WITH TIMEOUT ROWS]
[AFTER MATCH SKIP]
PATTERN (patternVariable[quantifier] [ patternVariable[quantifier]]*) WITHIN intervalExpression
DEFINE {patternVariable AS patternDefinationExpression [, patternVariable AS patternDefinationExpression]*}
)];
```

Clause	Description
PARTITION BY	Optional. You can use the clause to divide the rows of an input table into partitions based on one or more partition key columns.
ORDER BY	Optional. You can use the clause to sort the rows of a partition based on one or more columns. If you use multiple columns to sort the rows, specify EVENT TIME or PROCESS TIME as the first column.
MEASURES	You can use the clause to define the output events that are generated based on the matched input events.
ONE ROW PER MATCH	If you use the clause, only one output event is generated for each match.
ONE ROW PER MATCH WITH TIMEOUT ROWS	If you use the clause, an output event is generated for each match or time-out. The time-out period is specified by the WITHIN clause in PATTERN.  Note Blink V3.6.0 and later do not support the ONE ROW PER MATCH WITH TIMEOUT ROWS clause due to Calcite upgrades.
ALL ROWS PER MATCH	If you use the clause, an output event is generated for each input event upon each match.
ALL ROWS PER MATCH WITH TIMEOUT ROWS	If you use the clause, an output event is generated for each input event upon each match or time-out. The time-out period is specified by the WITHIN clause in PATTERN.  Note Blink V3.6.0 and later do not support the ALL ROWS PER MATCH WITH TIMEOUT ROWS clause due to Calcite upgrades.

Clause	Description
[ONE ROW PER MATCH ALL ROWS PER MATCH ONE ROW PER MATCH WITH TIMEOUT ROWS ALL ROWS PER MATCH WITH TIMEOUT ROWS]	Optional. By default, ONE ROW PER MATCH is used.
AFTER MATCH SKIP TO NEXT ROW	If you use the clause, the next match starts to be performed from the event that follows the first event in the sequence of matched events.
AFTER MATCH SKIP PAST LAST ROW	If you use the clause, the next match starts to be performed from the event that follows the last event in the sequence of matched events.
AFTER MATCH SKIP TO FIRST patternItem	If you use the clause, the next match starts to be performed from the first event that corresponds to patternItem in the sequence of matched events.
AFTER MATCH SKIP TO LAST patternItem	If you use the clause, the next match starts to be performed from the last event that corresponds to patternItem in the sequence of matched events.
PATTERN	You can use the clause to specify the rule that the sequence of events to be identified must meet. You must enclose the rule in parentheses (). The rule is specified by a set of custom patternVariable variables.  Note - If two patternVariable variables are separated with a space, no events exist between the events that correspond to the patternVariable variables. - If two patternVariable variables are separated with an arrow sign (->), other events may exist between the events that correspond to the patternVariable variables. - Blink V3.6.0 and later do not support the PATTERN clause due to Calcite upgrades.

Parameter description

- quantifier

The quantifier parameter specifies the number of occurrences of events that meet the patternVariable definition.

Value	Description
*	Zero or multiple times.
+	One or more times.
?	Zero or once.
{n}	n times.

Value	Description
{n,}	At least n times.
{n, m}	Ranges from n times to m times.
{,m}	At most m times.

By default, greedy matching is performed. For example, if the PATTERN clause is `A -> B+ -> C` and the input is `a bc1 bc2 c`, the output is `a bc1 bc2 c`. In the input, `bc1` and `bc2` means that the results must match B and C. To perform reluctant matching, append the quantifier with a question mark (?). You can use the following reluctant quantifiers:

- `*?`
- `+?`
- `{n}?`
- `{n,}?`
- `{n, m}?`
- `{,m}?`

 **Note** Blink V3.x and later do not support (e1 e2+) greedy matching. You can use `e1 e2+ e3` as not `e2` as an alternative. In the alternative method, at least one `e3` entry must be included to make sure that the output data is returned as expected.

In this case, the output that is generated for the input and the PATTERN setting in the preceding example changes to `a bc1 bc2, a bc1 bc2 c`.

- The WITHIN clause defines the maximum time span of the events that meet the specified rule in an event sequence.
 - The format of static windows is `INTERVAL 'string' timeUnit [ TO timeUnit ]`. For example, `INTERVAL '10' SECOND`, `INTERVAL '45' DAY`, `INTERVAL '10:20' MINUTE TO SECOND`, `INTERVAL '10:20.10' MINUTE TO SECOND`, `INTERVAL '10:20' HOUR TO MINUTE`, `INTERVAL '1-5' YEAR TO MONTH`.
 - The format of dynamic windows is `INTERVAL intervalExpression`, for example, `INTERVAL A.windowTime + 10`. In this example, A is the first patternVariable variable in the PATTERN clause. When you specify intervalExpression, you can use only the first patternVariable variable in the PATTERN clause. When you specify the intervalExpression parameter, you can use user defined functions (UDFs). The intervalExpression result indicates the window size that is measured in milliseconds. The data type of the result must be LONG.
 - The DEFINE statement specifies the meanings of patternVariable variables in the PATTERN clause. If a patternVariable variable is not defined in the DEFINE statement, the patternVariable variable is valid for each event.
- Functions in MEASURES and DEFINE statements

Function	Description
Row Pattern Column References	This function uses the format of <code>patternVariable.col</code> . This function is used to access the specified column of an event that corresponds to a patternVariable variable.

Function	Description
PREV	This function can be used in only the DEFINE statement. In most cases, this function is used in conjunction with the Row Pattern Column References function. This function is used to access the specified column of a previous event that has a specified offset. The event occurs before the events that are specified by the PATTERN clause. For example, you can use DOWN AS DOWN.price < PREV(DOWN.price) . In this example, PREV(A.price) specifies the price column value for the event that occurs before the current event. Note DOWN.price is equivalent to PREV(DOWN.price, 0) . PREV(DOWN.price) is equivalent to PREV(DOWN.price, 1) .
FIRST or LAST	In most cases, the FIRST or the LAST function is used in conjunction with the Row Pattern Column References function. You can use the FIRST or the LAST function to access the event that has a specified offset in the sequence of events. The sequence of events is specified by the PATTERN clause. The following examples are used to explain the functions: FIRST(A.price, 3) specifies the fourth event in the sequence of events that match the pattern A . LAST(A.price, 3) specifies the last but three event in the sequence of events that match the pattern A .

• Output columns

Function	Output column
ONE ROW PER MATCH	The columns that are specified in PARTITION BY and MEASURES are included. The columns that are specified in PARTITION BY do not need to be specified in MEASURES again.
ONE ROW PER MATCH WITH TIMEOUT ROWS	An output event is generated for each match or time-out. The time-out period is specified by the WITHIN clause in PATTERN.

- Note**
- When you define the PATTERN clause, we recommend that you define the WITHIN clause. If the WITHIN clause is not defined, the state size may grow larger.
 - The first column that is specified in the ORDER BY clause must be EVENT TIME or PROCESS TIME.

Examples

- Syntax in the example

```
SELECT *
FROM Ticker MATCH_RECOGNIZE (
  PARTITION BY symbol
  ORDER BY tstamp
  MEASURES STRT.tstamp AS start_tstamp,
  LAST(DOWN.tstamp) AS bottom_tstamp,
  LAST(UP.tstamp) AS end_tstamp
  ONE ROW PER MATCH
  AFTER MATCH SKIP TO NEXT ROW
  PATTERN (STRT DOWN+ UP+) WITHIN INTERVAL '10' SECOND
  DEFINE
  DOWN AS DOWN.price < PREV(DOWN.price),
  UP AS UP.price > PREV(UP.price)
);
```

- Test data

timestamp (TIMESTAMP)	card_id(VARCHAR)	location (VARCHAR)	action (VARCHAR)
2018-04-13 12:00:00	1	Beijing	Consumption
2018-04-13 12:05:00	1	Shanghai	Consumption
2018-04-13 12:10:00	1	Shenzhen	Consumption
2018-04-13 12:20:00	1	Beijing	Consumption

- Syntax in the test

Each credit card is identified by a unique card ID that is specified by the card_id parameter. If payments with a credit card are made within 10 minutes at two different locations, an alert is triggered. This helps you monitor unauthorized operations of credit cards.

```

CREATE TABLE datahub_stream (
  `timestamp`      TIMESTAMP,
  card_id          VARCHAR,
  location         VARCHAR,
  `action`         VARCHAR,
  WATERMARK wf FOR `timestamp` AS withOffset(`timestamp`, 1000)
) WITH (
  type = 'datahub'
  ...
);
CREATE TABLE rds_out (
  start_timestamp  TIMESTAMP,
  end_timestamp    TIMESTAMP,
  card_id          VARCHAR,
  event            VARCHAR
) WITH (
  type = 'rds'
  ...
);
--Define the computational logic.
insert into rds_out
select
  `start_timestamp`,
  `end_timestamp`,
  card_id, `event`
from datahub_stream
MATCH_RECOGNIZE (
  PARTITION BY card_id --Partition the table by card ID. The data that has the same card ID is allocated to
the same compute node.
  ORDER BY `timestamp` --Sort events by time in a window.
  MEASURES      --Define how to generate output events based on the input events that are matched.
    e2.`action` as `event`,
    e1.`timestamp` as `start_timestamp`, --Specify the time of the first event as the start_timestamp v
alue.
    LAST(e2.`timestamp`) as `end_timestamp` --Specify the time of the last event as the end_timestam
p value.
  ONE ROW PER MATCH      --The system generates an output event for each match.
  AFTER MATCH SKIP TO NEXT ROW --The system performs the next match from the next row after each m
atch.
  PATTERN (e1 e2+) WITHIN INTERVAL '10' MINUTE --Define two events: e1 and e2.
  DEFINE          --Define the meanings of patternVariable variables in the PATTERN clause.
    e1 as e1.action = 'Consumption', --Mark the action of the e1 event as Consumption.
    e2 as e2.action = 'Consumption' and e2.location <> e1.location --Mark the action of the e2 event as Con
sumption. The locations of e1 and e2 are different.
);

```

 **Note** In some scenarios, some data meets the CEP condition but no outputs are returned. This occurs because only data that meets the watermark > e2.ts condition is processed. No input data flows into the system after e2 and the watermark is always e2.ts - 1000. As a result, the e2 data cannot be processed. Therefore, no outputs are returned.

- Test result

start_timestamp (TIMESTAMP)	end_timestamp (TIMESTAMP)	card_id (VARCHAR)	event (VARCHAR)
2018-04-13 12:00:00.0	2018-04-13 12:05:00.0	1	Consumption
2018-04-13 12:05:00.0	2018-04-13 12:10:00.0	1	Consumption

3.1.4.8. Create a view

In Realtime Compute, you can define a view to query data in one or more tables in a database. You can simplify the development process by using views.

A view displays a logical table that helps to describe the computational logic. It does not physically store any data. To simplify queries, you can use the SQL statement to define a manageable view. A view can be created based on fields of one or more tables in a database. Views do not require any disk space.

Syntax

```
CREATE VIEW viewName
  [ (columnName[ , columnName]*) ]
  AS queryStatement;
```

Example 1

```
CREATE VIEW LargeOrders(r, t, c, u) AS
SELECT rowtime,
       productId,
       c,
       units
FROM Orders;
INSERT INTO rds_output
SELECT
  r,t,c,u
FROM LargeOrders;
```

Example 2

Input example

a (VARCHAR)	b (BIGINT)	c (TIMESTAMP)
test1	1	1506823820000
test2	1	1506823850000
test1	1	1506823810000
test2	1	1506823840000

a (VARCHAR)	b (BIGINT)	c (TIMESTAMP)
test2	1	1506823870000
test1	1	1506823830000
test2	1	1506823860000

Sample code

```
CREATE TABLE datahub_stream (  
  a VARCHAR,  
  b BIGINT,  
  c TIMESTAMP,  
  d AS PROCTIME()  
) WITH (  
  type='datahub',  
  endPoint='http://dh-cn-hangzhou.aliyuncs.com',  
  project='blink_test',  
  topic='window_count',  
  accessId='yourAccessId',  
  accessKey='yourAccessSecret'  
);  
CREATE TABLE rds_output (  
  a varchar,  
  b TIMESTAMP,  
  cnt BIGINT,  
  PRIMARY KEY(a)  
)with(  
  type = 'rds',  
  url='jdbc:mysql://rm-bp1gz4k20****.mysql.rds.aliyuncs.com:3306/****',  
  tableName='yourTableName',  
  userName='yourAccessUserName',  
  password='yourAccessPassword'  
);  
CREATE VIEW rds_view AS  
SELECT a,  
  CAST(HOP_START(d, interval '5' second, interval '30' second) AS TIMESTAMP) AS cc,  
  sum(b) AS cnt  
FROM datahub_stream  
GROUP BY HOP(d, interval '5' second, interval '30' second),a;  
INSERT INTO rds_output  
SELECT  
  a,  
  cc,  
  cnt  
FROM rds_view  
WHERE cnt=4
```

Output example

a (VARCHAR)	b (TIMESTAMP)	cnt (BIGINT)
test2	2017-11-06 16:54:10	4

3.1.4.9. Window functions

3.1.4.9.1. Overview

This topic describes the window functions, time attributes, and window types that Flink SQL supports.

Window functions

Flink SQL supports aggregation over infinite windows. You do not need to explicitly define windows in your SQL statements. Flink SQL also supports aggregation over a specific window. For example, to count the number of users who clicked a URL in the last minute, you can define a window to collect the data about user clicks that occur in the last minute. Then, you can compute the data in the window to obtain the result.

Flink SQL supports window aggregates and over aggregates. This topic describes window aggregates. Window aggregates support the windows that are defined based on the following two time attributes: event time and processing time. For each time attribute, Flink SQL supports three window types: tumbling window, sliding window, and session window.

Time attributes

Flink SQL supports two time attributes: event time and processing time. Realtime Compute for Apache Flink aggregates data in windows based on the following time attributes:

- **Event time**: the event time that you provide in the table schema. In most cases, the event time is the original time when data is created.
- **Processing time**: the local system time at which the system processes an event.

 **Note** For more information about the time attributes, see [Time attributes](#).

Cascading windows

The event time attribute of the Rowtime column no longer takes effect after a window operation is complete. You can use a helper function such as `TUMBLE_ROWTIME`, `HOP_ROWTIME`, or `SESSION_ROWTIME` to obtain `max(rowtime)` of the rowtime column in a window. You can use the obtained value as the rowtime of the time window. The value is `window_end - 1` and is of the `TIMESTAMP` data type. The `TIMESTAMP` value has the rowtime attribute. For example, if the time span for a window is `[00:00, 00:15]`, `00:14:59.999` is returned.

In the following example, 1-hour tumbling windows are used to aggregate data based on the aggregation results of 1-minute tumbling windows. This helps you meet various window requirements.

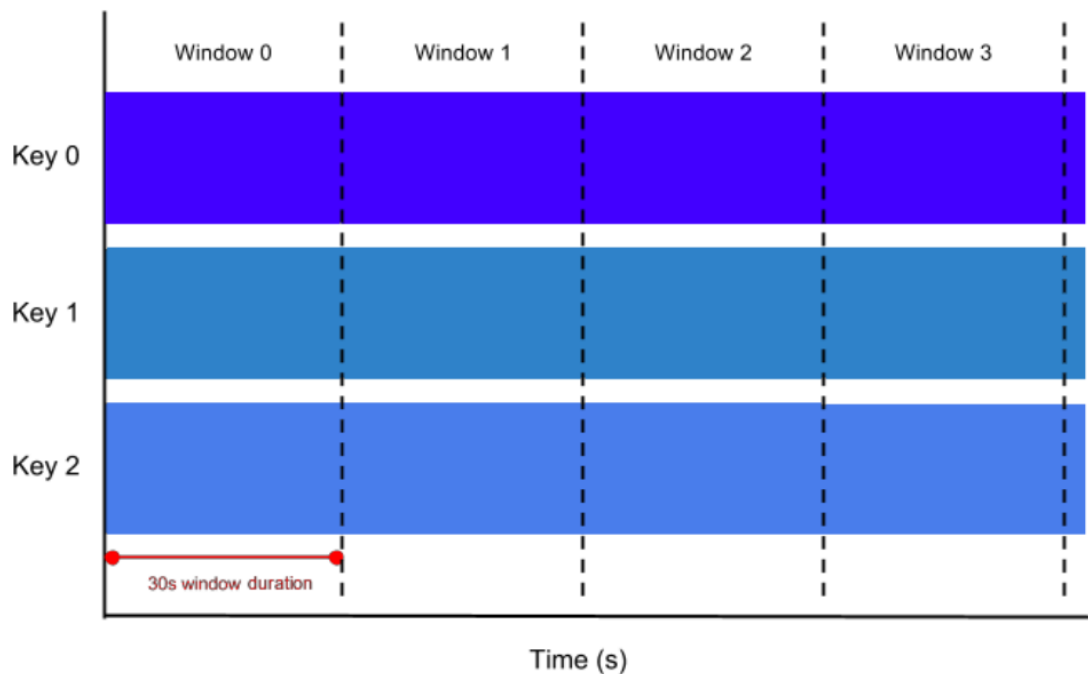
```
CREATE TABLE user_clicks(
  username varchar,
  click_url varchar,
  ts timeStamp,
  WATERMARK wk FOR ts as withOffset(ts, 2000) --Define a watermark for the rowtime.
) with (
  type='datahub',
  ...
);
CREATE TABLE tumble_output(
  window_start TIMESTAMP,
  window_end TIMESTAMP,
  username VARCHAR,
  clicks BIGINT
) with (
  type='print'
);
CREATE VIEW one_minute_window_output as
SELECT
  // Use each TUMBLE_ROWTIME value as the aggregation time for the level-two window.
  TUMBLE_ROWTIME(ts, INTERVAL '1' MINUTE) as rowtime,
  username,
  COUNT(click_url) as cnt
FROM user_clicks
GROUP BY TUMBLE(ts, INTERVAL '1' MINUTE), username;
INSERT INTO tumble_output
SELECT
  TUMBLE_START(rowtime, INTERVAL '1' HOUR),
  TUMBLE_END(rowtime, INTERVAL '1' HOUR),
  username,
  SUM(cnt)
FROM one_minute_window_output
GROUP BY TUMBLE(rowtime, INTERVAL '1' HOUR), username;
```

3.1.4.9.2. Tumbling window

This topic describes how to use the tumbling window function in Realtime Compute for Apache Flink.

Introduction to a tumbling window

A tumbling window assigns segments of a data stream to a window with a specified size. Generally, tumbling windows are fixed in size and do not overlap each other. For example, if a 5-minute tumbling window is defined, an infinite data stream is divided based on the time period into windows such as [0:00 - 0:05) , [0:05, 0:10) , and [0:10, 0:15) . The following figure shows a 30-second tumbling window.



Syntax

The TUMBLE function is used to define a tumbling window in a GROUP BY clause.

```
TUMBLE(<time-attr>, <size-interval>)
<size-interval>: INTERVAL 'string' timeUnit
```

Note The `<time-attr>` parameter must be a valid time attribute in a time stream to specify whether the time is the processing time or event time. For more information, see [Overview](#), [Time attributes](#), and [Watermark](#).

Window identifier functions

A window identifier function specifies the start time, end time, or the time attribute of a window. The time attribute is used to aggregate lower-level windows.

Window identifier function	Return type	Description
<code>TUMBLE_START(time-attr, size-interval)</code>	TIMESTAMP	Returns the start time (including the boundary value) of a window. For example, if the window is [00:10, 00:15) , 00:10 is returned.

Window identifier function	Return type	Description
<code>TUMBLE_END(time-attr, size-interval)</code>	TIMESTAMP	Returns the end time (including the boundary value) of a window. For example, if the window is <code>[00:00, 00:15]</code> , <code>00:15</code> is returned.
<code>TUMBLE_ROWTIME(time-attr, size-interval)</code>	TIMESTAMP(rowtime-attr)	Returns the end time (excluding the boundary value) of a window. For example, if the window is <code>[00:00, 00:15]</code> , <code>00:14:59.999</code> is returned. The return value is a rowtime attribute, based on which time operations can be performed. For example, a cascading window function can be used only in windows based on the event time.
<code>TUMBLE_PROCTIME(time-attr, size-interval)</code>	TIMESTAMP(rowtime-attr)	Returns the end time (excluding the boundary value) of a window. For example, if the window is <code>[00:00, 00:15]</code> , <code>00:14:59.999</code> is returned. The return value is a proctime attribute, based on which time operations can be performed. For example, a cascading window function can be used only in windows defined based on the processing time.

Example of counting the number of clicks per user on a specific website every one minute based on the event time

- Test data

username (VARCHAR)	click_url (VARCHAR)	ts (TIMESTAMP)
Jark	<code>http://taobao.com/xxx</code>	2017-10-10 10:00:00.0
Jark	<code>http://taobao.com/xxx</code>	2017-10-10 10:00:10.0
Jark	<code>http://taobao.com/xxx</code>	2017-10-10 10:00:49.0
Jark	<code>http://taobao.com/xxx</code>	2017-10-10 10:01:05.0
Jark	<code>http://taobao.com/xxx</code>	2017-10-10 10:01:58.0
Timo	<code>http://taobao.com/xxx</code>	2017-10-10 10:02:10.0

- Test statements

```
CREATE TABLE user_clicks(
  username varchar,
  click_url varchar,
  ts timestamp,
  WATERMARK wk FOR ts as withOffset(ts, 2000) -- Define a watermark for rowtime.
) with (
  type='datahub',
  ...
);
CREATE TABLE tumble_output(
  window_start TIMESTAMP,
  window_end TIMESTAMP,
  username VARCHAR,
  clicks BIGINT
) with (
  type='RDS'
);
INSERT INTO tumble_output
SELECT
  TUMBLE_START(ts, INTERVAL '1' MINUTE) as window_start,
  TUMBLE_END(ts, INTERVAL '1' MINUTE) as window_end,
  username,
  COUNT(click_url)
FROM user_clicks
GROUP BY TUMBLE(ts, INTERVAL '1' MINUTE), username;
```

- Test results

window_start (TIMESTAMP)	window_end (TIMESTAMP)	username (VARCHAR)	clicks (BIGINT)
2017-10-10 10:00:00.0	2017-10-10 10:01:00.0	Jark	3
2017-10-10 10:01:00.0	2017-10-10 10:02:00.0	Jark	2
2017-10-10 10:02:00.0	2017-10-10 10:03:00.0	Timo	1

Example of counting the number of clicks per user on a specific website every one minute based on the processing time

- Test data

username (VARCHAR)	click_url (VARCHAR)
Jark	<code>http://taobao.com/xxx</code>
Jark	<code>http://taobao.com/xxx</code>
Jark	<code>http://taobao.com/xxx</code>

username (VARCHAR)	click_url (VARCHAR)
Jark	<code>http://taobao.com/xxx</code>
Jark	<code>http://taobao.com/xxx</code>
Timo	<code>http://taobao.com/xxx</code>

- Test statements

```
CREATE TABLE window_test (  
  username VARCHAR,  
  click_url VARCHAR,  
  ts as PROCTIME()  
) WITH (  
  type='datahub',  
  ...  
);  
CREATE TABLE tumble_output(  
  window_start TIMESTAMP,  
  window_end TIMESTAMP,  
  username VARCHAR,  
  clicks BIGINT  
) with (  
  type='print'  
);  
INSERT INTO tumble_output  
SELECT  
  TUMBLE_START(ts, INTERVAL '1' MINUTE),  
  TUMBLE_END(ts, INTERVAL '1' MINUTE),  
  username,  
  COUNT(click_url)  
FROM window_test  
GROUP BY TUMBLE(ts, INTERVAL '1' MINUTE), username;
```

- Test results

window_start (TIMESTAMP)	window_end (TIMESTAMP)	username (VARCHAR)	clicks (BIGINT)
<code>2019-04-11 14:43:00.000</code>	<code>2019-04-11 14:44:00.000</code>	Jark	5
<code>2019-04-11 14:43:00.000</code>	<code>2019-04-11 14:44:00.000</code>	Timo	1

 **Note** Local debugging is instantaneous and the processing time may be less than 1s. Therefore, if the processing time attribute is used to aggregate data in windows, local debugging may fail.

3.1.4.9.3. Sliding windows

This topic describes how to use sliding window functions in Realtime Compute for Apache Flink.

Note In Realtime Compute for Apache Flink, sliding windows cannot be used in conjunction with `LAST_VALUE`, `FIRST_VALUE`, or `TopN` functions.

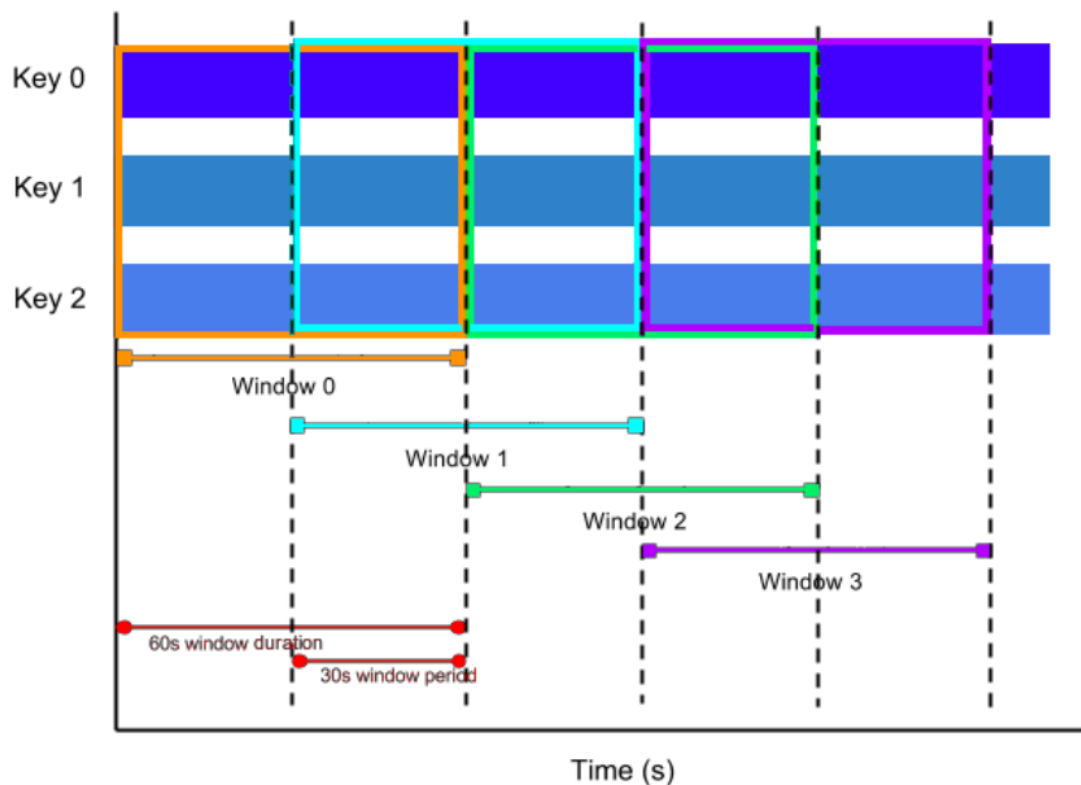
Introduction

A sliding window is also known as a hop window. Unlike tumbling windows, sliding windows can overlap with each other.

A sliding window is defined by the following parameters: slide and size. The slide parameter specifies the length of a sliding step. The size parameter specifies the size of the window.

- If the value of slide is smaller than that of size, the windows overlap with each other and each element is assigned to multiple windows.
- If the value of slide is equal to that of size, the windows are tumbling windows.
- If the value of slide is greater than that of size, the windows do not overlap with each other but are separated with gaps.

In most cases, most elements are assigned to multiple windows and the windows overlap with each other. Sliding windows are used to calculate moving averages. For example, to calculate the data average in the last 5 minutes every 10 seconds, set slide to 10 seconds and set size to 5 minutes. The following figure shows sliding windows for which the slide value is 30 seconds and the size value is 1 minute.



Syntax of sliding window functions

You can use the HOP function to define a sliding window in a GROUP BY clause.

```
HOP(<time-attr>, <slide-interval>, <size-interval>)
<slide-interval>: INTERVAL 'string' timeUnit
<size-interval>: INTERVAL 'string' timeUnit
```

Note

The `<time-attr>` parameter must be a valid time attribute field in a stream. This parameter specifies whether the time attribute is processing time or event time. For more information about [Time attributes](#) and [Watermark](#), see [Overview](#).

Window identifier functions

A window identifier function specifies the start time, end time, or time attribute of a specified window. The time attribute is used to aggregate lower-level windows.

Function	Return type	Description
<code>HOP_START (<time-attr>, <slide-interval>, <size-interval>)</code>	TIMESTAMP	Returns the start time of the window. The boundary for the start time is included in the time span of the window. For example, if the time span of a window is <code>[00:10, 00:15)</code> , <code>00:10</code> is returned.
<code>HOP_END (<time-attr>, <slide-interval>, <size-interval>)</code>	TIMESTAMP	Returns the end time of the window. The boundary for the end time is included in the time span of the window. For example, if the time span of a window is <code>[00:00, 00:15)</code> , <code>00:15</code> is returned.
<code>HOP_ROWTIME (<time-attr>, <slide-interval>, <size-interval>)</code>	TIMESTAMP (rowtime-attr)	Returns the end time of the window. The boundary for the end time is excluded from the time span of the window. For example, if the time span of a window is <code>[00:00, 00:15)</code> , <code>00:14:59.999</code> is returned. The returned value is a rowtime attribute value based on which time operations can be performed. This function can be used in only the windows that are based on the event time, such as cascading windows. For more information, see Cascading windows .
<code>HOP_PROCTIME (<time-attr>, <slide-interval>, <size-interval>)</code>	TIMESTAMP (rowtime-attr)	Returns the end time of the window. The boundary for the end time is excluded from the time span of the window. For example, if the time span of a window is <code>[00:00, 00:15)</code> , <code>00:14:59.999</code> is returned. The returned value is a proctime attribute value based on which time operations can be performed. This function can be used in only the windows that are based on the processing time, such as cascading windows. For more information, see Cascading windows .

Examples

In the following example, a 1-minute window slides once every 30 seconds. You can use the windows to count the number of clicks per user over the last minute every 30 seconds.

- Test data

username (VARCHAR)	click_url (VARCHAR)	ts (TIMESTAMP)
Jark	http://taobao.com/xxx	2017-10-10 10:00:00.0
Jark	http://taobao.com/xxx	2017-10-10 10:00:10.0
Jark	http://taobao.com/xxx	2017-10-10 10:00:49.0
Jark	http://taobao.com/xxx	2017-10-10 10:01:05.0
Jark	http://taobao.com/xxx	2017-10-10 10:01:58.0
Timo	http://taobao.com/xxx	2017-10-10 10:02:10.0

- Test statements

```
CREATE TABLE user_clicks (  
  username VARCHAR,  
  click_url VARCHAR,  
  ts TIMESTAMP,  
  WATERMARK wk FOR ts AS WITHOFFSET (ts, 2000)--Define the watermark for the rowtime.  
) WITH ( TYPE = 'datahub',  
  ...);  
CREATE TABLE hop_output (  
  window_start TIMESTAMP,  
  window_end TIMESTAMP,  
  username VARCHAR,  
  clicks BIGINT  
) WITH (TYPE = 'rds',  
  ...);  
INSERT INTO  
  hop_output  
SELECT  
  HOP_START (ts, INTERVAL '30' SECOND, INTERVAL '1' MINUTE),  
  HOP_END (ts, INTERVAL '30' SECOND, INTERVAL '1' MINUTE),  
  username,  
  COUNT (click_url)  
FROM  
  user_clicks  
GROUP BY  
  HOP (ts, INTERVAL '30' SECOND, INTERVAL '1' MINUTE),  
  username
```

- Test result

window_start (TIMESTAMP)	window_end (TIMESTAMP)	username (VARCHAR)	clicks (BIGINT)
2017-10-10 09:59:30.0	2017-10-10 10:00:30.0	Jark	2
2017-10-10 10:00:00.0	2017-10-10 10:01:00.0	Jark	3
2017-10-10 10:00:30.0	2017-10-10 10:01:30.0	Jark	2
2017-10-10 10:01:00.0	2017-10-10 10:02:00.0	Jark	2
2017-10-10 10:01:30.0	2017-10-10 10:02:30.0	Jark	1
2017-10-10 10:02:00.0	2017-10-10 10:03:00.0	Timo	1
2017-10-10 10:02:30.0	2017-10-10 10:03:30.0	Timo	1

If a sliding window cannot read the time at which data enters the window, the start time of the first window is moved forward. You can use the following formula to calculate the time interval by which the start time is moved forward: Time interval = Window duration - Sliding step.

Window duration (seconds)	Sliding step (seconds)	Event Time	Start time of the first window	End time of the first window
120	30	2019-07-31 10:00:00.0	2019-07-31 09:58:30.0	2019-07-31 10:00:30.0
60	10	2019-07-31 10:00:00.0	2019-07-31 09:59:10.0	2019-07-31 10:00:10.0

3.1.4.9.4. Session window

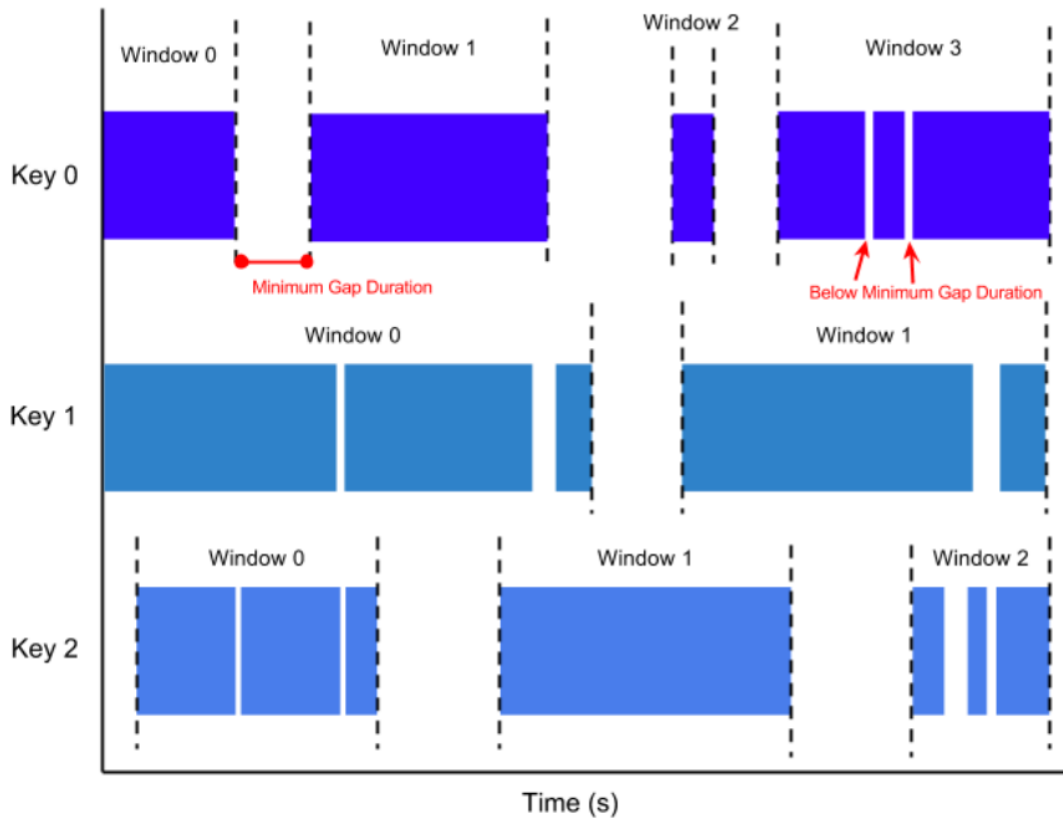
This topic describes how to use the session window function in Realtime Compute for Apache Flink.

Description

A session window groups elements by session activity. Unlike tumbling and sliding windows, session windows do not overlap and are not fixed in size. If a session window does not receive elements within a period of time, the session is disconnected and the window is closed.

A session window is configured by using a gap, which defines the length of the inactive period. For example, a data stream that represents mouse click activities may include highly clustered mouse click events, separated by inactive periods. The data that arrives after a specified gap is assigned to a new window.

The following figure shows a session window. Different keys form different windows due to differences in data distribution.



Syntax

The SESSION function is used to define a session window in a GROUP BY clause.

```
SESSION(<time-attr>, <gap-interval>)
<gap-interval>: INTERVAL 'string' timeUnit
```

Note The `<time-attr>` parameter must be a valid time attribute in a data stream to specify whether the time is the processing time or event time. For more information, see [Overview](#), [Time attributes](#), and [Watermark](#).

Window identifier functions

A window identifier function specifies the start or end time of a window, or the window time attribute for the aggregation of lower-level windows.

Window identifier function	Return type	Description
<code>SESSION START (<time-attr>, <gap-interval>)</code>	Timestamp	Returns the start time (including the boundary value) of the window. For example, if the window is <code>[00:10, 00:15]</code> , <code>00:10</code> is returned. This is the first time recorded in the session window.
<code>SESSION END (<time-attr>, <gap-interval>)</code>	Timestamp	Returns the end time (including the boundary value) of the window. For example, if the window is <code>[00:00, 00:15]</code> , <code>00:15</code> is returned. This is the last time recorded in the session window plus <code><gap-interval></code> .
<code>SESSION ROWTIME (<time-attr>, <gap-interval>)</code>	TIMESTAMP (rowtime-attr)	Returns the end time (excluding the boundary value) of the window. For example, if the window is <code>(00:00, 00:15)</code> , <code>00:14:59.999</code> is returned. The return value is an event time attribute, based on which time type operations such as window cascading can be performed. The function applies only to windows based on the event time.
<code>SESSION PROCTIME (<time-attr>, <gap-interval>)</code>	TIMESTAMP (rowtime-attr)	Returns the end time (excluding the boundary value) of the window. For example, if the window is <code>(00:00, 00:15)</code> , <code>00:14:59.999</code> is returned. The return value is a processing time attribute, based on which time type operations such as window cascading can be performed. The function applies only to windows based on the processing time.

Example

The following example describes how to compute the number of clicks per user during each active session. The session timeout interval is 30 seconds.

- Test data

username (VARCHAR)	click_url (VARCHAR)	ts (TIMESTAMP)
Jark	<code>http://taobao.com/xxx</code>	<code>2017-10-10 10:00:00.0</code>
Jark	<code>http://taobao.com/xxx</code>	<code>2017-10-10 10:00:10.0</code>
Jark	<code>http://taobao.com/xxx</code>	<code>2017-10-10 10:00:49.0</code>
Jark	<code>http://taobao.com/xxx</code>	<code>2017-10-10 10:01:05.0</code>
Jark	<code>http://taobao.com/xxx</code>	<code>2017-10-10 10:01:58.0</code>
Timo	<code>http://taobao.com/xxx</code>	<code>2017-10-10 10:02:10.0</code>

- Test statements

```
CREATE TABLE user_clicks(  
  username varchar,  
  click_url varchar,  
  ts timeStamp,  
  WATERMARK wk FOR ts as withOffset(ts, 2000) -- Define a watermark for rowtime.  
) with (  
  type='datahub',  
  ...  
);  
CREATE TABLE session_output(  
  window_start TIMESTAMP,  
  window_end TIMESTAMP,  
  username VARCHAR,  
  clicks BIGINT  
) with (  
  type='rds',  
  ...  
);  
INSERT INTO session_output  
SELECT  
  SESSION_START(ts, INTERVAL '30' SECOND),  
  SESSION_END(ts, INTERVAL '30' SECOND),  
  username,  
  COUNT(click_url)  
FROM user_clicks  
GROUP BY SESSION(ts, INTERVAL '30' SECOND), username;
```

- Test results

window_start (TIMESTAMP)	window_end (TIMESTAMP)	username (VARCHAR)	clicks (BIGINT)
2017-10-10 10:00:00.0	2017-10-10 10:00:40.0	Jark	2
2017-10-10 10:00:49.0	2017-10-10 10:01:35.0	Jark	2
2017-10-10 10:01:58.0	2017-10-10 10:02:28.0	Jark	1
2017-10-10 10:02:10.0	2017-10-10 10:02:40.0	Timo	1

3.1.4.9.5. OVER window

An OVER window is a standard window used in traditional databases. Over aggregate is different from window aggregate. In streaming data that uses OVER windows, each element corresponds to an OVER window. An OVER window can be determined based on an actual row or an actual value (timestamp value) of an element. Elements of a stream are distributed across multiple windows.

In a stream that applies the OVER window, each element corresponds to an OVER window and triggers data computing once. The row determined by each element that triggers computing is the last row of the window where the element is located. In the underlying implementation of Realtime Compute, the OVER window data is centrally managed. Only one copy of the data is stored. Logically, an OVER window is created for each element. Realtime Compute for Apache Flink calculates the data for each OVER window and then deletes the data that is no longer used after the calculation is complete.

Syntax

```
SELECT
  agg1(col1) OVER (definition1) AS colName,
  ...
  aggN(colN) OVER (definition1) AS colNameN
FROM Tab1;
```

- `agg1(col1)`: aggregates input data based on the `col1` column specified by GROUP BY.
- `OVER (definition1)`: defines an OVER window.
- `AS colName`: specifies the alias of a column.

Note

- `OVER (definition1)` for `agg1` through `aggN` must be the same.
- The alias specified by `AS` can be queried by using an outer SQL statement.

Window types

In Flink SQL, OVER windows are defined in compliance with the standard SQL syntax. The traditional OVER windows are not classified into fine-grained window types. OVER windows are classified into the following two types based on the ways of determining computed rows:

- **ROWS OVER window**: Each row of elements is treated as a new computed row. A new window is generated for each row.
- **RANGE OVER window**: All rows of elements with the same timestamp are treated as one computed row and are assigned to the same window.

Attributes

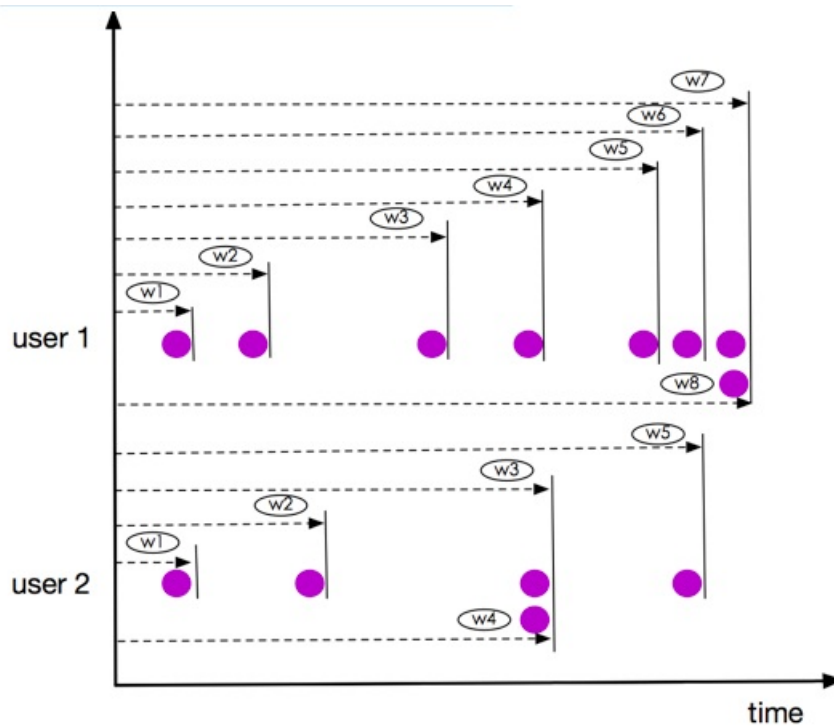
Orthogonal attribute	Description	proctime	eventtime
ROWS OVER Window	A window is determined based on the actual row of an element.	Supported	Supported
RANGE OVER Window	A window is determined based on the timestamp value of an element.	Supported	Supported

ROWS OVER window

- Description

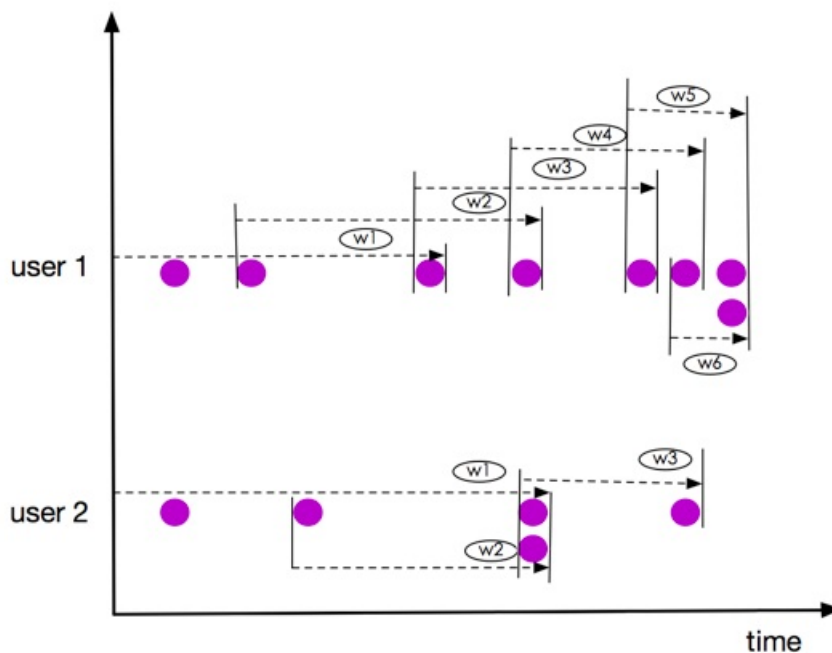
For a ROWS OVER window, a window is generated for each element. ROWS OVER windows have two types: unbounded and bounded.

The following figure shows unbounded ROWS OVER windows.



Note This figure shows that elements in windows w7 and w8 of user 1 arrive at the same time, and elements in windows w3 and w4 of user 2 arrive at the same time. However, the elements are assigned to different windows. ROWS OVER windows are different from RANGE OVER windows in this aspect.

The following figure shows bounded ROWS OVER windows, in which each window has three elements (two elements preceding the current element).



Note As shown in the preceding figure, elements in windows w5 and w6 of user 1 arrive at the same time, and elements in windows w1 and w2 of user 2 arrive at the same time. However, the elements are assigned to different windows. ROWS OVER windows are different from RANGE OVER windows in this aspect.

- Syntax

```
SELECT
  agg1(col1) OVER(
    [PARTITION BY (value_expression1,..., value_expressionN)]
    ORDER BY timeCol
    ROWS
    BETWEEN (UNBOUNDED | rowCount) PRECEDING AND CURRENT ROW) AS colName, ...
FROM Tab1;
```

- value_expression: specifies the value expression used for partitioning.
- timeCol: specifies the time field used to sort elements.
- rowCount: specifies the number of rows that precede the current row.

- Example

This example describes bounded ROWS OVER windows. Assume that an on-sale product table contains item IDs, item types, launch time, and prices. Calculate the highest price among the three products similar to the current product before the current product is on sale.

- Test data

Item ID	Item type	Launch time	Price
ITEM001	Electronic	2017-11-11 10:01:00	20
ITEM002	Electronic	2017-11-11 10:02:00	50
ITEM003	Electronic	2017-11-11 10:03:00	30
ITEM004	Electronic	2017-11-11 10:03:00	60
ITEM005	Electronic	2017-11-11 10:05:00	40
ITEM006	Electronic	2017-11-11 10:06:00	20
ITEM007	Electronic	2017-11-11 10:07:00	70
ITEM008	Clothes	2017-11-11 10:08:00	20

- Test statements

```
CREATE TABLE tmall_item(  
  itemID VARCHAR,  
  itemType VARCHAR,  
  onSellTime TIMESTAMP,  
  price DOUBLE,  
  WATERMARK onSellTime FOR onSellTime as withOffset(onSellTime, 0)  
)  
WITH (  
  type = 'sls',  
  ...  
);  
SELECT  
  itemID,  
  itemType,  
  onSellTime,  
  price,  
  MAX(price) OVER (  
    PARTITION BY itemType  
    ORDER BY onSellTime  
    ROWS BETWEEN 2 preceding AND CURRENT ROW) AS maxPrice  
FROM tmall_item;
```

- Test results

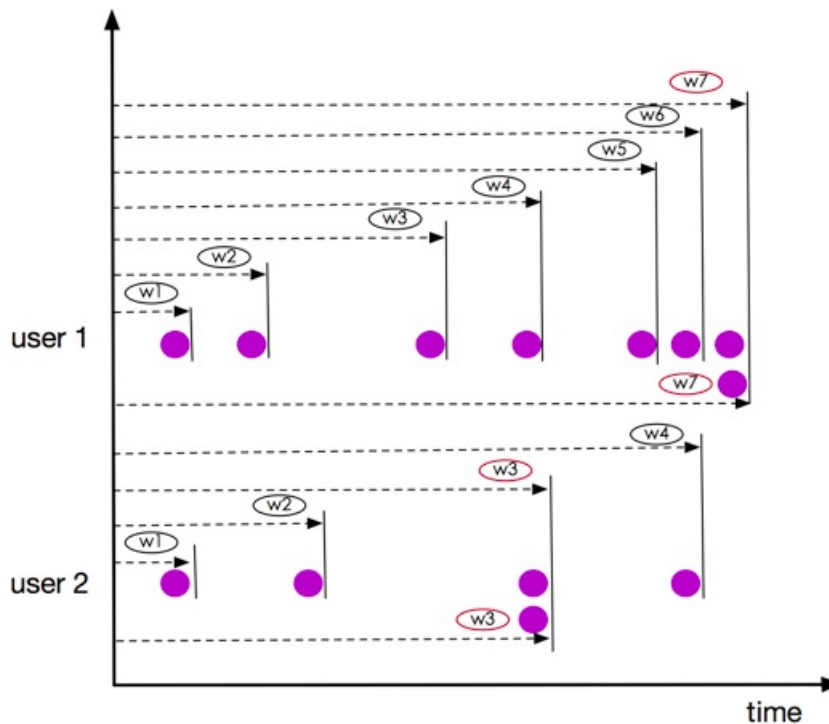
itemID	itemType	onSellTime	price	maxPrice
ITEM001	Electronic	2017-11-11 10:01:00	20	20
ITEM002	Electronic	2017-11-11 10:02:00	50	50
ITEM003	Electronic	2017-11-11 10:03:00	30	50
ITEM004	Electronic	2017-11-11 10:03:00	60	60
ITEM005	Electronic	2017-11-11 10:05:00	40	60
ITEM006	Electronic	2017-11-11 10:06:00	20	60
ITEM007	Electronic	2017-11-11 10:07:00	70	70
ITEM008	Clothes	2017-11-11 10:08:00	20	20

RANGE OVER window

- Description

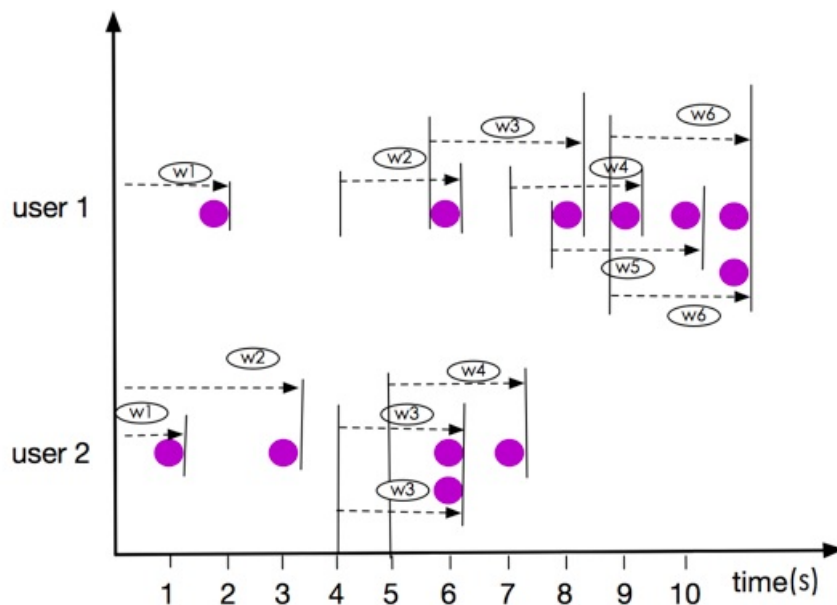
For a RANGE OVER window, all elements with the same timestamp are assigned to the same window. RANGE OVER windows have two types: unbounded and bounded.

The following figure shows unbounded RANGE OVER windows.



Note This figure shows that elements in the two w7 windows of user 1 arrive at the same time, and elements in the two w3 windows of user 2 arrive at the same time. Elements with the same timestamp are assigned to the same window. RANGE OVER windows are different from ROWS OVER windows in this aspect.

The following figure shows bounded RANGE OVER windows, in which a 3-second window has an interval of 2 seconds (INTERVAL '2' SECOND) .



Note As shown in the preceding figure, elements in the two w6 windows of user 1 arrive at the same time, and elements in the two w3 windows of user 2 arrive at the same time. Elements with the same timestamps are assigned to the same window. RANGE OVER windows are different from ROWS OVER windows in this aspect.

- Syntax

```
SELECT
  agg1(col1) OVER(
    [PARTITION BY (value_expression1,..., value_expressionN)]
    ORDER BY timeCol
    RANGE
    BETWEEN (UNBOUNDED | timeInterval) PRECEDING AND CURRENT ROW) AS colName,
  ...
FROM Tab1;
```

- value_expression: specifies the value expression used for partitioning.
- timeCol: specifies the time field used to sort elements.
- timeInterval: specifies the time interval between the time of the current row and that of the element row to which it can be traced back.

- Example

This example describes bounded RANGE OVER windows. Assume that an on-sale product table contains item IDs, item types, launch time, and prices. Calculate the highest price among similar products that are on sale two minutes earlier than the current product.

o Test data

Item ID	Item type	Launch time	Price
ITEM001	Electronic	2017-11-11 10:01:00	20
ITEM002	Electronic	2017-11-11 10:02:00	50
ITEM003	Electronic	2017-11-11 10:03:00	30
ITEM004	Electronic	2017-11-11 10:03:00	60
ITEM005	Electronic	2017-11-11 10:05:00	40
ITEM006	Electronic	2017-11-11 10:06:00	20
ITEM007	Electronic	2017-11-11 10:07:00	70
ITEM008	Clothes	2017-11-11 10:08:00	20

o Test statements

```
CREATE TABLE tmall_item(
  itemID VARCHAR,
  itemType VARCHAR,
  onSellTime TIMESTAMP,
  price DOUBLE,
  WATERMARK onSellTime FOR onSellTime as withOffset(onSellTime, 0)
)
WITH (
  type = 'sls',
  ...
);
SELECT
  itemID,
  itemType,
  onSellTime,
  price,
  MAX(price) OVER (
    PARTITION BY itemType
    ORDER BY onSellTime
    RANGE BETWEEN INTERVAL '2' MINUTE preceding AND CURRENT ROW) AS maxPrice
FROM tmall_item;
```

o Test results

itemID	itemType	onSellTime	price	maxPrice
ITEM001	Electronic	2017-11-11 10:01:00	20	20
ITEM002	Electronic	2017-11-11 10:02:00	50	50
ITEM003	Electronic	2017-11-11 10:03:00	30	50
ITEM004	Electronic	2017-11-11 10:03:00	60	60
ITEM005	Electronic	2017-11-11 10:05:00	40	60
ITEM006	Electronic	2017-11-11 10:06:00	20	40
ITEM007	Electronic	2017-11-11 10:07:00	70	70
ITEM008	Clothes	2017-11-11 10:08:00	20	20

3.1.4.10. Logical operations

3.1.4.10.1. =

Syntax

A = B

Input parameters

Parameter	Data type
A	INT
B	INT

Description

Returns TRUE if A is equal to B, and returns FALSE in other cases.

Example

- Input example

int1 (INT)	int2 (INT)	int3 (INT)
97	65	65

- Sample code

```
SELECT int1 as aa
FROM T1
WHERE int3 = int2;
```

- Output example

aa (INT)
97

3.1.4.10.2. >

Syntax

A > B

Input parameters

Parameter	Data type
A	INT
B	INT

Description

Returns TRUE if A is greater than B, and returns FALSE in other cases.

Example

- Input example

int1 (INT)	int2 (INT)	int3 (INT)
97	65	100

- Sample code

```
SELECT int1 as aa
FROM T1
WHERE int3 > int2;
```

- Output example

aa (INT)
97

3.1.4.10.3. >=

Syntax

```
A >= B
```

Input parameters

Parameter	Data type
A	INT
B	INT

Description

Returns TRUE if A is greater than or equal to B, and returns FALSE in other cases.

Example

- Input example

int1 (INT)	int2 (INT)	int3 (INT)
97	65	65
9	6	61

- Sample code

```
SELECT int1 as aa
FROM T1
WHERE int3 >= int2;
```

- Output example

aa (INT)
97
9

3.1.4.10.4. <

Syntax

A < B

Input parameters

Parameter	Data type
A	INT
B	INT

Description

Returns TRUE if A is smaller than B, and returns FALSE in other cases.

Example

- Input example

int1 (INT)	int2 (INT)	int3 (INT)
97	66	65
9	6	5

- Sample code

```
SELECT int1 as aa
FROM T1
WHERE int3 < int2;
```

- Output example

aa (INT)
97
9

3.1.4.10.5. <=

Syntax

A <= B

Input parameters

Parameter	Data type
A	INT

Parameter	Data type
B	INT

Description

Returns TRUE if A is smaller than or equal to B, and returns FALSE in other cases.

Example

- Input example

int1 (INT)	int2 (INT)	int3 (INT)
97	66	65
9	6	5

- Sample code

```
SELECT int1 as aa
FROM T1
WHERE int3 <= int2;
```

- Output example

aa (INT)
97
9

3.1.4.10.6. <>

Syntax

```
A <> B
```

Input parameters

Parameter	Data type
A	INT
B	INT

Description

Returns TRUE if A is not equal to B, and returns FALSE in other cases.

Example

- Input example

int1 (INT)	int2 (INT)	int3 (INT)
97	66	6

- Sample code

```
SELECT int1 as aa
FROM T1
WHERE int3 <> int2;
```

- Output example

aa (INT)
97

3.1.4.10.7. AND

Syntax

```
A AND B
```

Argument : Both A and B are of the BOOLEAN type.

Description

TRUE is returned if both A and B are TRUE. Otherwise, FALSE is returned.

Sample code

Input example

int1 (INT)	int2 (INT)	int3 (INT)
255	97	65

Sample code

```
SELECT int2 as aa
FROM T1
WHERE int1 = 255 AND int3 = 65;
```

Output example

aa (int)
97

3.1.4.10.8. BETWEEN AND

Syntax

```
A BETWEEN B AND C
```

Argument s: A, B, and C are values of the DOUBLE, BIGINT, INT, VARCHAR, DATE, TIMESTAMP, or TIME type.

Description

Use the BETWEEN operator to select a value that is between the two given argument values.

Example 1

Input example

int1 (INT)	int2 (INT)	int3 (INT)
90	80	100
11	10	7

Sample code

```
SELECT int1 as aa  
FROM T1  
WHERE int1 BETWEEN int2 AND int3;
```

Output example

aa(int)
90

Example 2

Input example

var1 (varchar)	var1 (varchar)	var1 (varchar)
b	a	c

Sample code

```
SELECT var1 as aa  
FROM T1  
WHERE var1 BETWEEN var2 AND var3;
```

Output example

aa (varchar)
b

Example 3

Input example

TIMESTAMP1 (TIMESTAMP)	TIMESTAMP2 (TIMESTAMP)	TIMESTAMP3 (TIMESTAMP)
1969-07-20 20:17:30	1969-07-20 20:17:20	1969-07-20 20:17:45

Sample code

```
SELECT TIMESTAMP1 as aa
FROM T1
WHERE TIMESTAMP1 BETWEEN TIMESTAMP2 AND TIMESTAMP3;
```

Output example

aa (TIMESTAMP)
1969-07-20 20:17:30

3.1.4.10.9. IS NOT FALSE

Syntax

```
A IS NOT FALSE
```

Argument: A of the BOOLEAN type

Description

TRUE is returned if A is true. FALSE is returned if A is false.

Examples

Input example

int1(INT)	int2(INT)
255	97

Sample code

```
SELECT int2 as aa
FROM T1
WHERE int1=255 IS NOT FALSE;
```

Output example

aa(int)
97

3.1.4.10.10. IS NOT NULL

Syntax

```
A IS NOT NULL
```

Arguments: A of the INT type

Description

TRUE is returned if A is null. Otherwise, FALSE is returned.

Examples

Input example

int1(INT)	int2(VARCHAR)
97	null
9	ww123

 **Note** Here, null indicates empty, rather than a null string.

Sample code

```
SELECT int1 as aa
FROM T1
WHERE int2 IS NOT NULL;
```

Output example

aa(int)
9

3.1.4.10.11. IS NOT TRUE

Syntax

```
A IS NOT TRUE
```

Argument : A specifies a condition of the BOOLEAN type.

Description

FALSE is returned if the condition is TRUE. TRUE is returned if the condition is FALSE.

Examples

Input example

int1 (INT)	int2 (INT)
255	97

Sample code

```
SELECT int2 as aa
FROM T1
WHERE int1 = 25 IS NOT TRUE;
```

Output example

aa (int)
97

3.1.4.10.12. IS NOT UNKNOWN

Syntax

```
A IS NOT UNKNOWN
```

Argument : A is of the BOOLEAN type.

Description

A is a logical comparison expression, such as $6 < 8$.

The value of A is TRUE or FALSE if a logical comparison between two numeric values is performed successfully. If a numeric value is compared with a non-numeric value, the logical comparison fails, and the value of A is neither TRUE nor FALSE. **IS NOT UNKNOWN** is used to check for such failures. It returns **FALSE** if the value of A is neither TRUE nor FALSE and returns **TRUE** if the value of A is TRUE or FALSE.

Examples

Input example

int1 (INT)	int2 (INT)
255	97

Sample code 1


```
SELECT int2 as aa  
FROM T1  
WHERE int1 = 25 IS NOT UNKNOWN;
```

Output example 1

aa (int)
97

Sample code 2

```
SELECT int2 as aa  
FROM T1  
WHERE int1 < null IS NOT UNKNOWN;
```

Output example 2

aa (int)
Null

3.1.4.10.13. IS NULL

Syntax

```
value IS NULL
```

Argument : value specifies a value of any type.

Description

TRUE is returned if A is null. Otherwise, FALSE is returned.

Examples

Input example

int1 (INT)	int2 (VARCHAR)
97	null
9	www

 **Note** In this case, null indicates empty, rather than a null string.

Sample code

```
SELECT int1 as aa
FROM T1
WHERE int2 IS NULL;
```

Output example

aa (int)
97

3.1.4.10.14. IS TRUE

Syntax

```
A IS TRUE
```

Argument: A of the BOOLEAN type

Description

TRUE is returned if A is TRUE. FALSE is returned if A is FALSE.

Examples

Input example

int1(INT)	int2(INT)
255	97

Sample code

```
SELECT int2 as aa
FROM T1
WHERE int1=255 IS TRUE;
```

Output example

aa(int)
97

3.1.4.10.15. IS UNKNOWN

Syntax

```
A IS UNKNOWN
```

Argument : A of the BOOLEAN type

Description

A is a logical comparison expression, such as 6 <> 8.

The value of A is TRUE or FALSE if a logical comparison between two numeric values is performed successfully. If a numeric value is compared with a non-numeric value, the logical comparison fails, and the value of A is neither TRUE nor FALSE. **IS UNKNOWN** is used to check for such failures. It returns **TRUE** if the value of A is neither TRUE nor FALSE and returns **FALSE** if the value of A is TRUE or FALSE.

Examples

Input example

int1(INT)	int2(INT)
255	97

Sample code 1

```
SELECT int2 as aa
FROM T1
WHERE int1=25 IS UNKNOWN;
```

Output example 1

aa(int)
Null

Sample code 2

```
SELECT int2 as aa
FROM T1
WHERE int1 > null IS UNKNOWN;
```

Output example 2

aa(int)
97

3.1.4.10.16. LIKE

Syntax

```
A LIKE B
```

Arguments : A and B of the VARCHAR type

Description

TRUE is returned if A matches B. Otherwise, FALSE is returned.

 **Note** The percent sign (%) can be used as a wildcard.

Examples

Sample code

int1(INT)	VARCHAR2(VARCHAR)	VARCHAR3(VARCHAR)
90	ss97	97ss
99	ss10	7ho7

Sample code 1

```
SELECT int1 as aa
FROM T1
WHERE VARCHAR2 LIKE 'ss%';
```

Output example 1

aa(int)
90
9

Sample code 2

```
SELECT int1 as aa
FROM T1
WHERE VARCHAR3 LIKE '%ss';
```

Output example 2

aa(int)
90

Sample code 3

```
SELECT int1 as aa
FROM T1
WHERE VARCHAR3 LIKE '%ho%';
```

Output example 3

aa(int)
99

3.1.4.10.17. NOT

Syntax

```
NOT A
```

Argument : A of the BOOLEAN type

Description

FALSE is returned if A is TRUE. TRUE is returned if A is FALSE.

Examples

Input example

int2(INT)	int3(INT)
97	65

Sample code

```
SELECT int2 as aa  
FROM T1  
WHERE NOT int3=62;
```

Output example

aa(int)
97

3.1.4.10.18. NOT BETWEEN AND

Syntax

```
A NOT BETWEEN B AND C
```

Arguments: A, B, and C can be of the DOUBLE, BIGINT, INT, VARCHAR, DATE, TIMESTAMP, or TIME types.

Description

The NOT BETWEEN operator is used to select a value that is not between the two given argument values.

Example 1

Input example

int1 (INT)	int2 (INT)	int3 (INT)
90	97	80
11	10	7

Sample code

```
SELECT int1 as aa
FROM T1
WHERE int1 NOT BETWEEN int2 AND int3;
```

Output example

aa (int)
11

Example 2

Input example

var1 (varchar)	var1 (varchar)	var1 (varchar)
d	a	c

Sample code

```
SELECT var1 as aa
FROM T1
WHERE var1 NOT BETWEEN var2 AND var3;
```

Output example

aa (varchar)
d

Example 3

Input example

TIMESTAMP1 (TIMESTAMP)	TIMESTAMP2 (TIMESTAMP)	TIMESTAMP3 (TIMESTAMP)
1969-07-20 20:17:30	1969-07-20 20:17:40	1969-07-20 20:17:45

Sample code

```
SELECT TIMESTAMP1 as aa
FROM T1
WHERE TIMESTAMP1 NOT BETWEEN TIMESTAMP2 AND TIMESTAMP3;
```

Output example

aa (TIMESTAMP)
1969-07-20 20:17:30

3.1.4.10.19. OR

Syntax

```
A OR B
```

Arguments: Both A and B are of the BOOLEAN type

Description

TRUE is returned if both A and B are TRUE. Otherwise, FALSE is returned.

Examples

Input example

int1 (INT)	int2 (INT)	int3 (INT)
255	97	65

Sample code

```
SELECT int2 as aa
FROM T1
WHERE int1 = 255 OR int3 = 65;
```

Output example

aa (int)
97

3.1.4.10.20. IN

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1,value2,...)
```

Arguments: Both value1 and value2 must be constants.

Description

Query records that match the arguments.

Examples

Input example

id (INT)	LastName (Varchar)
1	Adams
2	Bush
3	Carter

Sample code

```
SELECT * FROM T1
WHERE LastName IN ('Adams','Carter')
```

Output example

id (INT)	LastName (Varchar)
1	Adams
3	Carter

3.1.4.10.21. NOT IN

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name NOT IN (value1,value2,...)
```

Input parameters: value1 and value2, which must be constants

Description

Searches for records that do not match the values in the arguments.

Example

Input example

id (INT)	LastName (VARCHAR)
1	Adams

id (INT)	LastName (VARCHAR)
2	Bush
3	Carter

Sample code

```
SELECT * FROM T1
WHERE LastName NOT IN ('Adams','Carter')
```

Output example

id (INT)	LastName (VARCHAR)
2	Bush

3.1.4.10.22. IS DISTINCT FROM

Syntax

```
A IS DISTINCT FROM B
```

Arguments: A and B of generic types

Description

Check whether two arguments are equal. TRUE is returned if their data types or values are different. FALSE is returned if their data types and values are the same. If either or both arguments are null, they are considered to be the same.

Examples

Input example

A(Int)	B(Varchar)
97	97
null	sss
null	null

Sample code

```
SELECT
A IS DISTINCT FROM B as `result`
FROM T1
```

Output example

result(BOOLEAN)
true
true
false

3.1.4.10.23. IS NOT DISTINCT FROM

Syntax

```
A IS NOT DISTINCT FROM B
```

Arguments: A and B of generic types

Description

Check whether two arguments are equal. FALSE is returned if their data types or values are different. TRUE is returned if their data types and values are the same. If either or both arguments are null, they are considered to be the same.

Examples

Input example

A(Int)	B(Varchar)
97	97
null	sss
null	null

Sample code

```
SELECT  
A IS NOT DISTINCT FROM B as `result`  
FROM T1
```

Output example

result(BOOLEAN)
false
false
true

3.1.4.11. Built-in functions

3.1.4.11.1. String functions

3.1.4.11.1.1. Overview

CHAR_LENGTH

- Syntax: `CHAR_LENGTH(A)`
- Input parameter: The A parameter is of the VARCHAR data type.
- Description: Returns the number of characters in a string.

Example

Input example

var1 (VARCHAR)
ss
231ee

Sample code

```
SELECT CHAR_LENGTH(var1) as aa
FROM T1;
```

Output example

aa (INT)
2
5

ASCII2STR

- Syntax: `string ascii2str(int ascii)`
- Description: Converts a number into an ASCII character. If any argument is NULL, the return value is NULL.
- Input parameter: The value of the ascii parameter ranges from 1 to 127. If the value falls out of the range, an exception occurs.
- Example:

```
ascii2str(9) = '\t'
ascii2str(65) = 'A'
```

CONCAT

- Syntax: `string concat(string a, string b, ...)`
- Description: Concatenates all input strings and returns a new string. If any argument is NULL, the return value is NULL.
- Example:

```
concat('abc', 'def', 'gh') = 'abcdefgh'
concat('abc', NULL) = NULL
```

CONCAT_WS

- Syntax: `string concat_ws(string separator, string a, string b, ...)`
- Description: Concatenates all input strings. Different from the CONCAT function, this function adds a separator between two concatenated strings. If any argument is NULL, the return value is NULL.
- Example:

```
concat_ws(':', 'abc', 'def', 'gh') = 'abc:def:gh'
concat(':', 'abc', NULL) = NULL
```

SUBSTR

- Syntax

```
string substr(string a, int start)
string substr(string a, int start, int len)
```

- Description: Extracts a substring of the specified length from a string, starting from the specified position. If the length is not specified, a substring is extracted from the specified position to the end of the string. The value of the start parameter begins from 1. If the value is negative, a substring is extracted from the end of the string. If any argument is NULL, the return value is NULL.
- Example:

```
substr('galaxy_sql', 5) = 'xy_sql'
substr('galaxy_sql', -5) = 'y_sql'
substr('galaxy_sql', 5, 2) = 'xy'
```

LOWER

- Syntax

```
string lower(string a)
string lcase(string a)
```

- Description: Converts uppercase letters in a string into lowercase letters. If any argument is NULL, the return value is NULL.
- Example: `lower('fOoBaR') = 'foobar'`

UPPER

- Syntax

```
string upper(string a)
string ucase(string a)
```

- Description: Converts lowercase letters in a string into uppercase letters. If any argument is NULL, the return value is NULL.
- Example: `upper('fOoBaR') = 'FOOBAR'`

TRIM

- Syntax: `string trim(string a)`

- Description: Removes the leading and trailing spaces of a string. If any argument is NULL, the return value is NULL.
- Example: `trim(' foobar\t ') = 'foobar\t'`

LTRIM

- Syntax: `string ltrim(string a)`

- Description: Removes the leading spaces of a string. If any argument is NULL, the return value is NULL.
- Example: `ltrim(' foobar ') = 'foobar '`

RTRIM

- Syntax: `string rtrim(string a)`

- Description: Removes the trailing spaces of a string. If any argument is NULL, the return value is NULL.
- Example: `rtrim(' foobar ') = ' foobar'`

LPAD

- Syntax: `string lpad(string str, int len, string pad)`

- Description: Left-pads a string (str) with another string (pad), to the specified length. If any argument is NULL, the return value is NULL.
- Example:

```
lpad('hi', 5, '??') = '??? hi'
lpad('hi', 1, '??') = 'h'
lpad('---', 10, 'abc') = 'abcabca---'
```

RPAD

- Syntax: `string rpad(string str, int len, string pad)`

- Description: Right-pads a string (str) with another string (pad), to the specified length. If any argument is NULL, the return value is NULL.
- Example:

```
rpadd('hi', 5, '??') = 'hi???'  
rpadd('hi', 1, '??') = 'h'  
rpadd('---', 10, 'abc') = '---abcbca'
```

LENGTH

- Syntax: `int length(string str)`
- Description: Returns the length of a string. If any argument is NULL, the return value is NULL.
- Example:

```
length('hi') = 2  
length('I') = 1  
length(NULL) = NULL
```

REPEAT

- Syntax: `string repeat(string str, int n)`
- Description: Returns a new string that repeats the specified string for the specified number of times. If any argument is NULL, the return value is NULL.
- Example: `repeat('hi', 2) = 'hihi'`

REVERSE

- Syntax: `string reverse(string str)`
- Description: Reverses a string. If any argument is NULL, the return value is NULL.
- Example: `reverse('abc') = 'cba'`

SPLIT_EX

- Syntax: `string split_ex(string str, string sep, int index)`
- Description: Uses a separator to split a string into several segments and returns the segment identified by the index parameter. If the segment identified by the index parameter does not exist, NULL is returned. The value of the index parameter starts from 0. If any argument is NULL, the return value is NULL.
- Example:

```
split_ex('1.2.3.4', '.', 1) = '2'  
split_ex('1.2.3.4', '.', -1) = NULL  
split_ex('1.2.3.4', '.', 4) = NULL
```

REGEXP_REPLACE

- Syntax: `string regexp_replace(string str, string pattern, string replacement)`
- Description: Returns a string from the str string with its substrings that match the pattern regular expression replaced with the replacement string.

- Note: Backslashes (\) are used in a regular expression to escape special characters. If any argument is NULL, the return value is NULL.
- Example:

```
regexp_replace('100-200', '(\d+)', 'num') = 'num-num'  
regexp_replace('2014-03-13', '-', '') = '20140313'  
regexp_replace('foobar', 'oo|ar', '') = 'fb'
```

REGEXP_EXTRACT

- Syntax: `string regexp_extract(string str, string pattern, int index)`
- Description: Returns a substring of the str string that matches the pattern regular expression. Multiple substrings may all match the pattern. The index parameter identifies the substring to be returned. The value of the index parameter starts from 1.
- Note: A backslash (\) is used in a regular expression to escape special characters. If any argument is NULL, the return value is NULL.
- Example:

```
regexp_extract('foothebar', 'foo(. *?)( bar)', 2) = 'bar'  
regexp_extract('100-200', '(\d+)-(\d+)', 1) = '100'
```

INSTR

- Syntax: `int instr(string str, string substr)`
- Description: Returns the position where a substring first appears in a string. The return value starts from 1. If the specified substring does not exist in the string, 0 is returned. If any argument is NULL, the return value is NULL.
- Example:

```
instr('foobar', 'foo') = 1  
instr('foobar', 'abc') = 0
```

KEYVALUE

- Syntax: `string keyvalue(string str, string split1, string split2, string key_name)`
- Description: Parses key-value pairs from the specified string and returns the value that is mapped to the specified key. If the key does not exist, the return value is NULL. If any argument is NULL, the return value is NULL.
- Example:

```
keyvalue('k1=v1;k2=v2', ';', '=', 'k2') = 'v2'  
keyvalue('k1:v1,k2:v2', ',', ':', 'k3') = NULL
```

HASH

- Syntax: `int hash(string str)`
- Description: Returns the absolute value of the integer returned by the `hashCode()` function that takes the specified string as an input. If any argument is NULL, the return value is NULL.

MD5

- Syntax: `string md5(string str)`
- Description: Returns the MD5 value of a string. If any argument is NULL or an empty string, the return value is an empty string.

3.1.4.11.1.2. CHAR_LENGTH

Syntax

```
CHAR_LENGTH(A)
```

Input parameter

A: VARCHAR data type

Description

Returns the number of characters in a string.

Example

Input example

var1 (VARCHAR)
ss
231ee

Sample code

```
SELECT CHAR_LENGTH(var1) as aa
FROM T1;
```

Output example

aa (INT)
2
5

3.1.4.11.1.3. CHR

Syntax


```
VARCHAR CHR(INT ascii)
```

Arguments

ascii: specifies an integer of the INT type, ranging from 0 to 255. If the argument falls out of this range, null is returned.

Description

Convert ASCII codes into characters.

Examples

Input example

int1(INT)	int2(INT)	int3 (INT)
255	97	65

Sample code

```
SELECT CHR(int1) as var1, CHR(int2) as var2, CHR(int3) as var3  
FROM T1
```

Output example

var1(VARCHAR)	var2(VARCHAR)	var3(VARCHAR)
ÿ	a	A

3.1.4.11.1.4. CONCAT

Syntax

```
VARCHAR concat(VARCHAR var1, VARCHAR var2, ...)
```

Arguments

- var1: specifies a common string of the VARCHAR type.
- var2: specifies a common string of the VARCHAR type.

Description

Concatenate two or more strings into a single string. If any argument is NULL, the argument is skipped.

Examples

Input example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)
Hello	My	World
Hello	null	World
null	null	World
null	null	null

Sample code

```
SELECT CONCAT(var1, var2, var3) as var  
FROM T1
```

Output example

var (VARCHAR)
HelloMyWorld
HelloWorld
World
""

3.1.4.11.1.5. CONCAT_WS

Syntax

```
VARCHAR CONCAT_WS(VARCHAR separator, VARCHAR var1, VARCHAR var2, ...)
```

Arguments

- separator: specifies a separator of the VARCHAR type.
- var1: specifies a value of the VARCHAR type.
- var2: specifies a value of the VARCHAR type.

Description

Concatenate each argument's value with a separator to a new string and return the new string, whose length and type depend on the input values.

Note

- If the separator value is null, it is treated as an empty string for concatenating the argument's values.
- If any argument is null, the argument is skipped during concatenation.

Examples

Input example

sep (VARCHAR)	str1 (VARCHAR)	str2 (VARCHAR)	str3 (VARCHAR)
	Jack	Harry	John
null	Jack	Harry	John
	null	Harry	John
	Jack	null	null

Sample code

```
SELECT CONCAT_WS(sep, str1, str2, str3) as var
FROM T1
```

Output example

var (VARCHAR)
Jack Harry John
JackHarryJohn
Harry John
Jack

3.1.4.11.1.6. FROM_BASE64

Syntax

```
BINARY FROM_BASE64(str)
```

Arguments

str: specifies a Base64 string of the VARCHAR type.

Description

Parse a Base64 string into binary data.

Examples

Input example

a (INT)	b (BIGINT)	c (VARCHAR)
1	1L	null

Sample code

```
SELECT
from_base64(c) as var1,from_base64('SGVsbG8gd29ybGQ=') as var2
FROM T1
```

Output example

var1 (BINARY)	var2 (BINARY)
null	Byte Array: [72('H'), 101('e'), 108('l'), 108('l'), 111('o'), 32(' '), 119('w'), 111('o'), 114('r'), 108('l'), 100('d')]

3.1.4.11.1.7. HASH_CODE

Syntax

```
INT HASH_CODE(VARCHAR str)
```

Arguments

str: VARCHAR type

Description

Return the absolute value of a string on which hashCode() has been executed.

Examples

Input example

str1(VARCHAR)	str2(VARCHAR)	nullstr(VARCHAR)
k1=v1;k2=v2	k1:v1,k2:v2	null

Sample code

```
SELECT HASH_CODE(str1) as var1, HASH_CODE(str2) as var2, HASH_CODE(nullstr) as var3
FROM T1
```

Output example

var1(VARCHAR)	var2(VARCHAR)	var3(VARCHAR)
1099348823	401392878	null

3.1.4.11.1.8. INITCAP

Syntax

INITCAP(A)

Input parameter

A: VARCHAR data type

Description

Returns a string with the first letter of each word in uppercase and all other letters in lowercase.

Example

Input example

var1 (VARCHAR)
aADvbn

Sample code

```
SELECT INITCAP(var1) as aa
FROM T1;
```

Output example

aa (VARCHAR)
Aadvbn

3.1.4.11.1.9. JSON_VALUE

Syntax

VARCHAR JSON_VALUE(VARCHAR content, VARCHAR path)

Arguments

- content: specifies a JSON object to be parsed, which is represented as a JSON string of the VARCHAR type.
- path: specifies a path expression of the VARCHAR type to parse the JSON objects. The following table lists the path expressions that are currently supported. Path expressions supported

Symbol	Feature
\$	Root object
[]	Array subscript
*	Array wildcard

Symbol	Feature
.	Specified objects in a parent object

Description

Extract the value of the specified path from a JSON string. If the JSON string is invalid or any argument is NULL, the return value is NULL.

Examples

Input example

id (INT)	json(VARCHAR)	path(VARCHAR)
1	[10, 20, [30, 40]]	\$(2)[*]
2	{"aaa": "bbb", "ccc": {"ddd": "eee", "fff": "ggg", " + "hhh": ["h0", "h1", "h2"]}, "iii": "jjj"}	\$.ccc.hhh [*]
3	{"aaa": "bbb", "ccc": {"ddd": "eee", "fff": "ggg", " + "hhh": ["h0", "h1", "h2"]}, "iii": "jjj"}	\$.ccc.hhh [1]
4	[10, 20, [30, 40]]	NULL
5	NULL	\$(2) [*]
6	"{xx}"	"\$(2)[*]"

Sample code

```
SELECT
  id,
  JSON_VALUE(json, path) AS value
FROM
  T1
```

Output example

id (INT)	value (VARCHAR)
1	[30, 40]
2	["h0", "h1", "h2"]
3	H1
4	NULL
5	NULL

id (INT)	value (VARCHAR)
6	NULL

3.1.4.11.1.10. KEYVALUE

Syntax

```
VARCHAR KEYVALUE(VARCHAR str, VARCHAR split1, VARCHAR split2, VARCHAR key_name)
```

Arguments

- str: specifies a string of key-value pairs. This argument is of the VARCHAR type.
- split1: specifies a delimiter of the VARCHAR type to separate the key-value pairs.
- split2: specifies a delimiter of the VARCHAR type to separate the key and value in a key-value pair.
- key_name: specifies a key value of the VARCHAR type.

Description

Parse the key-value pairs in a string to locate a key-value pair that matches the key-value pair delimiter and key-value delimiter. Then return the key value corresponding to the key name in the key-value pair. If the key name does not exist or an exception occurs, the return value is NULL.

Examples

Input example

str (VARCHAR)	split1 (VARCHAR)	split2 (VARCHAR)	key1 (VARCHAR)
k1=v1; k2=v2	;	=	k2
null	;		:
k1:v1 k2:v2	null	=	:
k1:v1 k2:v2		=	null
k1:v1 k2:v2		=	:
k1:v1 k2:v2		=	:

Sample code

```
SELECT KEYVALUE(str, split1, split2, key1) as result  
FROM T1
```

Output example

result (VARCHAR)
v2
null
null
null
null
null

3.1.4.11.11. LOWER

Syntax

```
VARCHAR LOWER(A)
```

Argument

A: specifies a value of the VARCHAR type.

Description

Convert the specified string to a string of lowercase letters.

Examples

Input example

var1 (VARCHAR)
Ss
yyT

Sample code

```
SELECT LOWER(var1) as aa
FROM T1;
```

Output example

aa (VARCHAR)
ss
yyt

3.1.4.11.12. LPAD

Syntax

```
VARCHAR LPAD(VARCHAR str, INT len, VARCHAR pad)
```

Arguments

- **str**: specifies a source string of the VARCHAR type.
- **len**: specifies the length of a new string. This argument is of the INT type.
- **pad**: specifies a string to be repeatedly padded to the source string. This argument is of the VARCHAR type.

Description

Left pad a source string with another string several times until the new string reaches the specified length. If any argument is NULL, the return value is NULL. If the source string is an empty string or the target length is a negative number, the return value is NULL. If the string to be padded to the source string is an empty string, there are two scenarios. A trimmed source string is returned when the specified length is equal to or less than the length of the source string. NULL is returned when the specified length is greater than the length of the source string.

Examples

Input example

str (VARCHAR)	len (INT)	pad (VARCHAR)
""	-2	""
HelloWorld	15	John
John	2	C
C	4	HelloWorld
null	2	C
c	2	null
asd	2	""
""	2	s
asd	4	""
""	0	""

Sample code

```
SELECT LPAD(str, len, pad) AS result  
FROM T1
```

Output example

result (VARCHAR)
null
JohnHelloWorld
Jo
HelC
null
null
as
ss
null
""

3.1.4.11.1.3. MD5

Syntax

```
VARCHAR MD5(VARCHAR str)
```

Argument

str: specifies a string of the VARCHAR type.

Description

Return the MD5 value of a string. If the argument is NULL, the return value is NULL. If the argument is an empty string, the return value is an empty string.

Examples

Input example

str1 (VARCHAR)	str2 (VARCHAR)
k1 = v1; k2 = v2	null

Sample code

```
SELECT
  MD5(str1) as var1,
  MD5(str2) as var2
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)
19c17f42b4d6a90f7f9ffc2ea9bdd775	NA

3.1.4.11.1.14. OVERLAY

Syntax

```
VARCHAR OVERLAY ( ( VARCHAR x PLACING VARCHAR y FROM INT start_position [ FOR INT length ] ) )
```

Arguments

- x: VARCHAR type
- y: VARCHAR type
- start_position: INT type
- length (optional): INT type

Description

Replace a substring of x with y. The replacement starts from start_position and a total of length+1 characters are to be replaced.

Examples

Sample code

```
OVERLAY('abcdefg' PLACING 'hij' FROM 2 FOR 2) as result
FROM T1
```

Output example

result(VARCHAR)
ahijdefg

3.1.4.11.1.15. PARSE_URL

Syntax

```
VARCHAR PARSE_URL(VARCHAR urlStr, VARCHAR partToExtract [, VARCHAR key])
```

Arguments

- urlStr: specifies a URL string of the VARCHAR type.
- partToExtract: specifies a value of the VARCHAR type obtained after parsing.
- key: specifies an argument name of the VARCHAR type.

Description

Parse a URL to obtain the value of partToExtract, such as QUERY. The options for partToExtract include HOST, PATH, QUERY, REF, PROTOCOL, FILE, AUTHORITY, and USERINFO.

 **Note** If the partToExtract value is NULL, NULL is returned.

Examples

Input example

url1(VARCHAR)	nullstr(VARCHAR)
http://facebook.com/path/p1.php?query=1	null

Sample code

```
SELECT PARSE_URL(url1, 'QUERY', 'query') as var1,
       PARSE_URL(url1, 'QUERY') as var2,
       PARSE_URL(url1, 'HOST') as var3,
       PARSE_URL(url1, 'PATH') as var4,
       PARSE_URL(url1, 'REF') as var5,
       PARSE_URL(url1, 'PROTOCOL') as var6,
       PARSE_URL(url1, 'FILE') as var7,
       PARSE_URL(url1, 'AUTHORITY') as var8,
       PARSE_URL(nullstr, 'QUERY') as var9,
       PARSE_URL(url1, 'USERINFO') as var10,
       PARSE_URL(nullstr, 'QUERY', 'query') as var11
FROM T1
```

Output example

var1(VARCHAR)	var2(VARCHAR)	var3(VARCHAR)	var4(VARCHAR)	var5(VARCHAR)	var6(VARCHAR)	var7(VARCHAR)	var8(VARCHAR)	var9(VARCHAR)	var10(VARCHAR)	var11(VARCHAR)
1	query=1	facebook.com	/path/p1.php	null	http	/path/p1.php?query=1	facebook.com	null	null	null

3.1.4.11.1.6. POSITION

Syntax

INTEGER POSITION(x IN y)

Arguments

- x: specifies a value of the VARCHAR type.
- y: specifies a value of the VARCHAR type.

Description

Return the position of the first occurrence of string x within string y. The function is similar to LOCATE(substr,str).

Examples

Sample code

```
POSITION('in' IN 'china') as result
FROM T1
```

Output example

result (INT)
3

3.1.4.11.17. REGEXP

Syntax

BOOLEAN REGEXP(VARCHAR str, VARCHAR pattern)

Arguments

- str: specifies a string of the VARCHAR type.
- pattern: specifies a pattern of the VARCHAR type.

Description

Perform regular expression matching on a string to check whether it matches the specified pattern. If the string or pattern is empty or null, FALSE is returned.

Examples

Input example

str1(VARCHAR)	pattern1 (VARCHAR)
k1=v1;k2=v2	k2*
k1:v1 k2:v2	k3

str1(VARCHAR)	pattern1 (VARCHAR)
null	k3
k1:v1 k2:v2	null
k1:v1 k2:v2	(

Sample code

```
SELECT REGEXP(str1, pattern1) as result
FROM T1
```

Output example

result(BOOLEAN)
true
false
null
null
false

3.1.4.11.1.18. REGEXP_EXTRACT

Syntax

```
VARCHAR REGEXP_EXTRACT(VARCHAR str, VARCHAR pattern, INT index)
```

Arguments

- **str**: specifies a source string of the VARCHAR type.
- **pattern**: specifies the regular expression pattern used to match the string to be extracted. This argument is of the VARCHAR type.
- **index**: specifies the index number of the substring to be extracted from the source string. This argument is of the INT type.

 **Note** Specify constants in the regular expression with Java format because SQL string constants are completely compiled to Java code. For example, you can specify a digit (0-9) in the regular expression as `\d`, which is the same as the digit in a regular expression in Java code.

Description

Use the regular expression pattern to match and extract the indexed substring from a source string. The index starts from 1. If any argument is NULL or the regular expression is invalid, the return value is NULL.

Examples

Input example

str1 (VARCHAR)	pattern1 (VARCHAR)	index1 (INT)
foothebar	foo(. ?)(bar)	2
100-200	(\d+)-(\d+)	1
null	foo(. ?)(bar)	2
foothebar	null	2
foothebar	""	2
foothebar	(	2

Sample code

```
SELECT REGEXP_EXTRACT(str1, pattern1, index1) as result  
FROM T1
```

Output example

result (VARCHAR)
bar
100
null
null
null
null

3.1.4.11.19. REGEXP_REPLACE

Syntax

```
VARCHAR REGEXP_REPLACE(VARCHAR str, VARCHAR pattern, VARCHAR replacement)
```

Arguments

- str: specifies a string of the VARCHAR type.

- **pattern**: specifies the regular expression pattern used to match the string to be replaced. This argument is of the VARCHAR type.
- **replacement**: specifies the string used to replace others. This argument is of the VARCHAR type.

 **Note** Specify constants in the regular expression with Java format because SQL string constants are completely compiled to Java code. For example, you can specify a digit (0-9) in the regular expression as `\d`, which is the same as the digit in a regular expression in Java code.

Description

Replace a substring that matches the regular expression pattern in a source string with a specified string, and return a new string. If any argument is NULL or the regular expression is invalid, the return value is NULL.

Examples

Input example

str1 (VARCHAR)	pattern1 (VARCHAR)	replace1 (VARCHAR)
2014-03-13	-	""
null	-	""
2014-03-13	-	null
2014-03-13	""	s
2014-03-13	(	s
100-200	(\d+)	num

Sample code

```
SELECT REGEXP_REPLACE(str1, pattern1, replace1) as result
FROM T1
```

Output example

result (VARCHAR)
20140313
null
null
2014-03-13
null
num-num

3.1.4.11.1.20. REPEAT

Syntax

```
VARCHAR REPEAT(VARCHAR str, INT n)
```

Arguments

- str: specifies the string to be repeated. This argument is of the VARCHAR type.
- n: specifies the number of repetitions. This argument is of the INT type.

Description

Return a new string consisting of N repetitions of the specified string. If any argument is NULL, the return value is NULL. If the value of n is 0 or a negative number, the return value is an empty string.

Examples

Input example

str(VARCHAR)	n(INT)
J	9
Hello	2
Hello	-9
null	9

Sample code

```
SELECT REPEAT(str,n) as var1  
FROM T1
```

Output example

result(VARCHAR)
JJJJJJJJ
HelloHello
""
null

3.1.4.11.1.21. REVERSE

Syntax

VARCHAR REVERSE(VARCHAR str)

Arguments

str: specifies a string of the VARCHAR type.

Description

Return a string in the reverse order of the specified string. If the argument is NULL, the return value is NULL.

Examples

Input example

str1(VARCHAR)	str2(VARCHAR)	str3 (VARCHAR)	str4 (VARCHAR)
iPhoneX	Alibaba	World	null

Sample code

```
SELECT REVERSE(str1) as var1,REVERSE(str2) as var2,  
       REVERSE(str3) as var3,REVERSE(str4) as var4  
FROM T1
```

Output example

var1(VARCHAR)	var2(VARCHAR)	var3(VARCHAR)	var4(VARCHAR)
XenohPi	ababilA	dlroW	null

3.1.4.11.1.22. RPAD

Syntax

VARCHAR RPAD(VARCHAR str, INT len, VARCHAR pad)

Arguments

- str: specifies a source string of the VARCHAR type.
- len: specifies the length of a new string. This argument is of the INT type.
- pad: specifies a string to be repeatedly padded to the source string. This argument is of the VARCHAR type.

Description

Right pad a string with another string several times until the new string reaches the specified length. If any argument is NULL, the return value is NULL. If the source string is an empty string or the specified length is a negative number, the return value is NULL. If the string to be padded to the source string is an empty string, there are two scenarios. A trimmed source string is returned when the specified length is equal to or less than the length of the source string. NULL is returned when the specified length is greater than the length of the source string.

Examples

Input example

str (VARCHAR)	len (INT)	pad (VARCHAR)
''	-2	''
HelloWorld	15	John
John	2	C
C	4	HelloWorld
null	2	C
c	2	null
asd	2	''
''	2	s
asd	4	''
''	0	''

Sample code

```
SELECT RPAD(str, len, pad) as result  
FROM T1
```

Output example

result (VARCHAR)
null
HelloWorldJohnj
Jo
CHel
null
null

result (VARCHAR)
as
ss
null
""

3.1.4.11.1.23. SPLIT_INDEX

Syntax

```
VARCHAR SPLIT_INDEX(VARCHAR str, VARCHAR sep, INT index)
```

Arguments

- str: specifies the string to be split. This argument is of the VARCHAR type.
- sep: specifies the separator of the VARCHAR type.
- index: specifies the index number of the substring to be obtained from the split string. This argument is of the INT type.

Description

Use a separator to split a string into several substrings and return the indexed substring. If no substring is indexed, NULL is returned. The index starts from 0. If any argument is NULL, the return value is NULL.

Examples

Input example

str (VARCHAR)	sep (VARCHAR)	index (INT)
Jack,John,Mary	,	2
Jack,John,Mary	,	3
Jack,John,Mary	null	0
null	,	0

Sample code

```
SELECT SPLIT_INDEX(str, sep, index) as var1  
FROM T1
```

Output example

var1 (VARCHAR)
Mary
null
null
null

3.1.4.11.1.24. STR_TO_MAP

Syntax

```
MAP STR_TO_MAP(VARCHAR text)
MAP STR_TO_MAP(VARCHAR text, VARCHAR listDelimiter, VARCHAR keyValueDelimiter)
```

Arguments

- text: specifies given text of the VARCHAR type.
- listDelimiter: specifies the delimiter used to split the given text into key-value pairs. This argument is of the VARCHAR type. The default delimiter is a comma (,).
- keyValueDelimiter: specifies the delimiter used to separate the key and value in a key-value pair. This argument is of the VARCHAR type. The default delimiter is an equal sign (=).

 **Note** The delimiters are defined by Java regular expressions. If a special character is used, it needs to be converted to an escape character.

Description

Use listDelimiter to split the given text into key-value pairs, use keyValueDelimiter to separate the key and value in each key-value pair, and then generate and return a map.

Examples

Sample code

```
SELECT
  STR_TO_MAP('k1 = v1, k2 = v2') ['k1'] as a
FROM T1
```

Output example

a (VARCHAR)
v1

3.1.4.11.1.25. SUBSTRING

Syntax

```
VARCHAR SUBSTRING(VARCHAR a, INT start)
VARCHAR SUBSTRING(VARCHAR a, INT start, INT len)
```

Arguments

- **a**: specifies the string from which a substring is to be obtained. This argument is of the VARCHAR type.
- **len**: specifies the length of the substring to be obtained. This argument is of the INT type.
- **start**: specifies the start position where the substring is to be obtained from the specified string. This argument is of the INT type.

Description

Obtain a substring of the specified length from a string, starting from the specified position. If the length is not specified, the entire string is obtained. start begins with 1. If the value is 0, it is regarded as 1. If the value is negative, obtain a substring of the specified length backwards starting from the last character in the string.

Examples

Input example

str (VARCHAR)	nullstr (VARCHAR)
k1 = v1; k2 = v2	null

Sample code

```
SELECT SUBSTRING(' ', 222222222) as var1,
       SUBSTRING(str, 2) as var2,
       SUBSTRING(str, -2) as var3,
       SUBSTRING(str, -2, 1) as var4,
       SUBSTRING(str, 2, 1) as var5,
       SUBSTRING(str, 22) as var6,
       SUBSTRING(str, -22) as var7,
       SUBSTRING(str, 1) as var8,
       SUBSTRING(str, 0) as var9,
       SUBSTRING(nullstr, 0) as var10
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)	var4 (VARCHAR)	var5 (VARCHAR)	var6 (VARCHAR)	var7 (VARCHAR)	var8 (VARCHAR)	var9 (VARCHAR)	var10 (VARCHAR)
' '	k1 = v1; k2 = v2	v2	v	1	' '	' '	k1 = v1; k2 = v2	k1 = v1; k2 = v2	null

3.1.4.11.1.26. TO_BASE64

Syntax

```
VARCHAR TO_BASE64(bin)
```

Argument

bin: specifies a value of the BINARY type.

Description

Convert the input value of the BINARY type to a Base64 string.

Examples

Input example

c (VARCHAR)
SGVsbG8gd29ybGQ =
SGk =
SGVsbG8 =

Sample code

```
SELECT TO_BASE64(FROM_BASE64(c)) as var1  
FROM T1
```

Output example

var1 (VARCHAR)
SGVsbG8gd29ybGQ =
SGk =
SGVsbG8 =

3.1.4.11.1.27. TRIM

Syntax

```
VARCHAR TRIM( VARCHAR x )
```

Arguments

x: VARCHAR type

Description

Trim leading or trailing characters from a string. The most common use is to trim leading or trailing blank spaces.

Examples

Test example

```
SELECT TRIM(' Sample '); as result  
FROM T1
```

Output example

result(VARCHAR)
Sample

3.1.4.11.1.28. UPPER

Syntax

```
VARCHAR UPPER(A)
```

Arguments

A: VARCHAR type

Description

Convert the specified string to a string of uppercase letters.

Examples

Input example

var1(VARCHAR)
ss
ttee

Sample code

```
SELECT UPPER(var1) as aa  
FROM T1;
```

Output example

aa(VARCHAR)
SS
TTEE

3.1.4.11.2. Mathematical unctons

3.1.4.11.2.1. Addition

Syntax

A + B

Arguments

A and B: INT type

Description

Return the result of adding B to A.

Examples

Input example

int1(INT)	int2(INT)	int3(INT)
10	20	30

Sample code

<pre>SELECT int1+int2+int3 as aa FROM T1;</pre>

Output example

aa(INT)
60

3.1.4.11.2.2. Subtraction

Syntax

A - B

Arguments

A and B: specify values of the INT type.

Description

Return the result of subtracting B from A.

Examples

Input example

int1 (INT)	int2 (INT)	int3 (INT)
10	10	30

Sample code

```
SELECT int3 - int2 - int1 as aa
FROM T1;
```

Output example

aa (INT)
10

3.1.4.11.2.3. Multiplication

Syntax

```
A * B
```

Arguments

A and B: specify values of the INT type.

Description

Return the result of multiplying A by B.

Examples

Input example

int1 (INT)	int2 (INT)	int3 (INT)
10	20	3

Sample code

```
SELECT int1*int2*int3 as aa
FROM T1;
```

Output example

aa (INT)
600

3.1.4.11.2.4. Division

Syntax

A/B

Arguments

A and B: INT type

Description

Return the result of dividing A by B.

Examples

Input example

int1 (INT)	int2 (INT)
8	4

Sample code

<pre>SELECT int1 / int2 as aa FROM T1;</pre>

Output example

aa(INT)
2

3.1.4.11.2.5. ABS

Syntax

DOUBLE ABS(A)

Arguments

A: DOUBLE type

Description

Return the absolute value of the specified number.

Examples

Input example

int1(DOUBLE)
4.3

Sample code

```
SELECT ABS(in1) as aa
FROM T1;
```

Output example

aa(DOUBLE)
4.3

3.1.4.11.2.6. ACOS

Syntax

```
DOUBLE ACOS(A)
```

Argument

A: specifies a number of DOUBLE type.

Description

Return the arccos value of the specified number.

Examples

Input example

int1 (DOUBLE)
0.7173560908995228
0.4

Sample code

```
SELECT ACOS(in1) as aa
FROM T1;
```

Output example

aa (DOUBLE)
0.7707963267948966
1.1592794807274085

3.1.4.11.2.7. BIN

Syntax

```
VARCHAR BIN(BIGINT number)
```

Arguments

number: BIGINT type. INT type arguments support implicit conversion, whereas other types of arguments do not.

Description

Convert a value of the BIGINT type to a binary string.

Examples

Input example

id(INT)	x (BIGINT)
1	12L
2	10L
3	0L
4	10000000000L

Sample code

```
SELECT id, bin(x) as var1
FROM T1
```

Output example

id(INT)	var1(VARCHAR)
1	1100
2	1010
3	0
4	1001010100000010111110010000000000

3.1.4.11.2.8. ASIN

Syntax

```
DOUBLE ASIN(A)
```

Arguments

A: DOUBLE type

Description

Return the arcsine value of the specified number.

Examples

Input example

int1 (DOUBLE)
0.7173560908995228
0.4

Sample code

```
SELECT ASIN(in1) as aa
FROM T1;
```

Output example

aa (DOUBLE)
0.8
0.41151684606748806

3.1.4.11.2.9. ATAN

Syntax

```
DOUBLE ATAN(A)
```

Arguments

A: DOUBLE type

Description

Return the arctangent value of the specified number.

Examples

Input example

int1 (DOUBLE)
0.7173560908995228
0.4

Sample code

```
SSELECT ATAN(in1) as aa  
FROM T1;
```

Output example

aa (DOUBLE)
0.6222796222326533
0.3805063771123649

3.1.4.11.2.10. BITAND

Syntax

```
INT BITAND(INT number1,INT number2)
```

Arguments

- number1: specifies a common argument of the INT type.
- number2: specifies a common argument of the INT type.

Description

Perform a bitwise **AND** operation on the specified values. The input and output values are both of the INT type.

Examples

Input example

a(INT)	b(INT)
2	3

Sample code

```
SELECT BITAND(a, b) as intt  
FROM T1
```

Output example

intt(INT)
2

3.1.4.11.2.11. BITNOT

Syntax

```
INT BITNOT((INT number)
```

Arguments

number: specifies a common argument of the INT type.

Description

Perform a bitwise NOT operation on the specified value. The input and output values are both integers.

Examples

Input example

a(INT)
7

Sample code

```
SELECT BITNOT(a) as var1,  
FROM T1
```

Output example

var1(INT)
0xffff8

3.1.4.11.2.12. BITOR

Syntax

```
INT BITOR(INT number1, INT number2)
```

Arguments

- number1: specifies a common argument of the INT type.
- number2: specifies a common argument of the INT type.

Description

Perform a bitwise OR operation on the specified values. The input and output values are both of the INT type.

Examples

Input example

a(INT)	b(INT)
2	3

Sample code

```
SELECT BITOR(a, b) as var1,  
FROM T1
```

Output example

var1(INT)
3

3.1.4.11.2.13. BITXOR

Syntax

```
INT BITXOR(INT number1, INT number2)
```

Arguments

- number1: specifies a common value of the INT type.
- number2: specifies a common value of the INT type.

Description

Return the bitwise XOR (exclusive OR) of two given numbers. The argument and result are both of the INT type.

Examples

Input example

a (INT)	b (INT)
2	3

Sample code

```
SELECT BITXOR(a, b) as var1,
FROM T1
```

Output example

var1 (INT)
1

3.1.4.11.2.14. CARDINALITY

Syntax

```
CARDINALITY(str)
```

Arguments

str: specifies an array.

Description

Return the number of elements in an array.

Examples

Input example

str(INT)	b (Array[INT])
1	Array(1, 2, 3)

Sample code

```
SELECT cardinality(b) AS result
FROM T1
```

Output example

result(INT)
3

3.1.4.11.2.15. COS

Syntax

```
DOUBLE COS(A)
```

Arguments

A: DOUBLE type

Description

Return the cosine value of the specified number.

Examples

Input example

in1(DOUBLE)
0.8
0.4

Sample code

```
SELECT COS(in1) as aa
FROM T1;
```

Output example

aa(DOUBLE)
0.6967067093471654
0.9210609940028851

3.1.4.11.2.16. COT

Syntax

```
DOUBLE COT(A)
```

Arguments

A: specifies a value of the DOUBLE type.

Description

Return the cotangent value of the specified number.

Examples

Input example

in1 (DOUBLE)
0.8

in1 (DOUBLE)
0.4

Sample code

```
SELECT COT(in1) as aa
FROM T1;
```

Output example

aa (DOUBLE)
1.0296385570503641
0.4227932187381618

3.1.4.11.2.17. E

Syntax

```
DOUBLE E()
```

Arguments

DOUBLE type

Description

Return the DOUBLE type value of natural constant E.

Examples

Input example

id(INT)	X(DOUBLE)
1	8.0

Sample code

```
SELECT id, e() as dou1, E() as dou2
FROM T1
```

Output example

id(INT)	dou1(DOUBLE)	dou2 (DOUBLE)
1	2.718281828459045	2.718281828459045

3.1.4.11.2.18. EXP

Syntax

```
DOUBLE EXP(A)
```

Argument

A: specifies a value of the DOUBLE type.

Description

Return the power of e, where e is the base of the natural logarithm.

Examples

Input example

in1 (DOUBLE)
8.0
10.0

Sample code

```
SELECT EXP(in1) as aa  
FROM T1;
```

Output example

aa (DOUBLE)
2980.9579870417283
22026.465794806718

3.1.4.11.2.19. FLOOR

Syntax

```
DOUBLE FLOOR(A)
```

Arguments

A: DOUBLE type

Description

Return the maximum integer that is less than or equal to the provided number.

Examples

Input example

in1(DOUBLE)
8.123
10.321

Sample code

```
SELECT FLOOR(in1) as aa
FROM T1;
```

Output example

aa(INT)
8
10

3.1.4.11.2.20. LN

Syntax

```
DOUBLE ln(DOUBLE number)
```

Arguments

number: specifies a number of the DOUBLE type. If the input value is of the VARCHAR or BIGINT type, it is implicitly converted to the DOUBLE type before an operation.

Description

Return the natural logarithm of the specified number. The return value is a logarithm of the DOUBLE type.

Examples

Input example

ID (INT)	X (DOUBLE)
1	100.0
2	8.0

Sample code

```
SELECT id, ln(x) as dou1, ln(e()) as dou2
FROM T1
```

Output example

ID (INT)	dou1 (DOUBLE)	dou2 (DOUBLE)
1	4.605170185988092	1.0
2	2.0794415416798357	1.0

3.1.4.11.2.21. LOG

Syntax

```
DOUBLE LOG(DOUBLE base, DOUBLE x)
DOUBLE LOG(DOUBLE x)
```

Arguments

- base: specifies a value of the DOUBLE type.
- x: specifies a value of the DOUBLE type.

Description

Return the natural logarithm of x to the base specified by the base argument. The return value is a logarithm of the DOUBLE type. If there is only one argument value, the return value is a natural logarithm to the base specified by the argument.

Examples

Input example

ID (INT)	BASE (DOUBLE)	X (DOUBLE)
1	10.0	100.0
2	2.0	8.0

Sample code

```
SELECT id, LOG(base, x) as dou1, LOG(2) as dou2
FROM T1
```

Output example

ID (INT)	dou1 (DOUBLE)	dou2 (DOUBLE)
1	2.0	0.6931471805599453

ID (INT)	dou1 (DOUBLE)	dou2 (DOUBLE)
2	3.0	0.6931471805599453

3.1.4.11.2.22. LOG10

Syntax

```
DOUBLE LOG10(DOUBLE x)
```

Argument

x: specifies a value of the DOUBLE type.

Description

Return the base-10 logarithm of x. If x is NULL, the return value is NULL. If x is a negative number, an exception occurs.

Examples

Input example

id (INT)	X (INT)
1	100
2	10

Sample code

```
SELECT id, log10(x) as dou1  
FROM T1
```

Output example

ID (INT)	dou1 (DOUBLE)
1	2.0
2	1.0

3.1.4.11.2.23. LOG2

Syntax

```
DOUBLE LOG2(DOUBLE x)
```

Arguments

x: specifies a multiple of 2. This argument is of the DOUBLE type.

Description

Return the base-2 logarithm of x. If x is NULL, the return value is NULL. If x is a negative number, an exception occurs.

Examples

Input example

id (INT)	X (INT)
1	8
2	2

Sample code

```
SELECT id, log2(x) as dou1  
FROM T1
```

Output example

ID (INT)	dou1 (DOUBLE)
1	3.0
2	1.0

3.1.4.11.2.24. PI

Syntax

```
DOUBLE PI()
```

Description

Obtain the ratio of circumference to diameter, namely, Pi.

Examples

Input example

id (INT)	X (INT)
1	8

Sample code

```
SELECT id, PI() as dou1  
FROM T1
```

Output example

ID (INT)	dou1 (DOUBLE)
1	3.141592653589793

3.1.4.11.2.25. POWER

Syntax

```
DOUBLE POWER(A, B)
```

Input parameters

A and B: DOUBLE data type

Description

Returns the result of raising A to the power of B.

Example

Input example

int1 (DOUBLE)	in2 (DOUBLE)
2.0	4.0

Sample code

```
SELECT POWER(in1, in2) as aa  
FROM T1;
```

Output example

aa (DOUBLE)
16.0

3.1.4.11.2.26. RAND

Syntax

```
DOUBLE RAND([BIGINT seed])
```

Arguments

seed: specifies a random number of the BIGINT type. This argument specifies the start number of a random number sequence.

Description

Return a random number of the DOUBLE type. The return value range is `[0, 1)`.

Examples

Input example

id(INT)	X(INT)
1	8

Sample code

```
SELECT id, rand(1) as dou1, rand(3) as dou2
FROM T1
```

Output example

id(INT)	dou1(DOUBLE)	dou2(DOUBLE)
1	0.7308781907032909	0.731057369148862

3.1.4.11.2.27. SIN

Syntax

```
DOUBLE SIN(A)
```

Arguments

A: specifies a number of the DOUBLE type.

Description

Return the sine value of the specified number.

Examples

Input example

in1 (DOUBLE)
8.0
0.4

Sample code

```
SELECT SIN(in1) as aa
FROM T1;
```

Output example

aa (DOUBLE)
0.9893582466233818
0.3894183423086505

3.1.4.11.2.28. SQRT

Syntax

```
DOUBLE SQRT(A)
```

Arguments

A: specifies a number of the DOUBLE type.

Description

Return the square root of the specified number.

Examples

Input example

in1 (DOUBLE)
8.0

Sample code

```
SELECT SQRT(in1) as aa  
FROM T1;
```

Output example

aa (DOUBLE)
2.8284271247461903

3.1.4.11.2.29. TAN

Syntax

```
DOUBLE TAN(A)
```

Arguments

A: DOUBLE type

Description

Return the tangent value of the specified number.

Examples

Input example

in1(DOUBLE)
8.0
0.4

Sample code

```
SELECT TAN(in1) as aa  
FROM T1;
```

Output example

aa(DOUBLE)
1.0296385570503641
0.4227932187381618

3.1.4.11.2.30. CEIL

Syntax

```
value_class CEIL(value)
```

Arguments

value: specifies a value of the INT, DOUBLE, or BIGINT type.

Description

Return the smallest integer greater than or equal to a specified value. The value_class data type returned by CEIL is the same as that of the argument.

Examples

Input example

ID (INT)	value1 (INT)	value2 (DOUBLE)
1	3	3.6

Sample code

```
SELECT CEIL (value2) as result1
SELECT CEIL (value2) as result2
FROM T1
```

Output example

ID (INT)	result1 (INT)	result2 (DOUBLE)
1	3	4.0

3.1.4.11.2.31. CHARACTER_LENGTH

Syntax

```
INTEGER CHARACTER_LENGTH( VARCHAR x )
```

Arguments

x: specifies a value of the VARCHAR type.

Description

Return the number of characters contained in a string.

Examples

Input example

ID (INT)	X (VARCHAR)
1	StreamCompute

Sample code

```
SELECT CHARACTER_LENGTH( x ) as result
FROM T1
```

Output example

ID (INT)	result (INT)
1	13

3.1.4.11.2.32. DEGREES

Syntax

```
DOUBLE DEGREES(double x)
```

Arguments

x: specifies a radian value of the DOUBLE type.

Description

Return the result of converting radians to degrees.

Examples

Sample code

```
SELECT DEGREES( PI() ) as result
FROM T1
```

Output example

result(DOUBLE)
180.0

3.1.4.11.2.33. MOD

Syntax

```
Integer MOD(Integer x,Integer y)
```

Arguments

- x: specifies a value of the Integer type.
- y: specifies a value of the Integer type.

Description

Return the remainder when integer x is divided by integer y, without considering the quotient. When x is a negative integer or both x and y are negative integers, the result is negative.

Examples

Input example

X (INT)	Y (INT)
29	3
-29	3
-29	-3

Sample code

```
SELECT MOD(x,y) as result  
FROM T1
```

Output example

ID (INT)	result (INT)
1	2
2	-2
3	-2

3.1.4.11.2.34. ROUND

Syntax

```
DECIMAL ROUND(DECIMAL x, INT n)
```

Argument

x: DECIMAL type

Description

Round the x argument's value to n decimal places.

Examples

Input example

in1 (DECIMAL)
0.7173560908995228
0.4

Sample code

```
SELECT ROUND(in1,2) as `result`  
FROM T1;
```

Output example

result (DECIMAL)
0.72
0.40

3.1.4.11.3. Date functions

3.1.4.11.3.1. CURRENT_TIMESTAMP

Syntax

```
CURRENT_TIMESTAMP
```

Arguments

None

Description

Return the current UTC (GMT+0) timestamp, which is eight hours earlier than Beijing time.

Examples

Sample code

```
SELECT CURRENT_TIMESTAMP as var1  
FROM T1
```

Output example

var1 (TIMESTAMP)
2007-04-30 13:10:02.047

3.1.4.11.3.2. DATEDIFF

Syntax

```
INT DATEDIFF(VARCHAR enddate, VARCHAR startdate)  
INT DATEDIFF(TIMESTAMP enddate, VARCHAR startdate)  
INT DATEDIFF(VARCHAR enddate, TIMESTAMP startdate)  
INT DATEDIFF(TIMESTAMP enddate, TIMESTAMP startdate)
```

Arguments

- startdate: specifies a start date of the TIMESTAMP or VARCHAR type. The format of a VARCHAR type date is yyyy-MM-dd or yyyy-MM-dd HH:mm:ss.
- enddate: specifies an end date of the TIMESTAMP or VARCHAR type. The format of a VARCHAR type date is yyyy-MM-dd or yyyy-MM-dd HH:mm:ss.

Description

Compute the number of days between the specified start date and end date. The return value is an integer of the `TIMESTAMP` type or in the format of `yy-MM-dd HH:mm:ss` or `yy-MM-dd`. If any argument is `NULL` or a parsing error occurs, the return value is `NULL`.

Examples

Input example

<code>datetime1(VARCHAR)</code>	<code>datetime2 (VARCHAR)</code>	<code>nullstr(VARCHAR)</code>
2017-10-15 00:00:00	2017-09-15 00:00:00	null

Sample code

```
SELECT DATEDIFF(datetime1, datetime2) as int1,
       DATEDIFF(TIMESTAMP '2017-10-15 23:00:00',datetime2) as int2,
       DATEDIFF(datetime2,TIMESTAMP '2017-10-15 23:00:00') as int3,
       DATEDIFF(datetime2,nullstr) as int4,
       DATEDIFF(nullstr,TIMESTAMP '2017-10-15 23:00:00') as int5,
       DATEDIFF(nullstr,datetime2) as int6,
       DATEDIFF(TIMESTAMP '2017-10-15 23:00:00',TIMESTAMP '2017-9-15 00:00:00')as int7
FROM T1
```

Output example

<code>int1(INT)</code>	<code>int2(INT)</code>	<code>int3(INT)</code>	<code>int4(INT)</code>	<code>int5(INT)</code>	<code>int6(INT)</code>	<code>int7 (INT)</code>
30	31	-31	null	null	null	31

3.1.4.11.3.3. DATE_ADD

Syntax

```
VARCHAR DATE_ADD(VARCHAR startdate, INT days)
VARCHAR DATE_ADD(TIMESTAMP time, INT days)
```

Arguments

- `startdate`: specifies a date of the `VARCHAR` type. The format is `yyyy-MM-dd` or `yyyy-MM-dd HH:mm:ss`.
- `time`: specifies a date of the `TIMESTAMP` type.
- `days`: specifies the number of days of the `INT` type.

Description

Return a date that is X (specified by days) days after the specified date of the `VARCHAR` type. The date format can be `yyyy-MM-dd hh:mm:ss`, `yyyy-MM-dd`, and timestamp string. The return value is a string type date in the format of `yyyy-MM-dd`. If any argument is `NULL` or a parsing error occurs, the return value is `NULL`.

Examples

Input example

datetime1 (VARCHAR)	nullstr (VARCHAR)
2017-09-15 00:00:00	null

Sample code

```
SELECT DATE_ADD(datetime1, 30) as var1,  
       DATE_ADD(TIMESTAMP '2017-09-15 23:00:00',30) as var2,  
       DATE_ADD(nullstr,30) as var3  
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)
2017-10-15	2017-10-15	null

3.1.4.11.3.4. DATE_FORMAT

Syntax

```
VARCHAR DATE_FORMAT(TIMESTAMP time, VARCHAR to_format)  
VARCHAR DATE_FORMAT(VARCHAR date, VARCHAR to_format)  
VARCHAR DATE_FORMAT(VARCHAR date, VARCHAR from_format, VARCHAR to_format)
```

Input parameters

- **date**: specifies a date of the VARCHAR data type. The default format is yyyy-MM-dd HH:mm:ss.
- **time**: specifies a date of the TIMESTAMP data type.
- **from_format**: specifies a source date format.
- **to_format**: specifies a target date format.

Description

Converts a specified date from a source format to a target format. The date or time parameter specifies a date. The from_format parameter is optional and specifies the source date format. The default format is yyyy-MM-dd hh:mm:ss. The to_format parameter specifies the target date format. The return value is a date in the specified target format. If any argument is NULL or a parsing error occurs, the return value is NULL.

Example

Input example

date1 (VARCHAR)	datetime1 (VARCHAR)	nullstr (VARCHAR)
0915-2017	2017-09-15 00:00:00	null

Sample code

```
SELECT DATE_FORMAT(datetime1, 'yyMMdd') as var1,
       DATE_FORMAT(nullstr, 'yyMMdd') as var2,
       DATE_FORMAT(datetime1, nullstr) as var3,
       DATE_FORMAT(date1, 'MMdd-yyyy', nullstr) as var4,
       DATE_FORMAT(date1, 'MMdd-yyyy', 'yyyyMMdd') as var5,
       DATE_FORMAT(TIMESTAMP '2017-09-15 23:00:00', 'yyMMdd') as var6
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)	var4 (VARCHAR)	var5 (VARCHAR)	var6 (VARCHAR)
170915	null	null	null	20170915	170915

3.1.4.11.3.5. DATE_SUB

Syntax

```
VARCHAR DATE_SUB(VARCHAR startdate, INT days)
VARCHAR DATE_SUB(TIMESTAMP time, INT days)
```

Arguments

- startdate: specifies a date of the VARCHAR type. The format is yyyy-MM-dd or yyyy-MM-dd HH:mm:ss.
- time: specifies a date of the TIMESTAMP type.
- days: specifies the number of days. This argument is of the INT type.

Description

Return the date that is X (specified by days) days before the specified date. The return value is a VARCHAR type date in the format of yyyy-MM-dd. If any argument is NULL or a parsing error occurs, the return value is NULL.

Examples

Input example

date1 (VARCHAR)	nullstr (VARCHAR)
2017-10-15	null

Sample code

```
SELECT DATE_SUB(date1, 30) as var1,  
       DATE_SUB(TIMESTAMP '2017-10-15 23:00:00',30) as var2,  
       DATE_SUB(nullstr,30) as var3  
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)
2017-09-15	2017-09-15	null

3.1.4.11.3.6. DAYOFMONTH

Syntax

```
INT DAYOFMONTH(TIMESTAMP time)  
INT DAYOFMONTH(DATE date)
```

Arguments

- date: specifies a date of the DATE type.
- time: specifies a date of the TIMESTAMP type.

Description

Return the day number of the specified timestamp or date. Valid range: [1,31].

Examples

Input example

tsStr (VARCHAR)	dateStr (VARCHAR)	tdate (DATE)	ts (TIMESTAMP)
2017-10-15 00:00:00	2017-09-15	2017-11-10	2017-10-15 00:00:00

Sample code

```
SELECT DAYOFMONTH(TIMESTAMP '2016-09-15 00:00:00') as int1,  
       DAYOFMONTH(DATE '2017-09-22') as int2,  
       DAYOFMONTH(tdate) as int3,  
       DAYOFMONTH(ts) as int4,  
       DAYOFMONTH(CAST(dateStr AS DATE)) as int5,  
       DAYOFMONTH(CAST(tsStr AS TIMESTAMP)) as int6  
FROM T1
```

Output example

int1 (INT)	int2 (INT)	int3 (INT)	int4 (INT)	int5 (INT)	int6 (INT)
15	22	10	15	15	15

3.1.4.11.3.7. FROM_UNIXTIME

Syntax

```
VARCHAR FROM_UNIXTIME(BIGINT unixtime[, VARCHAR format])
```

Arguments

- **unixtime**: specifies a Unix timestamp (in seconds) of the BIGINT type. An exception occurs if this argument is of another type.
- **format**: specifies the target date format of the VARCHAR type. The default format is yyyy-MM-dd hh:mm:ss. This argument is optional.

Description

Return a VARCHAR type date in the specified date format. The default format is yyyy-MM-dd HH:mm:ss. If any argument is NULL or a parsing error occurs, the return value is NULL.

Examples

Input example

unixtime1(INT)	nullstr(VARCHAR)
1505404800	null

Sample code

```
SELECT FROM_UNIXTIME(unixtime1) as var1,
       FROM_UNIXTIME(unixtime1,'MMdd-yyyy') as var2,
       FROM_UNIXTIME(unixtime1,nullstr) as var3
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)
2017-09-15 00:00:00	0915-2017	null

3.1.4.11.3.8. HOUR

Syntax

```
INT HOUR(TIMESTAMP timestamp)
INT HOUR(TIME time)
```

Arguments

- **time**: specifies a time value of the TIME type.

- time: specifies a time value of the TIMESTAMP type.

Description

Return the hours (in 24-hour format) from the specified time or timestamp string. The return value ranges from 0 to 23.

Examples

Input example

datetime1 (VARCHAR)	time1 (VARCHAR)	time2 (TIME)	timestamp1 (TIMESTAMP)
2017-10-15 11:12:13	22:23:24	22:23:24	2017-10-15 11:12:13

Sample code

```
SELECT HOUR(TIMESTAMP '2016-09-20 23:33:33') as int1,  
       HOUR(TIME '23:30:33') as int2,  
       HOUR(time2) as int3,  
       HOUR(timestamp1) as int4,  
       HOUR(CAST(time1 AS TIME)) as int5,  
       HOUR(CAST(datetime1 AS TIMESTAMP)) as int6  
FROM T1
```

Output example

int1 (INT)	int2 (INT)	int3 (INT)	int4 (INT)	int5 (INT)	int6 (INT)
23	23	22	11	22	11

3.1.4.11.3.9. LOCALTIME

Syntax

```
TIME LOCALTIME
```

Arguments

The function is a non-parameterized constructor and can be used directly as a variable.

Description

Return the current time of the TIME type in the session time zone.

Examples

Sample code

```
SELECT LOCALTIME as `result`  
FROM T1
```

Output example

result(TIME)
19:00:47

3.1.4.11.3.10. MINUTE

Syntax

```
INT MINUTE(TIMESTAMP timestamp)
INT MINUTE(TIME time)
```

Arguments

- time: specifies a time value of the TIME type.
- time: specifies a time value of the TIMESTAMP type.

Description

Return the minutes from the specified time value. Valid range: [0, 59].

Examples

Input example

datetime1 (VARCHAR)	time1 (VARCHAR)	time2 (TIME)	timestamp1 (TIMESTAMP)
2017-10-15 11:12:13	22:23:24	22:23:24	2017-10-15 11:12:13

Sample code

```
SELECT MINUTE(TIMESTAMP '2016-09-20 23:33:33') as int1,
       MINUTE(TIME '23:30:33') as int2,
       MINUTE(time2) as int3,
       MINUTE(timestamp1) as int4,
       MINUTE(CAST(time1 AS TIME)) as int5,
       MINUTE(CAST(datetime1 AS TIMESTAMP)) as int6
FROM T1
```

Output example

int1 (INT)	int2 (INT)	int3 (INT)	int4 (INT)	int5 (INT)	int6 (INT)
33	30	23	12	23	12

3.1.4.11.3.11. MONTH

Syntax


```
INT MONTH(TIMESTAMP time)
INT MONTH(DATE date)
```

Arguments

- time: specifies a time value of the TIMESTAMP type.
- date: specifies a time value of the TIME type.

Description

Return the month number of the specified time or date. Valid range: [1, 12].

Examples

Input example

tsStr (VARCHAR)	dateStr (VARCHAR)	tdate (DATE)	ts (TIMESTAMP)
2017-10-15 00:00:00	2017-09-15	2017-11-10	2017-10-15 00:00:00

Sample code

```
SELECT MONTH(TIMESTAMP '2016-09-15 00:00:00') as int1,
       MONTH(DATE '2017-09-22') as int2,
       MONTH(tdate) as int3,
       MONTH(ts) as int4,
       MONTH(CAST(dateStr AS DATE)) as int5,
       MONTH(CAST(tsStr AS TIMESTAMP)) as int6
FROM T1
```

Output example

int1 (INT)	int2 (INT)	int3 (INT)	int4 (INT)	int5 (INT)	int6 (INT)
9	9	11	10	9	10

3.1.4.11.3.12. NOW

Syntax

```
BIGINT NOW()
```

Arguments

If no argument is specified, the Unix timestamp (in seconds) of the current system time is returned.

Description

Return the Unix timestamp (in seconds) of the current time zone. You can also specify an INT type argument as an offset (in seconds) and use UNIX_TIMESTAMP(CAST(NOW AS VARCHAR)) to convert the INT type argument to a timestamp value. Then add the offset to the current timestamp to return a value.

Examples

Input example

b(VARCHAR)
null

Sample code

```
SELECT NOW() as big1, NOW(b) as big2
FROM T1
```

Output example

big1(BIGINT)	big2 (BIGINT)
1403006911	null

3.1.4.11.3.13. SECOND

Syntax

```
INT SECOND(TIMESTAMP timestamp)
INT SECOND(TIME time)
```

Arguments

- time: specifies a time value of the TIME type.
- time: specifies a time value of the TIMESTAMP type.

Description

Extract the seconds from the specified time value. The return value ranges from 0 to 59.

Examples

Input example

datetime1(VARCHAR)	time1 (VARCHAR)	time2(TIME)	timestamp1(TIMESTAMP)
2017-10-15 11:12:13	22:23:24	22:23:24	2017-10-15 11:12:13

Sample code

```
SELECT SECOND(TIMESTAMP '2016-09-20 23:33:33') as int1,  
       SECOND(TIME '23:30:33') as int2,  
       SECOND(time2) as int3,  
       SECOND(timestamp1) as int4,  
       SECOND(CAST(time1 AS TIME)) as int5,  
       SECOND(CAST(datetime1 AS TIMESTAMP)) as int6  
FROM T1
```

Output example

int1(INT)	int2(INT)	int3(INT)	int4 (INT)	int5 (INT)	int6 (INT)
33	33	24	13	24	13

3.1.4.11.3.14. TIMESTAMPADD

Syntax

```
TIMESTAMP TIMESTAMPADD(interval,INT int_expr,TIMESTAMP datetime_expr)  
DATE TIMESTAMPADD(interval,INT int_expr,DATE datetime_expr)
```

Arguments

- interval

FRAC_SECOND	Specifies an interval in milliseconds.
SECOND	Specifies an interval in seconds.
MINUTE	Specifies an interval in minutes.
HOURL	Specifies an interval in hours.
DAY	Specifies an interval in days.
WEEK	Specifies an interval in weeks.
MONTH	Specifies an interval in months.
QUARTER	Specifies an interval in quarters.
YEAR	Specifies an interval in years.

- int_expr: specifies an interval that is added to the specified timestamp or date. The value must be an integer.
- datetime_expr: specifies a timestamp or date to which the interval is added.

Description

The data type of the result returned by the `TIMESTAMPADD` function is the same as that of the `datetime_expr` argument. The `TIMESTAMPADD` function adds an interval (`int_expr`) to a timestamp or date (`datetime_expr`) and returns the updated time.

Examples

Input example

a (TIMESTAMP)	b (DATE)
2018-07-09 10:23:56	1990-02-20

Sample code

```
SELECT
TIMESTAMPADD(HOUR,3,a) AS `result1`,
TIMESTAMPADD(DAY,3,b) AS `result2`
FROM T1
```

Output example

result1 (TIMESTAMP)	result2 (DATE)
2018-07-09 13:23:56	1990-02-23

3.1.4.11.3.15. TO_DATE

Syntax

```
Date TO_DATE(INT time)
Date TO_DATE(VARCHAR date)
Date TO_DATE(VARCHAR date, VARCHAR format)
```

Input parameters

- `time`: specifies the number of days from January 1, 1970 to a specified date. This parameter is of the `INT` data type.
- `date`: specifies a date of the `VARCHAR` data type. The default format is `yyyy-MM-dd`.
- `format`: specifies a date format of the `VARCHAR` data type.

Description

Converts a date of the `INT` or `VARCHAR` data type into a date of the `DATE` data type.

Example

Input example

date1 (bigint)	date2 (VARCHAR)	date3 (VARCHAR)
100	2017-09-15	20170915

Sample code

```
SELECT TO_DATE(date1) as var1,  
       TO_DATE(date2) as var2,  
       TO_DATE(date3, "yyyyMMdd") as var3  
FROM T1
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)	var3 (VARCHAR)
1970-04-11	2017-09-15	2017-09-15

3.1.4.11.3.16. TO_TIMESTAMP

Syntax

```
TIMESTAMP TO_TIMESTAMP(BIGINT time)  
TIMESTAMP TO_TIMESTAMP(VARCHAR date)  
TIMESTAMP TO_TIMESTAMP(VARCHAR date, VARCHAR format)
```

Arguments

- time: specifies a time value of the BIGINT type. The unit is milliseconds.
- date: specifies a date of the VARCHAR type. The default format is yyyy-MM-dd HH:mm:ss[.SSS].
- format: specifies the source date format of the VARCHAR type.

Description

Convert a date of the BIGINT or VARCHAR type to a date of the TIMESTAMP type.

Examples

Input example

timestamp1 (bigint)	timestamp2 (VARCHAR)	timestamp3 (VARCHAR)
1513135677000	2017-09-15 00:00:00	20170915000000

Sample code

```
SELECT TO_TIMESTAMP(timestamp1) as var1,  
       TO_TIMESTAMP(timestamp2) as var2,  
       TO_TIMESTAMP(timestamp3, "yyyyMMddHHmmss") as var3  
FROM T1
```

Output example

var1 (TIMESTAMP)	var2 (TIMESTAMP)	var3 (TIMESTAMP)
2017-12-13 03:27:57.0	2017-09-15 00:00:00.0	2017-09-15 00:00:00.0

3.1.4.11.3.17. UNIX_TIMESTAMP

Syntax

```
BIGINT UNIX_TIMESTAMP()
BIGINT UNIX_TIMESTAMP(VARCHAR date)
BIGINT UNIX_TIMESTAMP(VARCHAR date, VARCHAR format)
```

Arguments

- **date**: specifies a date of the VARCHAR type. The default format is yyyy-MM-dd HH:mm:ss.
- **format**: specifies the source date format of the VARCHAR type. The default format is yyyy-MM-dd HH:mm:ss.

Description

Convert the specified date to a Unix timestamp (in seconds) of the BIGINT type. Both the date and format parameters are optional. If neither is specified, the Unix timestamp (in seconds) of the current time is returned. If any argument is NULL or a parsing error occurs, the return value is NULL.

Examples

Input example

nullstr (VARCHAR)
null

Sample code

```
SELECT UNIX_TIMESTAMP() as big1,
       UNIX_TIMESTAMP(nullstr) as big2
FROM T1
```

Output example

big1 (BIGINT)	big2 (BIGINT)
1403006911	null

3.1.4.11.3.18. WEEK

Syntax

```
INT WEEK(DATE date)
INT WEEK(TIMESTAMP time)
```

Arguments

- date: specifies a date of the DATE type.
- time: specifies a date of the TIMESTAMP type.

Description

Calculate the week number of the specified date in the year. The valid week number range is from 1 to 53.

Examples

Input example

dateStr(VARCHAR)	date1 (DATE)	ts1 (TIMESTAMP)
2017-09-15	2017-11-10	2017-10-15 00:00:00

Sample code

```
SELECT WEEK(TIMESTAMP '2017-09-15 00:00:00') as int1,
       WEEK(date1) as int2, WEEK(ts1) as int3, WEEK(CAST(dateStr AS DATE)) as int4
FROM T1
```

Output example

int1 (INT)	int2 (INT)	int3 (INT)	int4 (INT)
37	45	41	37

3.1.4.11.3.19. YEAR

Syntax

```
INT YEAR(TIMESTAMP time)
INT YEAR(DATE date)
```

Arguments

- date: specifies a date of the DATE type.
- time: specifies a date of the TIMESTAMP type.

Description

Extract the year from the specified date.

Examples

Input example

tsStr(VARCHAR)	dateStr(VARCHAR)	tdate (DATE)	ts(TIMESTAMP)
2017-10-15 00:00:00	2017-09-15	2017-11-10	2017-10-15 00:00:00

Sample code

```
SELECT YEAR(TIMESTAMP '2016-09-15 00:00:00') as int1,
       YEAR(DATE '2017-09-22') as int2,
       YEAR(tdate) as int3,
       YEAR(ts) as int4,
       YEAR(CAST(dateStr AS DATE)) as int5,
       YEAR(CAST(tsStr AS TIMESTAMP)) as int6
FROM T1
```

Output example

int1 (BIGINT)	int2 (BIGINT)	int3 (BIGINT)	int4 (BIGINT)	int5 (BIGINT)	int6 (BIGINT)
2016	2017	2017	2017	2015	2017

3.1.4.11.4. Conditional functions

3.1.4.11.4.1. CASE WHEN

Syntax

```
CASE WHEN a THEN b [WHEN c THEN d]* [ELSE e] END
```

Input parameters

Parameter	Data type
a	BOOLEAN
b	BOOLEAN
c	BOOLEAN
d	BOOLEAN
e	BOOLEAN

Description

- If a is true, b is returned.
- If a is false and c is true, d is returned.
- If both a and c are false, e is returned.

Example

Note

If the CASE WHEN function returns a constant string, spaces are added after the string. An example is described as follows. When the return value is "ios", several spaces are added after this string.

```
case when device_type = 'android'
then 'android'
else 'ios'
end as os
```

You can resolve the issue in the following two ways:

- Use the trim() method to remove the spaces. As shown in the following example, you can change os to trim(os) in the statement.
- Use the cast() method to convert the constant string into the VARCHAR data type.

- Input example

device_type (VARCHAR)
android
ios
win

- Sample code 1: using the trim() method

```
SELECT
  trim(os), // The trim() method is used.
  CHAR_LENGTH(trim(os)) // The trim() method is used.
from(
  SELECT
    case when device_type = 'android'
    then 'android'
    else 'ios'
  end as os
FROM T1
);
```

Sample code 2: using the cast() method

```
SELECT
  os,
  CHAR_LENGTH(os)
from
(SELECT
  case when device_type = 'android'
  then cast('android' as varchar) // The cast() method is used.
  else cast('ios' as varchar) // The cast() method is used.
  end as os
FROM T1
);
```

- Output example

os (VARCHAR)	length (INT)
android	7
ios	3
ios	3

3.1.4.11.4.2. COALESCE

Syntax

```
COALESCE(A,B,...)
```

Arguments

A and B: specify values to be checked. NULL values are allowed in the list of values. Non-NULL values in the list must be of the same type. Otherwise, an exception occurs. The return type is the same as the argument type.

Description

Return the first non-NULL value in the specified list. If all values in the list are NULL, the return value is NULL.

 **Note** The list must contain at least one argument. Otherwise, an exception occurs.

Examples

Input example

var1(VARCHAR)	var2(VARCHAR)
null	30

Sample code

```
SELECT COALESCE(var1,var2) as aa  
FROM T1;
```

Output example

aa(VARCHAR)
30

3.1.4.11.4.3. IF

Syntax

```
T IF(BOOLEAN testCondition, T valueTrue, T valueFalseOrNull)
```

 **Note** T indicates any return type.

Arguments

- testCondition: BOOLEAN type
- valueTrue and valueFalseOrNull: These two arguments must be of the same type. The specific type is not restricted.

Description

Use the BOOLEAN value of testCondition as the criterion. If the value is true, valueTrue is returned. If the value is false, valueFalseOrNull is returned. If testCondition is NULL, valueFalseOrNull is returned. If any other argument is NULL, the operation is performed based on normal semantics, and the return value is of the T type.

Examples

Input example

int1(INT)	int2(INT)	str1(VARCHAR)	str2(VARCHAR)
1	2	Jack	Harry
1	2	Jack	null
1	2	null	Harry

Sample code

```
SELECT IF(int1 < int2,str1, str2) as int1  
FROM T1
```

Output example

int1 (VARCHAR)
Jack
Jack
null

3.1.4.11.4.4. IS_ALPHA

Syntax

```
BOOLEAN IS_ALPHA(VARCHAR str)
```

Arguments

str: specifies a string of the VARCHAR type.

Description

Check whether the specified string contains only letters. If it contains only letters, TRUE is returned. Otherwise, FALSE is returned.

Examples

Input example

e (VARCHAR)	f (VARCHAR)	g (VARCHAR)
3	asd	null

Sample code

```
SELECT IS_ALPHA(e) as boo1, IS_ALPHA(f) as boo2, IS_ALPHA(g) as boo3  
FROM T1
```

Output example

boo1 (BOOLEAN)	boo2 (BOOLEAN)	boo3 (BOOLEAN)
false	Yes	false

3.1.4.11.4.5. IS_DECIMAL

Syntax

```
BOOLEAN IS_DECIMAL(VARCHAR str)
```

Input parameter

str: specifies a string of the VARCHAR data type.

Description

If a specified string contains only decimal characters, true is returned. Otherwise, false is returned.

Example

Input example

a (VARCHAR)	b (VARCHAR)	c (VARCHAR)	d (VARCHAR)	e (VARCHAR)	f (VARCHAR)	g (VARCHAR)
1	123	2	11.4445	3	asd	null

Sample code

```
SELECT IS_DECIMAL(a) as boo1,IS_DECIMAL(b) as boo2,IS_DECIMAL(c) as boo3,  
       IS_DECIMAL(d) as boo4,IS_DECIMAL(e) as boo5,IS_DECIMAL(f) as boo6,IS_DECIMAL(g) as boo7  
FROM T1
```

Output example

boo1 (BOOLEAN)	boo2 (BOOLEAN)	boo3 (BOOLEAN)	boo4 (BOOLEAN)	boo5 (BOOLEAN)	boo6 (BOOLEAN)	boo7 (BOOLEAN)
true	true	true	true	true	false	false

3.1.4.11.4.6. IS_DIGIT

Syntax

```
BOOLEAN IS_DIGIT(VARCHAR str)
```

Arguments

str: specifies a string of the VARCHAR type.

Description

Check whether the specified string contains only digits. If it only contains digits, TRUE is returned. Otherwise, FALSE is returned. The return value is of the BOOLEAN type.

Examples

Input example

e(VARCHAR)	f(VARCHAR)	g(VARCHAR)
3	asd	null

Sample code

```
SELECT IS_DIGIT(e) as boo1,IS_DIGIT(f) as boo2,IS_DIGIT(g) as boo3
FROM T1
```

Output example

boo1(BOOLEAN)	boo2(BOOLEAN)	boo3(BOOLEAN)
true	false	false

3.1.4.11.4.7. NULLIF

Syntax

```
NULLIF(A,B)
```

Arguments

A and B: INT type

Description

If A and B have the same value, null is returned. If A and B have different values, the value of the first argument is returned.

Examples

Input example

var1(INT)	var2(INT)
30	30

Sample code

```
SELECT NULLIF(var1,var2) as aa
FROM T1;
```

Output example

aa(VARCHAR)
null

3.1.4.11.5. Table-valued functions

3.1.4.11.5.1. GENERATE_SERIES

Syntax

```
GENERATE_SERIES(INT from, INT to)
```

Arguments

- from: specifies a lower limit of the INT type.
- to: specifies an upper limit of the INT type.

Description

Generate a series of continuous values from the lower limit to the upper limit minus one.

Examples

Input example

s(INT)	e(INT)
1	3
-2	1

Sample code

```
SELECT s, e, v
FROM T1, lateral table(GENERATE_SERIES(s, e))
as T(v)
```

Output example

s(INT)	e(INT)	v(INT)
1	3	1
1	3	2
-2	1	-2
-2	1	-1
-2	1	-0

3.1.4.11.5.2. JSON_TUPLE

Syntax

```
JSON_TUPLE(str, path1, path2 ..., pathN)
```

Arguments

- str: specifies a JSON string.

- path1 to pathN: specify strings that represent paths, which do not need to begin with a dollar sign (\$).

Description

Obtain the value represented by each path string from a JSON string.

Examples

Input example

d(VARCHAR)	s(VARCHAR)
{ "qwe" : " asd" , " qwe2" : " asd2" , " qwe3" : " asd3" }	qwe3
{ "qwe" : " asd4" , " qwe2" : " asd5" , " qwe3" : " asd3" }	qwe2

Sample code

```
SELECT d,v
FROM T1,lateral table(JSON_TUPLE(d,'qwe',s))
as T(v)
```

Output example

d(VARCHAR)	v(VARCHAR)
{ "qwe" : " asd" , " qwe2" : " asd2" , " qwe3" : " asd3" }	asd
{ "qwe" : " asd" , " qwe2" : " asd2" , " qwe3" : " asd3" }	asd3
{ "qwe" : " asd4" , " qwe2" : " asd5" , " qwe3" : " asd3" }	asd4
{ "qwe" : " asd4" , " qwe2" : " asd5" , " qwe3" : " asd3" }	asd5

3.1.4.11.5.3. STRING_SPLIT

Syntax

```
STRING_SPLIT(VARCHAR,separator)
```

Arguments

separator: specifies a separator of the VARCHAR type.

Description

Use a separator to split a string into multiple substrings.

Examples

Input example

d (VARCHAR)	s (VARCHAR)
abc-bcd	-
hhh	-

Sample code

```
SELECT d,v
FROM T1, lateral table(STRING_SPLIT(d,s))
as T(v)
```

Output example

d (VARCHAR)	v (VARCHAR)
abc-bcd	abc
abc-bcd	bcd
hhh	hhh

3.1.4.11.6. Type conversion functions

3.1.4.11.6.1. CAST

Syntax

```
CAST(A AS type)
```

Argument

A: For more information about the argument type, see [Data types](#).

Description

Convert a value to a specified type.

Examples

Input example

var1 (VARCHAR)	var2 (INT)
1000	30

Sample code

```
SELECT CAST(var1 AS INT) as aa
FROM T1;
```

Output example

aa (INT)
1000

3.1.4.11.7. Aggregate functions

3.1.4.11.7.1. AVG

Syntax

```
AVG(A)
```

Argument

A: specifies a value of numeric types including TINYINT, SMALLINT, INT, BIGINT, FLOAT, DECIMAL, and DOUBLE.

Description

Return the average number.

Examples

Input example

var1 (INT)	var2 (INT)
4	30
6	30

Sample code

```
SELECT AVG(var1) as aa
FROM T1;
```

Output example

aa (INT)
5

3.1.4.11.7.2. CONCAT_AGG

Syntax

```
CONCAT_AGG([linedelimiter,] value)
```

Argument

linedelimiter: specifies a connector, which currently can only be a string constant.

Description

Use a connector to concatenate multiple fields. The default connector is "\n". The return value is a new concatenated string of the VARCHAR type.

Examples

Input example

c (VARCHAR)
Hi
Hi
Hi
Hi
Hi
Hi
Hi
Hi
Hi
Hi

Sample code

```
SELECT concat_agg(c) as var1,  
       concat_agg('-', c) as var2  
FROM MyTable  
GROUP BY c
```

Output example

var1 (VARCHAR)	var2 (VARCHAR)
Hi\nHi\nHi\nHi\nHi\nHi\nHi\nHi\nHi	Hi-Hi-Hi-Hi-Hi-Hi-Hi-Hi-Hi

3.1.4.11.7.3. COUNT

Syntax

```
COUNT(A)
```

Arguments

- A
- Supported data types include numeric types (TINYINT, SMALLINT, INT, BIGINT, FLOAT, DECIMAL, and DOUBLE), BOOLEAN, and VARCHAR.
 - The DATE, TIME, TIMESTAMP, and VARBINARY data types are not supported.

Description

Return the number of input parameters.

Examples

Input example

var1 (VARCHAR)
1000
100
10
1

Sample code

```
SELECT COUNT(var1) as aa
FROM T1;
```

Output example

aa (BIGINT)
4

3.1.4.11.7.4. FIRST_VALUE

Syntax

```
T FIRST_VALUE(value: T)
```

To obtain the first record based on parameters, define the function as follows:

```
T FIRST_VALUE(value: T, order: Long)
```

Arguments

- value: specifies a value of any type. Only one argument is supported.
- order: specifies a record with the smallest order value as the first record.

Description

Obtain the first non-null record of a stream.

Examples

Input example

a (BIGINT)	b (INT)	c (VARCHAR)
1L	1	"Hello"
2L	2	"Hello"
3L	3	"Hello"
4L	4	"Hello"
5L	5	"Hello"
6L	6	"Hello"
7L	7	"Hello World"
8L	8	"Hello World"
20L	20	"Hello World"

Sample code

```
SELECT  
c,  
first_value(b) OVER (PARTITION BY c ORDER BY proctime RANGE UNBOUNDED preceding) as var1  
from T1
```

Output example

c (VARCHAR)	var1 (INT)
Hello	1

c (VARCHAR)	var1 (INT)
Hello	1
Hello	1
Hello	1
Hello	1
Hello	1
Hello World	7
Hello World	7
Hello World	7

3.1.4.11.7.5. LAST_VALUE

Syntax

```
T LAST_VALUE(value: T)
```

To return the last value in an ordered set of values, define the function as follows:

```
T LAST_VALUE(value: T, order: Long)
```

Arguments

- value: specifies a value of any type.
- order: specifies the order of the values. A value with the greatest order value is considered as the last value.

Description

Obtain the last data record of a stream.

Examples

Input example

a(BIGINT)	b(INT)	c(VARCHAR)
1L	1	"Hello"
2L	2	"Hello"
3L	3	"Hello"
4L	4	"Hello"

a(BIGINT)	b(INT)	c(VARCHAR)
5L	5	"Hello"
6L	6	"Hello"
7L	7	"Hello World"
8L	8	"Hello World"
20L	20	"Hello World"

Sample code

```
SELECT
c,
last_value(b) OVER (PARTITION BY c ORDER BY proctime RANGE UNBOUNDED preceding) as var1
from T1
```

Output example

c(VARCHAR)	var1(INT)
Hello	1
Hello	2
Hello	3
Hello	4
Hello	5
Hello	6
Hello World	7
Hello World	8
Hello World	20

3.1.4.11.7.6. MAX

Syntax

```
MAX(A)
```

Arguments

A

- Supported data types include numeric values (TINYINT, SMALLINT, INT, BIGINT, FLOAT, DECIMAL, and DOUBLE), BOOLEAN, and VARCHAR.
- The DATE, TIME, TIMESTAMP, and VARBINARY data types are not supported.

Description

Return the maximum value of all the input values.

Examples

Input example

var1(INT)
4
8

Sample code

```
SELECT MAX(a) as aa
FROM T1;
```

Output example

aa(INT)
8

3.1.4.11.7.7. MIN

Syntax

```
MIN(A)
```

Arguments

A

- Supported data types include numeric values (TINYINT, SMALLINT, INT, BIGINT, FLOAT, DECIMAL, and DOUBLE), BOOLEAN, and VARCHAR.
- The DATE, TIME, TIMESTAMP, and VARBINARY data types are not supported.

Description

Return the minimum value of all the input values.

Examples

Input example

var1(INT)
4
8

Sample code

```
SELECT MIN(a) as aa
FROM T1;
```

Output example

aa(INT)
4

3.1.4.11.7.8. STDDEV_POP

Syntax

```
T STDDEV_POP(T value)
```

Argument

value: specifies a number of the BIGINT or DOUBLE type.

Description

Return the standard deviation for the population of all values in the specified expression.

Examples

Input example

a (DOUBLE)	c (VARCHAR)
0	Hi
1	Hi
2	Hi
3	Hi
4	Hi
5	Hi
6	Hi

a (DOUBLE)	c (VARCHAR)
7	Hi
8	Hi
9	Hi

Sample code

```
SELECT c, STDDEV_POP(a) as dou
FROM MyTable
GROUP BY c
```

Output example

c (VARCHAR)	dou1 (DOUBLE)
Hi	2.8722813232690143

3.1.4.11.7.9. SUM

Syntax

```
SUM(A)
```

Arguments

A: specifies a value of the numeric types including TINYINT, SMALLINT, INT, BIGINT, FLOAT, DECIMAL, and DOUBLE.

Description

Return the sum of all input values.

Examples

Input example

var1 (INT)
4
4

Sample code

```
SELECT sum(a) as aa
FROM T1;
```

Output example

aa (INT)
8

3.1.4.11.7.10. VAR_POP

Syntax

T VAR_POP(T value)

Argument

value: specifies a number of the BIGINT or DOUBLE type.

Description

Return the statistical variance for the population of all values in the specified expression.

Examples

Input example

a (BIGINT)	c (VARCHAR)
2900	Hi
2500	Hi
2600	Hi
3100	Hello
11000	Hello

Sample code

SELECT VAR_POP(a) as `result`, c FROM MyTable GROUP BY c

Output example

result (BIGINT)	c
28889	Hi
15602500	Hello

3.1.4.11.8. Other functions

3.1.4.11.8.1. UUID

Syntax

```
VARCHAR UUID()
```

Arguments

None

Description

Return a globally unique identifier.

Examples

Sample code

```
SELECT uuid() as `result`  
FROM T1
```

Output example

result(VARCHAR)
a364e414-e68b-4e5c-9166-65b3a153e257

3.1.4.12. UDXs

3.1.4.12.1. Overview

This topic describes how to build a development environment and use user-defined extensions (UDXs) in Realtime Compute for Apache Flink.



Notice

- Only Realtime Compute for Apache Flink in exclusive mode supports UDXs.
- Blink is developed based on open source Flink SQL by Alibaba Cloud Realtime Compute for Apache Flink to improve computing performance. UDXs can only be used in Blink.

UDX types

Realtime Compute for Apache Flink supports three types of UDXs, as described in the following table.

UDX type	Description

UDX type	Description
UDF	User-defined scalar function. The relationship between the input and output of UDFs is one-to-one mapping, which indicates that one value is returned each time a UDF reads one row of data.
UDAF	User-defined aggregation function. The relationship between the input and output of UDAFs is many-to-one mapping. A UDAF aggregates multiple input records into one output record. A UDAF can be used with the GROUP BY clause of SQL.
UDTF	User-defined table-valued function. When a UDTF is called, it generates multiple columns or rows of data.

Example

Realtime Compute for Apache Flink provides an example of a UDX to facilitate your business development. This example demonstrates the implementation of a UDF, UDAF, and UDTF.

Note

- In this example, a development environment of the required version is configured. You do not need to build another development environment.
- The example provides Maven projects. You can use IntelliJ IDEA for development.

- Realtime Compute for Apache Flink V3.0

[Blink_UDX_3x](#)

- Realtime Compute for Apache Flink V2.0

[Blink_UDX_2x](#)

- Realtime Compute for Apache Flink V1.0

[Blink_UDX_1x](#)

Build a development environment

The development of UDXs depends on the following JAR packages. You can download the packages as needed.

- Realtime Compute for Apache Flink versions earlier than V3.2.1

- [flink-streaming-java_2.11](#)
- [flink-table_2.11](#)
- [flink-core-blink-2.2.4](#)

- Realtime Compute for Apache Flink V3.2.1 and later

Add a [POM dependency](#) based on your open source software version. Download and view the [example of a complete dependency](#).

 **Note** If you want to use Snapshot, you can add a [POM dependency](#) based on your Snapshot version.

Register and use resources

1. Log on to the Realtime Compute development platform.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Resources** tab.
4. In the upper-right corner of the **Resources** pane, click **Create Resource**.
5. In the **Upload Resource** dialog box, configure resource parameters.

Parameter	Description
Location	You can upload JAR packages only from your machine in the Realtime Compute for Apache Flink console.  Note The maximum size of a JAR package that can be uploaded from your machine is 300 MB. If the JAR package exceeds 300 MB, you must upload it to the Object Storage Service (OSS) bucket bound to your cluster or use APIs to upload it.
Resource	Click Upload Resource and select the resource that you want to reference.
Resource Name	Enter a name for the resource.
Resource Description	Enter a resource description.
Resource Type	Select the type of the resource. The type can be JAR, DICTIONARY, or PYTHON.

6. In the **Resources** pane, find the new resource, and move the pointer over **More** in the Actions column.
7. In the drop-down list, select **Reference**.
8. In the code editor, declare the UDX at the beginning. The following statement is an example:

```
CREATE FUNCTION stringLengthUdf AS 'com.hjc.test.blink.sql.udx.StringLengthUdf';
```

Types of parameters and return values

When you define Java UDXs in Realtime Compute for Apache Flink, you can use Java data types in parameters and return values. The following table lists the mappings between Realtime Compute for Apache Flink and Java data types.

Data type of Realtime Compute for Apache Flink	Java data type
TINYINT	java.lang.Byte
SMALLINT	java.lang.Short
INT	java.lang.Integer

Data type of Realtime Compute for Apache Flink	Java data type
BIGINT	java.lang.Long
FLOAT	java.lang.Float
DOUBLE	java.lang.Double
DECIMAL	java.math.BigDecimal
BOOLEAN	java.lang.Boolean
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp
CHAR	java.lang.Character
STRING	java.lang.String
VARBINARY	java.lang.byte[]
ARRAY	Not supported
MAP	Not supported

Obtain parameters of UDXs

UDXs support an optional `open(FunctionContext context)` method. You can use `FunctionContext` to pass custom configuration items.

Assume that you must add the following two parameters to your job:

```
testKey1=lincoln  
test.key2=todd
```

Take a UDTF as an example. You can use `context.getJobParameter` in the open method to obtain parameters. The following code shows an example:

```
public void open(FunctionContext context) throws Exception {  
    String key1 = context.getJobParameter("testKey1", "empty");  
    String key2 = context.getJobParameter("test.key2", "empty");  
    System.err.println(String.format("end open: key1:%s, key2:%s", key1, key2));  
}
```

3.1.4.12.2. UDF

This topic describes how to build a development environment, write business logic code, and publish a user-defined scalar function (UDF) in Realtime Compute for Apache Flink.

 **Notice** Only Realtime Compute for Apache Flink in exclusive mode supports user-defined extensions (UDXs).

Description

A UDF maps zero, one, or multiple scalar values to a new scalar value.

Build a development environment

For more information about how to build a development environment, see [Build a development environment](#).

Write business logic code

To define a UDF, you must extend the `ScalarFunction` class by implementing the `eval` method. The `open` and `close` methods are optional.

 **Notice** UDFs return the same output for the same input by default. However, a UDF where an external service is called may return different output results even if the input values are the same. If a UDF cannot generate the same output for the same input, we recommend that you use the `overrideIsDeterministic` method to make it return `false`. Otherwise, the output may not meet your expectations in some cases. For example, a UDF operator moves forward.

The following sample code is written in Java:

```
package com.hjc.test.blink.sql.udx;
import org.apache.flink.table.functions.FunctionContext;
import org.apache.flink.table.functions.ScalarFunction;
public class StringLengthUdf extends ScalarFunction {
    // The open method is optional.
    // To use the open method, you must add 'import org.apache.flink.table.functions.FunctionContext;' to the code.
    @Override
    public void open(FunctionContext context) {
    }
    public long eval(String a) {
        return a == null ? 0 : a.length();
    }
    public long eval(String b, String c) {
        return eval(b) + eval(c);
    }
    // The close method is optional.
    @Override
    public void close() {
    }
}
```

Write SQL statements

You can write SQL statements in a specified class. The following example shows the SQL statements in a UDX:


```
-- udf str.length()
CREATE FUNCTION stringLengthUdf AS 'com.hjc.test.blink.sql.udx.StringLengthUdf';
create table sls_stream(
  a int,
  b int,
  c varchar
) with (
  type='sls',
  endPoint='<yourEndpoint>',
  accessKeyId='<yourAccessId>',
  accessKeySecret='<yourAccessSecret>',
  startTime = '2017-07-04 00:00:00',
  project='<yourProjectName>',
  logStore='<yourLogStoreName>',
  consumerGroup='consumerGroupTest1'
);
create table rds_output(
  id int,
  len bigint,
  content VARCHAR
) with (
  type='rds',
  url='yourDatabaseURL',
  tableName='<yourDatabaseTableName>',
  userName='<yourDatabaseUserName>',
  password='<yourDatabasePassword>'
);
insert into rds_output
select
  a,
  stringLengthUdf(c),
  c as content
from sls_stream;
```

Register and use resources

1. Log on to the Realtime Compute development platform.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Resources** tab.
4. In the upper-right corner of the **Resources** pane, click **Create Resource**.
5. In the **Upload Resource** dialog box, configure resource parameters.

Parameter	Description

Parameter	Description
Location	You can upload JAR packages only from your machine in the Realtime Compute for Apache Flink console.  Note The maximum size of a JAR package that can be uploaded from your machine is 300 MB. If the JAR package exceeds 300 MB, you must upload it to the Object Storage Service (OSS) bucket bound to your cluster or use APIs to upload it.
Resource	Click Upload Resource and select the resource that you want to reference.
Resource Name	Enter a name for the resource.
Resource Description	Enter a resource description.
Resource Type	Select the type of the resource. The type can be JAR, DICTIONARY, or PYTHON.

- In the **Resources** pane, find the new resource, and move the pointer over **More** in the Actions column.
- In the drop-down list, select **Reference**.
- In the code editor, declare the UDX at the beginning. The following statement is an example:

```
CREATE FUNCTION stringLengthUdf AS 'com.hjc.test.blink.sql.udx.StringLengthUdf';
```

Publish and use a UDF

Click **Publish** on the **Development** page for the job where you want to publish a UDF. Then, find the job on the **Administration** page and click **Start** in the Actions column to publish the UDF.

FAQ

Q: Why does the random number generator always generate the same value at runtime?

A: If no input parameters are passed to a UDX and you do not declare it as nondeterministic, the UDX may be optimized during compilation to return a constant value. To avoid this, you can use the `override isDeterministic` method to make it return `false`.

3.1.4.12.3. UDAF

This topic describes how to build a development environment, write business logic code, and publish a user-defined aggregation function (UDAF) in Realtime Compute for Apache Flink.

 **Notice** Only Realtime Compute for Apache Flink in exclusive mode supports user-defined extensions (UDXs).

Description

A UDAF aggregates multiple values into a single value.

Methods of the UDAF abstract class

 **Note** A UDAF can be implemented in Java or Scala. However, we recommend that you use Java because Scala data types may cause unnecessary performance overhead.

The following code shows the core methods of the `AggregateFunction` class.

- `createAccumulator` and `getValue` methods

```
/*
 * @param <T> The type of the output returned by a UDAF.
 * @param <ACC> The accumulator type of a UDAF. An accumulator stores the intermediate aggregation results of a UDAF. You can design an accumulator for each UDAF as required.
 */
public abstract class AggregateFunction<T, ACC> extends UserDefinedFunction {
    /*
     * Initializes the accumulator in AggregateFunction.
     * The system calls the following method before it aggregates data for the first time.
     */
    public ACC createAccumulator();
    /*
     * The system calls the following method after each aggregation is completed.
     */
    public T getValue(ACC accumulator);
}
```

-  **Note**
- The `createAccumulator` and `getValue` methods can be defined in the `AggregateFunction` abstract class.
 - A UDAF must contain at least one `accumulate` method.

- `accumulate` method

```
public void accumulate(ACC accumulator, ...[ user input]...);
```

-  **Note**
- You must implement an `accumulate` method to describe how to compute input data and update an accumulator to the aggregation result.
 - The first parameter of the `accumulate` method must be an accumulator of the `ACC` type defined in `AggregateFunction`. During code execution, the runtime code sends the previous value in the accumulator and the specified upstream data to the `accumulate` method for aggregation. The upstream data can be of any type and contains any number of data records.

- `retract` and `merge` methods

The `createAccumulator`, `getValue`, and `accumulate` methods can be used together to design a basic UDAF. However, Realtime Compute for Apache Flink also requires the `retract` and `merge` methods in some special scenarios.

In most scenarios, computing is early firing for an infinite stream. To refine the early fired results, you can implement a `retract` method. The SQL optimizer automatically determines the conditions in which data needs to be retracted and the operations needed to process data marked with retract tags. You must implement a `retract` method to retract input data.

```
public void retract(ACC accumulator, ...[ user input]...);
```

Note

- The `retract` method is the reverse operation of the `accumulate` method. For example, in a count UDAF, the number of data records in the computing result increases by one each time the `accumulate` method is called to process a data record, whereas the number of data records in the result decreases by one each time the `retract` method is called to process a data record.
- Similar to the `accumulate` method, the first parameter of the `retract` method must be an accumulator of the ACC type defined in `AggregateFunction`. During code execution, the runtime code sends the previous value in the accumulator and the specified upstream data to the `retract` method for aggregation. The upstream data can be of any type and contain any number of data records.

Realtime Compute for Apache Flink requires the `merge` method in some scenarios. For example, if you use a session window to aggregate data, you must use the `merge` method. Realtime Compute for Apache Flink can process out-of-order data. Newly arrived data may fill the gap between two separate sessions, which results in the merge of the two sessions. In this case, you must use the `merge` method to integrate multiple accumulators into one accumulator.

```
public void merge(ACC accumulator, Iterable<ACC> its);
```

Note

- The first parameter of the `merge` method must be an accumulator of the ACC type defined in `AggregateFunction`. After the `merge` method is executed, the state data of `AggregateFunction` is stored in the first accumulator.
- The second parameter of the `merge` method is an iterator of one or more accumulators of the ACC type.

Build a development environment

For more information about how to build a development environment, see [Build a development environment](#).

Write business logic code

The following Java code is an example:

```
import org.apache.flink.table.functions.AggregateFunction;
public class CountUdaf extends AggregateFunction<Long, CountUdaf.CountAccum> {
    // Define the data schema of the accumulator that stores the state data of a count UDAF.
    public static class CountAccum {
        public long total;
    }
    // Initialize the accumulator of the count UDAF.
    public CountAccum createAccumulator() {
        CountAccum acc = new CountAccum();
        acc.total = 0;
        return acc;
    }
    // Call the getValue method to obtain the result of the count UDAF from the accumulator that stores the UDAF state data.
    public Long getValue(CountAccum accumulator) {
        return accumulator.total;
    }
    // Call the accumulate method to update the accumulator that stores the state data of the count UDAF based on the input data.
    public void accumulate(CountAccum accumulator, Object iValue) {
        accumulator.total++;
    }
    public void merge(CountAccum accumulator, Iterable<CountAccum> its) {
        for (CountAccum other : its) {
            accumulator.total += other.total;
        }
    }
}
```

 **Note** The open and close methods are optional for a subclass of `AggregateFunction`. For more information, see the examples of [UDF](#) or [UDTF](#).

Register and use resources

1. Log on to the Realtime Compute development platform.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Resources** tab.
4. In the upper-right corner of the **Resources** pane, click **Create Resource**.
5. In the **Upload Resource** dialog box, configure resource parameters.

Parameter	Description

Parameter	Description
Location	You can only upload JAR packages from your machine in the Realtime Compute for Apache Flink console.  Note The maximum size of a JAR package that can be uploaded from your machine is 300 MB. If the JAR package exceeds 300 MB, you must upload it to the Object Storage Service (OSS) bucket that is bound to your cluster or use an API to upload it.
Resource	Click Upload Resource to select the resource that you want to reference.
Resource Name	Enter a name for the resource.
Resource Description	Enter a resource description.
Resource Type	Select the type of the resource. The type can be JAR, DICTIONARY, or PYTHON.

- In the **Resources** pane, find the new resource, and move the pointer over **More** in the Actions column.
- In the drop-down list, select **Reference**.
- In the code editor, declare the UDX. The following statement is an example:

```
CREATE FUNCTION stringLengthUdf AS 'com.hjc.test.blink.sql.udx.StringLengthUdf';
```

Publish and use a UDAF

For more information about how to publish and use a UDAF, see [Publish a job](#) and [Start a job](#).

Example

```
-- Use a UDAF to calculate the count.
CREATE FUNCTION countUdaf AS 'com.hjc.test.blink.sql.udx.CountUdaf';
create table sls_stream(
  a int,
  b bigint,
  c varchar
) with (
  type='sls',
  endPoint='yourEndpoint',
  accessKeyId='yourAccessId',
  accessKeySecret='yourAccessSecret',
  startTime='2017-07-04 00:00:00',
  project='<yourProjectName>',
  logStore='stream-test2',
  consumerGroup='consumerGroupTest3'
);
create table rds_output(
  len1 bigint,
  len2 bigint
) with (
  type='rds',
  url='yourDatabaseURL',
  tableName='<yourDatabaseTableName>',
  userName='<yourDatabaseUserName>',
  password='<yourDatabasePassword>'
);
insert into rds_output
select
  count(a),
  countUdaf(a)
from sls_stream;
```

3.1.4.12.4. UDTF

This topic describes how to build a development environment, write code, and publish a user-defined table-valued function (UDTF) in Realtime Compute for Apache Flink.

 **Note** Only Realtime Compute for Apache Flink in exclusive mode supports user-defined extensions (UDXs).

Description

Similar to a user-defined scalar function (UDF), a UDTF uses zero, one, or multiple scalar values as input parameters (including variable-length parameters). Different from a UDF, a UDTF returns any number of rows, rather than a single value. The returned rows can consist of one or more columns.

Build a development environment

For more information, see [Build a development environment](#).

Write business logic code

A UDTF needs to implement the `eval` method in the `TableFunction` class. The `open` and `close` methods are optional. The following sample code is written in Java:

```
package com.hjc.test.blink.sql.udx;
import org.apache.flink.table.functions.FunctionContext;
import org.apache.flink.table.functions.TableFunction;
public class SplitUdtf extends TableFunction<String> {
    // The open method is optional. To use the open method, you must add 'import org.apache.flink.table.functions.FunctionContext;' to the code.
    @Override
    public void open(FunctionContext context) {
        // ...
    }
    public void eval(String str) {
        String[] split = str.split("\\|");
        for (String s : split) {
            collect(s);
        }
    }
    // The close method is optional.
    @Override
    public void close() {
        // ...
    }
}
```

Return multiple rows

A UDTF can convert the output result from a single row to multiple rows by calling the `collect` method multiple times.

Return multiple columns

A UDTF can also convert the output result from a single column to multiple columns. If you want a UDTF to return multiple columns, declare the return value as a tuple or row.

- Declare the return value as a tuple.

Realtime Compute for Apache Flink supports `Tuple1` to `Tuple25`, which define 1 to 25 fields. The following example is a UDTF that uses `Tuple3` to return three fields:


```
import org.apache.flink.api.java.tuple.Tuple3;
import org.apache.flink.table.functions.TableFunction;
// If the return value is declared as a tuple, you must explicitly declare the generic types of the tuple, such
as STRING, LONG, and INTEGER in this example.
public class ParseUdtf extends TableFunction<Tuple3<String, Long, Integer>> {
    public void eval(String str) {
        String[] split = str.split(",");
        // The code is for demonstration only. In practice, you may need to add more verification logic.
        String first = split[0];
        long second = Long.parseLong(split[1]);
        int third = Integer.parseInt(split[2]);
        Tuple3<String, Long, Integer> tuple3 = Tuple3.of(first, second, third);
        collect(tuple3);
    }
}
```

 **Note** If the return value is declared as a tuple, it can contain a maximum of 25 fields, and the field value cannot be null.

- Declare the return value as a row.

The following example is a UDTF that uses a row to return three fields:

```
import org.apache.flink.table.types.DataType;
import org.apache.flink.table.types.DataTypes;
import org.apache.flink.table.functions.TableFunction;
import org.apache.flink.types.Row;
public class ParseUdtf extends TableFunction<Row> {
    public void eval(String str) {
        String[] split = str.split(",");
        String first = split[0];
        long second = Long.parseLong(split[1]);
        int third = Integer.parseInt(split[2]);
        Row row = new Row(3);
        row.setField(0, first);
        row.setField(1, second);
        row.setField(2, third);
        collect(row);
    }
    @Override
    // If the return value is declared as a row, you must overload the getResultType method to explicitly declare the types of fields to be returned.
    public DataType getResultType(Object[] arguments, Class[] argTypes) {
        return DataTypes.createRowType(DataTypes.STRING, DataTypes.LONG, DataTypes.INT);
    }
}
```

 **Note** If the return value is declared as a row, the field value can be null. However, you must overload the `getResultType` method.

SQL syntax

A UDTF supports `CROSS JOIN` and `LEFT JOIN`. When you use a UDTF, you must add the keywords `LATERAL` and `TABLE`. You must also specify an alias for the UDTF, for example, `ParseUdtf` in the preceding code.

```
CREATE FUNCTION parseUdtf AS 'com.alibaba.blink.sql.udtf.ParseUdtf';
```

- **CROSS JOIN**

Each row in the left table is joined with a row of data that is generated by the UDTF. If the UDTF does not generate any data for a row, the row is not returned.

```
select S.id, S.content, T.a, T.b, T.c
from input_stream as S,
lateral table(parseUdtf(content)) as T(a, b, c);
```

- **LEFT JOIN**

Each row in the left table is joined with a row of data that is generated by the UDTF. If the UDTF does not generate any data for a row, the UDTF fields in the row are filled with null.

 **Note** A `LEFT JOIN` statement that uses a UDTF must end with `on true`.

```
select S.id, S.content, T.a, T.b, T.c
from input_stream as S
left join lateral table(parseUdtf(content)) as T(a, b, c) on true;
```

Register and use resources

1. Log on to the Realtime Compute development platform.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Resources** tab.
4. In the upper-right corner of the **Resources** pane, click **Create Resource**.
5. In the **Upload Resource** dialog box, configure resource parameters.

Parameter	Description
Location	You can only upload JAR packages from your machine in the Realtime Compute for Apache Flink console.  Note The maximum size of a JAR package that can be uploaded from your machine is 300 MB. If the JAR package exceeds 300 MB, you must upload it to the Object Storage Service (OSS) bucket bound to your cluster or use APIs to upload it.

Parameter	Description
Resource	Click Upload Resource and select the resource that you want to reference.
Resource Name	Enter a name for the resource.
Resource Description	Enter a resource description.
Resource Type	Select the type of the resource. The type can be JAR, DICTIONARY, or PYTHON.

6. In the **Resources** pane, find the new resource, and move the pointer over **More** in the Actions column.
7. In the drop-down list, select **Reference**.
8. In the code editor, declare the UDX at the beginning. The following code is an example:

```
CREATE FUNCTION stringLengthUdf AS 'com.hjc.test.blink.sql.udx.StringLengthUdf';
```

Publish and use a UDTF

For more information about how to publish and use a UDTF, see [Publish a job](#) and [Start a job](#).

Example

```
-- UDTF str.split("\\|");
create function splitUdtf as 'com.hjc.test.blink.sql.udx.SplitUdtf';
create table sls_stream(
  a INT,
  b BIGINT,
  c VARCHAR
) with (
  type='sls',
  endPoint='yourEndpoint',
  accessKeyId='yourAccessKeyId',
  accessKeySecret='yourAccessSecret',
  startTime = '2017-07-04 00:00:00',
  project='yourProjectName',
  logStore='yourLogStoreName',
  consumerGroup='consumerGroupTest2'
);
-- Pass the c field to splitUdtf to generate table T(s) that consists of one column and multiple rows after splitting. In the table name T(s), s indicates the field name.
create view v1 as
select a,b,c,s
from sls_stream,
lateral table(splitUdtf(c)) as T(s);
create table rds_output(
  id INT,
  len BIGINT,
  content VARCHAR
) with (
  type='rds',
  url='yourDatabaseURL',
  tableName='yourDatabaseTableName',
  userName='yourDatabaseUserName',
  password='yourDatabasePassword'
);
insert into rds_output
select
  a,b,s
from v1;
```

3.1.4.12.5. Develop a UDX by using IntelliJ IDEA

This topic describes how to develop a user-defined extension (UDX) in Realtime Compute for Apache Flink by using IntelliJ IDEA. The development process includes building a development environment and referencing a UDX in a Realtime Compute for Apache Flink job.

Context

Notice

- Only Realtime Compute for Apache Flink in exclusive mode supports UDXs.
- We recommend that you use [IntelliJ IDEA](#) to develop a UDX.

Configure Maven

1. Download a Maven installation package.
 - i. Go to the [download page at the official Apache Maven website](#).
 - ii. Download apache-maven-3.5.3-bin.tar.gz.
 - iii. Decompress the downloaded package to a specified directory, such as `/Users/<userName>/Documents/maven`.
2. Configure environment variables.
 - i. In the terminal, run the `vim ~/.bash_profile` command.
 - ii. In the `.bash_profile` file, add the following commands:

```
export M2_HOME=/Users/<userName>/Documents/maven/apache-maven-3.5.3
export PATH=$PATH:$M2_HOME/bin
```

- iii. Save and close the `.bash_profile` file.
 - iv. Run the `source ~/.bash_profile` command for the configuration to take effect.
3. Run the `mvn -v` command to check whether the configuration takes effect.

If information similar to the following is displayed, the configuration takes effect:

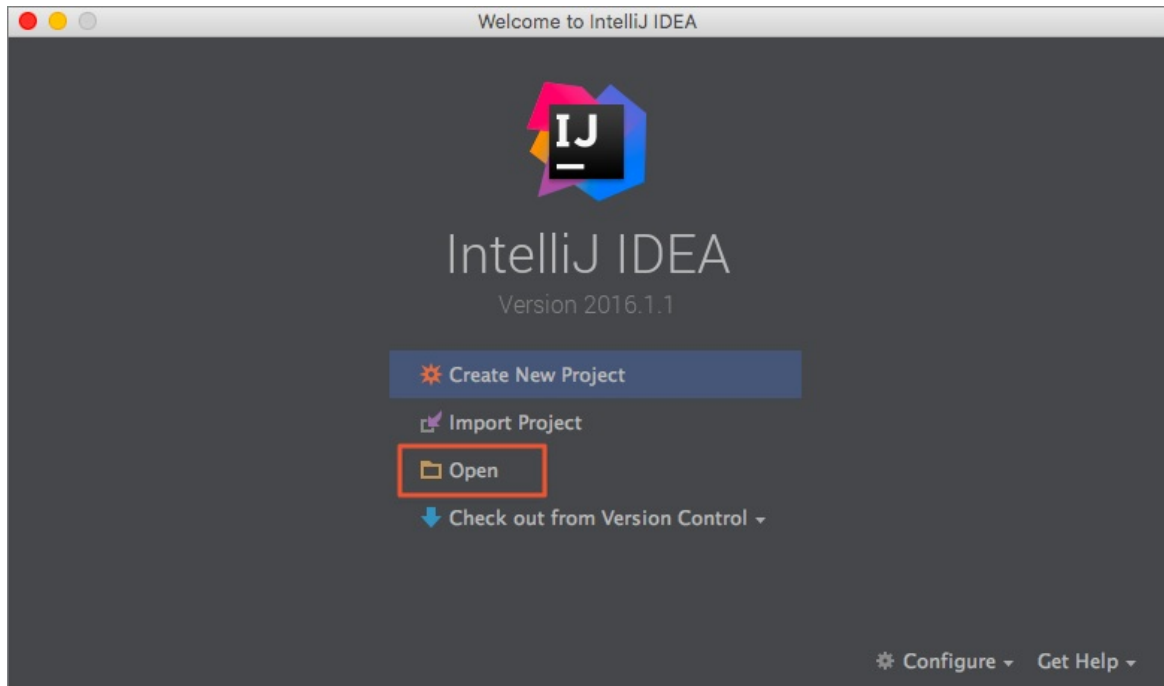
```
Apache Maven 3.5.0 (ff8f5e7444045639af65f6095c62210b5713f426; 2017-04-04T03:39:06+08:00)
Maven home: /Users/<userName>/Documents/maven/apache-maven-3.5.0
Java version: 1.8.0_121, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/jre
Default locale: zh_CN, platform encoding: UTF-8
OS name: "mac os x", version: "10.12.6", arch: "x86_64", family: "mac"
```

Build a development environment

1. Download the [UDX demo](#).
2. Decompress the downloaded package in a Linux operating system.

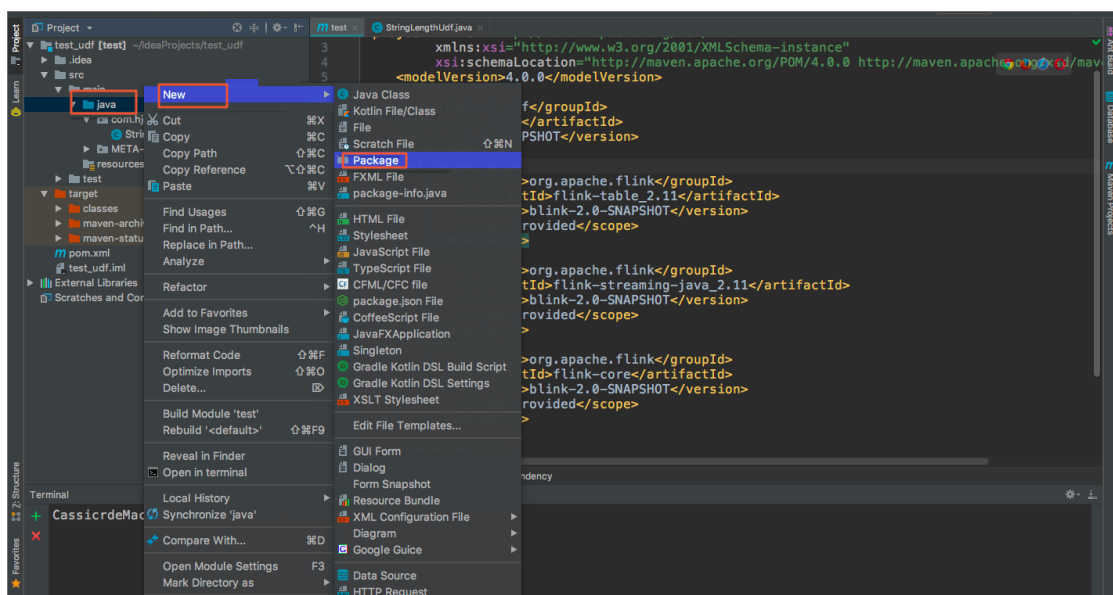
```
tar xzvf RealtimeCompute-udxDemo.gz
```

3. Open **IntelliJ IDEA** and click **Open** to open the demo.



Reference a JAR package in a job

1. Create a package.
 - i. Right-click `java` and navigate to **New > Package**.



- ii. In the **New Package** dialog box, enter a package name. In this example, a package named `com.hjc.test.blink.sql.udx` is created.
 - iii. Click **OK**.
 2. Create a class.
 - i. Right-click `com.hjc.test.blink.sql.udx` and navigate to **New > Java Class**.

- ii. In the **Create New Class** dialog box, enter a class name.

 **Note** Retain the default value **Class of Kind**.

- iii. Click **OK**.

3. Enter the following code in the class:

```
package com.hjc.test.blink.sql.udx;
import org.apache.flink.table.functions.FunctionContext;
import org.apache.flink.table.functions.ScalarFunction;
public class StringLengthUdf extends ScalarFunction {
    // The open method is optional.
    // To write the open method, you must add 'import org.apache.flink.table.functions.FunctionContext;
    ' to the code.
    @Override
    public void open(FunctionContext context) {
    }
    public long eval(String a) {
        return a == null ? 0 : a.length();
    }
    public long eval(String b, String c) {
        return eval(b) + eval(c);
    }
    // The close method is optional.
    @Override
    public void close() {
    }
}
```

4. In the terminal, run the `mvn package` or `mvn assembly:assembly` command to add the project to the JAR package.

 **Note**

- If you need to add a third-party dependency to the JAR package, run the `mvn assembly:assembly` command.
- The compiled JAR package is `RealtimeCompute-udxDemo/target/RTCompute-udx-1.0-SNAPSHOT.jar` or `RealtimeCompute-udxDemo/target/RTCompute-udx-1.0-SNAPSHOT-jar-with-dependencies.jar`. It contains a third-party dependency.

5. Reference the JAR package in a Realtime Compute for Apache Flink job.
 - i. Log on to the Realtime Compute development platform.
 - ii. In the top navigation bar, click **Development**
 - iii. In the left-side navigation pane, click the **Resources** tab.

- iv. In the upper-right corner of the **Resources** pane, click **Create Resource**.

Parameter	Description
Location	You can only upload JAR packages from your machine in the Realtime Compute for Apache Flink console.  Note The maximum size of a JAR package that can be uploaded from your machine is 300 MB. If the JAR package exceeds 300 MB, you must upload it to the Object Storage Service (OSS) bucket bound to your cluster or use APIs to upload it.
Resource	Click Upload Resource and select the resource that you want to reference.
Resource Name	Enter a name for the resource.
Resource Description	Enter a resource description.
Resource Type	Select the type of the resource. The type can be JAR, DICTIONARY, or PYTHON.

- v. In the **Resources** pane, find the new resource, and move the pointer over **More** in the Actions column.
- vi. In the drop-down list, select **Reference**.
- vii. In the code editor, declare the UDX at the beginning. The following code is an example:

```
CREATE FUNCTION stringLengthUdf AS 'com.hjc.test.blink.sql.udx.StringLengthUdf';
```


4.DataHub

4.1. Developer Guide

4.1.1. SDK overview

This topic describes the SDKs that are provided by DataHub.

Package name	Description
aliyun-sdk-datahub	The SDK is used to perform basic operations in DataHub. The SDK package provides API operations on basic objects in DataHub, including projects, topics, and shards.

DataHub provides SDKs for Java, Python, Go, and C++. The following sections describe how to use DataHub SDKs for Java and Python.

4.1.2. Prerequisites

4.1.2.1. Overview

Before you use MaxCompute SDKs, you must prepare the required environment.

- Create an Alibaba Cloud account that is authorized to access MaxCompute and obtain an AccessKey pair that consists of an AccessKey ID and an AccessKey secret.
- Obtain the endpoint of MaxCompute.

4.1.2.2. Obtain an AccessKey pair

4.1.2.2.1. Log on to the Apsara Uni-manager

Management Console

This topic describes how to log on to the Apsara Uni-manager Management Console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.

4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:

- It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the account and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.

- You have enabled MFA and bound an MFA device.

Enter a six-digit MFA authentication code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Operations Console User Guide*.

4.1.2.2.2. Obtain the AccessKey pair of an organization

This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only the operation administrators and level-1 organization administrators can obtain the AccessKey pair of an organization.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, find the organization that you want to add and click the  icon on the right of an organization.
5. Select **AccessKey** from the drop-down list.
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 of the level-1 organization use the AccessKey pair of the level-1 organization.

4.1.2.2.3. Obtain the AccessKey pair of an Alibaba Cloud account

To secure cloud resources, the system must verify the identity of visitors and ensure that they have the relevant permissions. You must obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account to access cloud resources. This topic describes how to obtain the AccessKey pair of an Alibaba Cloud account.

Procedure

1. Log on to the Apsara Uni-manager Management 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 your AccessKey pair.

Apsara Stack AccessKey Pair You must use the AccessKey pair when you access Apsara Stack resources.		
The AccessKey pair including the AccessKey ID and AccessKey secret is the credential for you to use Apsara Stack resources with full permissions. You must keep the AccessKey pair confidential.		
Region	AccessKey ID	AccessKey Secret
cn-qingdao-env4b-d01		Show

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

4.1.2.3. Obtain an endpoint

This topic describes how to obtain the endpoint of a DataHub machine in the Apsara Uni-manager Operations Console.

Context

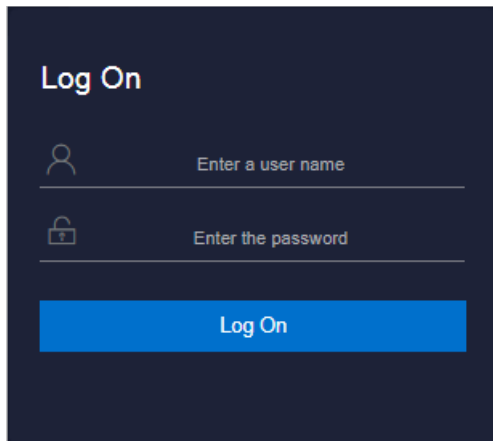
- The URL, username, and password that are used to log on to the Apsara Uni-manager Operations Console are obtained from the deployment engineers or administrators.

The URL of the Apsara Uni-manager Operations Console is in the following format: *ops.asconsole.intranet-domain-id.com*.

- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. Open your browser. In the address bar, enter the URL (*ops.asconsole.intranet-domain-id.com*). Then, 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** To obtain the username and password that are used to log on to the Apsara Uni-manager Operations Console, contact the deployment engineers or administrators.

If you log on to the Apsara Uni-manager Operations Console for the first time, you must change the password of your username.

To ensure the security of your account, make sure that the password meet the following requirements:

- The password contains uppercase or lowercase letters.
 - The password contains digits.
 - The password contains special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).
 - The password is 10 to 20 characters in length.
3. Click **Log On** to go to the Apsara Uni-manager Operations Console.
 4. In the top navigation bar of the Apsara Uni-manager Operations Console, click **O&M**. In the left-side navigation pane, choose **Product Management > Products**. In the **Apsara Stack O&M** section, click **Apsara Infrastructure Management Framework**.
 5. In the left-side navigation pane, choose **Operations > Machine Operations**. The **Machine Operations** page appears.
 6. Find the machine to which you want to log on, and click **Terminal** in the Actions column.

Machines

Project

All

Clusters

Enter a cluster name

Machines

Enter one or more hostnames/IP addresses

Batch Terminal

☐	Hostname	Clusters	Project	Region	Status	Machine Metrics	Actions
☐	</						

7. Run the following command on the machine to obtain JSON data:

```
curl -I "http://127.0.0.1:7070/api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration"

[admin@ ~]$ curl -I http://127.0.0.1:7070/api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration | grep frontend
100 3973 0 3973 0 0 122k 0 --:--:-- --:--:-- --:--:-- 122k "c.sr.id": "datahub-frontend",
  "c.sr.service_registration": "{\"datahub_frontend.endpoint\":\"datahub.cn-qingdao-env12-d01.dh.env12.shuguang.com\", \"datahub_frontend.port\":\"80,443\"}"
  "c.sr.service_registration": "{\"biggraph_dbname\":\"biggraph\", \"biggraph_dbpasswd\":\"rod2Cfec8Ruebeqd\", \"biggraph_dbport\":\"3122\", \"biggraph_dbuser\":\"biggraph\", \"biggraph_frontend_server.endpoint\":\"biggraph.cn-qingdao-env12-d01.odps.env12.ops.shuguang.com\", \"biggraph_frontend_server.httpsport\":\"443\", \"biggraph_frontend_server.port\":\"80\", \"biggraph_frontend_server_public.endpoint\":\"biggraph.cn-qingdao-env12-d01.odps.env12.shuguang.com\", \"biggraph_frontend_server_public.httpsport\":\"443\", \"biggraph_frontend_server_public.port\":\"80\", \"biggraph_project\":\"biggraph_internal_project\", \"cluster.name\":\"HybridOdpsCluster-A-20181110-d3cc\", \"quota.id\":\"9249\", \"quota.name\":\"biggraph_quota\"}"
  "c.sr.id": "odps-service-frontend",
  "c.sr.service_registration": "{\"odps_frontend_server.endpoint\":\"service.cn-qingdao-env12-d01.odps.env12.ops.shuguang.com\", \"odps_frontend_server.httpsport\":\"443\", \"odps_frontend_server.port\":\"80\", \"odps_frontend_server.vip\":\"\", \"odps_frontend_server_internet.endpoint\":\"service.cn-qingdao-env12-d01.odps.env12.shuguang.com\", \"odps_frontend_server_internet.httpsport\":\"443\", \"odps_frontend_server_internet.ip\":\"\", \"odps_frontend_server_internet.port\":\"80\", \"odps_frontend_server_public.endpoint\":\"service.cn-qingdao-env12-d01.odps.env12.shuguang.com\", \"odps_frontend_server_public.httpsport\":\"443\", \"odps_frontend_server_public.ip\":\"\", \"odps_frontend_server_public.port\":\"80\", \"tunnel_frontend_server.endpoint\":\"\"}"
```

8. Find the endpoint of *DataHub* based on the service ID (*datahub-frontend*) and the key value (*datahub_frontend.endpoint*) of *DataHub*.

```

[admin]
$curl -l /api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration | grep
datahub
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left   Speed
100 3973    0 3973    0    0 75893      0 --:--:-- --:--:-- --:--:-- 75893
ID
"c.sr.id": "datahub-fronten
d",
  "c.sr.service_registration": "{\n  "datahub_frontend.endpoint": "\n  "datahub.cn-qingdao-env12-d01.dh.env12.shugu
uang.com",\n  "datahub_frontend.port": "\n  80,443\n"}"
  "c.sr.id": "datahub-service",
  "c.sr.service_registration": "{\n  "admin_account.accesskey-id": "\n  FuZrpsHFw4pKFpCm",\n  "admin_account.access
key-secret": "\n  Hgppfxke5b0XTx2JDCnD9CrlDlTBwf",\n  "admin_account.id": "\n  1154035346989039",\n  "admin_account.passwor
d": "\n  Sqm0moaHnwyja06aqfw",\n  "admin_account.user": "\n  datahub_admin@aliyun.com",\n  "datahub_service_suffix": "\n  x",
  "rds_db_name": "\n  datahub_db",\n  "rds_host": "\n  datahub-db.mysql.minirds.env12.ops.shuguang.com",\n  "rds_passwd": "\n
ruevhgWzdy8Uc5vu",\n  "rds_port": "\n  3117",\n  "rds_user": "\n  datahub_db\n"}"
  "c.sr.id": "datahub-webconsole",
  "c.sr.service_registration": "{\n  "datahub_webconsole.endpoint": "\n  datahub.cn-qingdao-env12-d01.webconsole.
env12.shuguang.com",\n  "datahub_webconsole.port": "\n  80,443\n"}"
100 141k    0 141k    0    0 2652k      0 --:--:-- --:--:-- --:--:-- 133M

```

4.1.3. DataHub SDK for Java

4.1.3.1. Overview

You do not need to download the DataHub SDK for Java. You only need to configure the SDK as a Maven dependency in a pom.xml file. Add the following dependency configuration to the pom.xml file:

```

<dependency>
  <groupId>com.aliyun.datahub</groupId>
  <artifactId>aliyun-sdk-datahub</artifactId>
  <version>2.9.2-public</version>
</dependency>

```

Note If you use DataHub V3.8.1 or earlier, use the preceding dependency configuration, where the SDK version is 2.9.2.

```

<dependency>
  <groupId>com.aliyun.datahub</groupId>
  <artifactId>aliyun-sdk-datahub</artifactId>
  <version>2.13.1-public</version>
</dependency>

```

Note If you use DataHub V3.8.1 or later, use the preceding dependency configuration, where the SDK version is 2.13.1.

4.1.3.2. Preparations

To access DataHub, you must have an account that is authorized to access DataHub, the AccessKey for the account, and an endpoint of DataHub. If you are using Apsara Stack V2, obtain the endpoint from the CMDB system. If you are using Apsara Stack V3, obtain the endpoint from the Apsara Infrastructure Management Framework console.

The sample code is shown as follows:

```
String accessId = "Your AccessId";
String accessKey = "Your AccessKey";
String endpoint = "http://XXXXXX";
// Specify the endpoint you have obtained.
AliyunAccount account = new AliyunAccount(accessId, accessKey);
DatahubConfiguration conf = new DatahubConfiguration(account, endpoint);
```

The following example shows how to initialize a DataHub client:

```
DatahubClient client = new DatahubClient(conf);
```

 **Note** After the DataHub client is initialized, you can call the DatahubClient operation to access DataHub. You can use this client to perform all operations on DataHub because the client is thread-safe.

4.1.3.3. Ingest tuple data

4.1.3.3.1. Create a tuple topic

The sample code is shown as follows:

```
RecordSchema schema = new RecordSchema();
schema.addField(new Field("a", FieldType.STRING));
schema.addField(new Field("b", FieldType.BIGINT));
int shardCount = 5;
int lifeCycle = 3;
String topicName = "topic_example";
String topicDesc = "topic_example_desc";
client.createTopic(projectName, topicName, shardCount, lifeCycle, RecordType.TUPLE, schema, topicDesc);
// Wait until the shards are in Active status.
client.waitForShardReady(projectName, topicName);
```

 **Note** In the sample code, lifeCycle indicates the time-to-live of each record in the topic.

4.1.3.3.2. Obtain the list of shards

The sample code is shown as follows:

```
ListShardResult listShardResult = client.listShard(projectName, topicName);
```

4.1.3.3.3. Write data into DataHub

The sample code is shown as follows:

```

List<RecordEntry> recordEntries = new ArrayList<RecordEntry>();
// Write into DataHub starting from any active shard returned by listShardResult. In this example, start from
the first active shard.
String shardId = listShardResult.getShards().get(0).getShardId();
RecordEntry entry = new RecordEntry(schema);
entry.setString(0, "Test");
entry.setBigint(1, 5L);
entry.setShardId(shardId);
recordEntries.add(entry);
PutRecordsResult result = client.putRecords(projectName, topicName, recordEntries);
if (result.getFailedRecordCount() != 0) {
    List<ErrorEntry> errors = result.getFailedRecordError();
    // Process records failed to be written into DataHub.
    for (ErrorEntry e : result.getFailedRecordError()) {
        if (!e.getErrorCode().equals( "MalformedRecord" )) { // Retry the operation except that the MalformedRe
            cord, NoSuchProject, and NoSuchTopic errors occur.}
    }
}

```

 **Note** A request body size cannot exceed 4 MB. For more information, see Data ingestion methods.

4.1.3.3.4. Consume data

You can use this SDK to obtain the cursor that points to the first record to be consumed. The cursor type can be **OLDEST**, **LATEST**, or **SYSTEM_TIME**. **OLDEST**: the cursor that points to the earliest record in DataHub. **LATEST**: the cursor that points to the latest record in DataHub. **SYSTEM_TIME**: the cursor that points to the first record ingested after the specified time.

The sample code is shown as follows:


```
GetCursorResult cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
// If you want to obtain the cursor that points to the earliest record ingested during the last 24 hours, set CursorType to SYSTEM_TIME and set SYSTEM_TIME to System.currentTimeMillis() - 24 * 3600 * 1000 /* ms */.
int limit = 100;
String cursor = cursorRs.getCursor();
while (true) {
    try {
        GetBlobRecordsResult recordRs = client.getBlobRecords(projectName, topicName, shardId, cursor, limit);
        List<BlobRecordEntry> recordEntries = recordRs.getRecords();
        if (recordEntries.size() == 0) {
            // Data has not been updated. Try again later.
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        // The cursor that points to the next record to be consumed is obtained.
        cursor = recordRs.getNextCursor();
    } catch (InvalidCursorException ex) {
        // The cursor is invalid or has expired. Specify another record as the beginning of consumption.
        cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
        cursor = cursorRs.getCursor();
    } catch (DatahubClientException ex) {
        // An error occurred while obtaining the cursor. Try again.
        System.out.printf(ex.getMessage());
        ex.printStackTrace();
    }
}
```

4.1.3.4. Ingest blob data

4.1.3.4.1. Create a blob topic

The sample code is shown as follows:

```
int shardCount = 5;
int lifeCycle = 3;
String topicName = "topic_example";
String topicDesc = "topic_example_desc";
client.createTopic(projectName, topicName, shardCount, lifeCycle, RecordType.BLOB, topicDesc);
// Wait until the shards are in Active status.
client.waitForShardReady(projectName, topicName);
```

4.1.3.4.2. Obtain the list of shards

The sample code is shown as follows:

```
ListShardResult listShardResult = client.listShard(projectName, topicName);
```

4.1.3.4.3. Write data into DataHub

The sample code is shown as follows:

```
List<BlobRecordEntry> recordEntries = new ArrayList<BlobRecordEntry>();
// Write into DataHub starting from any active shard returned by listShardResult. In this example, start from
the first active shard.
String shardId = listShardResult.getShards().get(0).getShardId();
String data = String.valueOf(System.currentTimeMillis());
BlobRecordEntry entry = new BlobRecordEntry();
entry.setData(data.getBytes());
entry.setShardId(shardId);
recordEntries.add(entry);
PutBlobRecordsResult result = client.putBlobRecords(projectName, topicName, recordEntries);
if (result.getFailedRecordCount() != 0) {
    List<ErrorEntry> errors = result.getFailedRecordError();
    // Retry the operation except that the MalformedRecord, NoSuchProject, and NoSuchTopic errors occur.
}
```

 **Note** A request body size cannot exceed 4 MB.

4.1.3.4.4. Consume data

The sample code is shown as follows:

```
GetCursorResult cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
// If you want to obtain the cursor that points to the earliest record ingested during the last 24 hours, set CursorType to SYSTEM_TIME and set SYSTEM_TIME to System.currentTimeMillis() - 24 * 3600 * 1000 /* ms */.
int size = 100;
String endCursor = cursorRs.getCursor();
while (true) {
    try {
        GetBlobRecordsResult recordRs = client.getBlobRecords(projectName, topicName, shardId, cursor, limit);
        List<BlobRecordEntry> recordEntries = recordRs.getRecords();
        if(recordEntries.size() == 0) {
            // Data has not been updated. Try again later.
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        // The cursor that points to the next record to be consumed is obtained.
        cursor = recordRs.getNextCursor();
    } catch (InvalidCursorException ex) {
        // The cursor is invalid or has expired. Specify another record as the beginning of consumption.
        cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
        cursor = cursorRs.getCursor();
    } catch (DatahubClientException ex) {
        // An error occurred while obtaining the cursor. Try again.
        System.out.println(response.getMessage());
        e.printStackTrace();
    }
}
```

 **Note** Before being consumed, blob records that have been base64 encoded during ingestion must be decoded.

4.1.3.5. Data ingestion methods

4.1.3.5.1. Ingest data based on shard IDs

By using this method, you can write data into a specific shard and ensure that data is written into each shard in a specified order.

The sample code is shown as follows:

```
// Create a client.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Specify records to be ingested by DataHub.
RecordSchema schema = client.getTopic("projectName", "topicName").getRecordSchema();
List<RecordEntry> recordEntries = new ArrayList<>();
RecordEntry entry = new RecordEntry(schema);
for (int i=0; i<entry.getFieldCount(); i++) {
    entry.setBigint(i, 1);
}
// Write into DataHub starting from any active shard returned by listShardResult. In this example, start from
the first active shard.
String shardId = listShardResult.getShards().get(0).getShardId();
entry.setShardId(shardId);
recordEntries.add(entry);
// Write data into DataHub.
PutRecordsResult result = client.putRecords("projectName", "topicName", recordEntries);
```

4.1.3.5.2. Ingest data based on hash values

Specify a 128-bit hash value produced by MD5 message-digest algorithm. Use the hash value to map associated records to a shard based on the hash key range of the shards.

You cannot specify the ingestion order of records by using this method.

The sample code is shown as follows:

```
// Create a client.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Specify records to be ingested by DataHub.
RecordSchema schema = client.getTopic("projectName", "topicName").getRecordSchema();
List<RecordEntry> recordEntries = new ArrayList<>();
RecordEntry entry = new RecordEntry(schema);
for (int i=0; i<entry.getFieldCount(); i++) {
    entry.setBigint(i, 1);
}
entry.setHashKey("7FFFFFFFFFFFFFFD7FFFFFFFFFFFFFFD");
recordEntries.add(entry);
// Write the data into DataHub.
PutRecordsResult result = client.putRecords("projectName", "topicName", recordEntries);
```

4.1.3.5.3. Ingest data based on partition keys

Specify a partition key as a string. DataHub creates an MD5 hash from the string and maps the records to shards based on the hash key range of the shards.

You do not need to provide hash values by using this method.

The sample code is shown as follows:

```
// Create a client.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Specify records to be ingested by DataHub.
RecordSchema schema = client.getTopic("projectName", "topicName").getRecordSchema();
List<RecordEntry> recordEntries = new ArrayList<>();
RecordEntry entry = new RecordEntry(schema);
for (int i=0; i<entry.getFieldCount(); i++) {
    entry.setBigint(i, 1);
}
entry.setPartitionKey("TestPartitionKey");
recordEntries.add(entry);
// Write the data into DataHub.
PutRecordsResult result = client.putRecords("projectName", "topicName", recordEntries);
```

4.1.3.6. Create a DataConnector

4.1.3.6.1. Create a DataConnector for MaxCompute

The sample code is shown as follows:

```

public void createDataConnector () {
    // Create a DataConnector for MaxCompute
    // Configure the destination MaxCompute table.
    String odpsProject = "datahub_test";
    String odpsTable = "test_table";
    String odpsEndpoint = "http://service-all.ext.odps.example.com/api";
    String tunnelEndpoint = "http://dt-all.ext.odps.example.com";
    OdpsDesc odpsDesc = new OdpsDesc();
    odpsDesc.setProject(odpsProject);
    odpsDesc.setTable(odpsTable);
    odpsDesc.setOdpsEndpoint(odpsEndpoint);
    odpsDesc.setTunnelEndpoint(tunnelEndpoint);
    odpsDesc.setAccessId(accessId);
    odpsDesc.setAccessKey(accessKey);
    odpsDesc.setPartitionMode(OdpsDesc.PartitionMode.USER_DEFINE);
    // Select some or all columns in the topic to be archived to MaxCompute. The MaxCompute table must have the same columns in the same order as selected.
    List<String> jobIds = new ArrayList<String>();
    columnFields.add("f1");
    // By default, the partition mode is set to UserDefine.
    // If you set the partition mode to SYSTEM_TIME or EVENT_TIME, you must complete the following configuration.
    // The configuration begins.
    int timeRange = 15; // The partitioning interval, in minutes. Minimum value: 15.
    odpsDesc.setPartitionMode(OdpsDesc.PartitionMode.SYSTEM_TIME);
    odpsDesc.setTimeRange(timeRange);
    Map<String, String> partitionConfig = new LinkedHashMap<String, String>();
    // The partition key must be in the %Y%m%d%H%M format.
    partitionConfig.put("pt", "%Y%m%d");
    partitionConfig.put("ct", "%H%M");
    odpsDesc.setPartitionConfig(partitionConfig);
    // The configuration ends.
    client.createDataConnector(projectName, topicName, ConnectorType.SINK_ODPS, columnFields, odpsDesc);
    // You can obtain the DataConnector status periodically, such as every 15 minutes, to check whether all shards are being archived properly to the destination platform.
    String shard = "0";
    GetDataConnectorShardStatusResult getDataConnectorShardStatusResult =
        client.getDataConnectorShardStatus(projectName, topicName, ConnectorType.SINK_ODPS, shard);
    System.out.println(getDataConnectorShardStatusResult.getCurSequence());
    System.out.println(getDataConnectorShardStatusResult.getLastErrorMessage());
}

```

4.1.3.6.2. Create a DataConnector for AnalyticDB or ApsaraDB RDS for MySQL

Sample code:

```
public void createADSDDataConnector () {
    // Create a DataConnector for AnalyticDB or ApsaraDB RDS for MySQL.
    // Configure the destination table.
    String dbHost = "127.0.0.1";
    int dbPort = 3306;
    String dbName = "db";
    String user = "123";
    String password = "123";
    String tableName = "table";
    DatabaseDesc desc = new DatabaseDesc();
    desc.setHost(dbHost);
    desc.setPort(dbPort);
    desc.setDatabase(dbName);
    desc.setUser(user);
    desc.setPassword(password);
    desc.setTable(tableName);
    // The maximum size of records that can be written to DataHub at a time. Unit: Byte.
    desc.setMaxCommitSize(512L);
    // Specify whether to ignore errors and continue the process.
    desc.setIgnore(true);
    // Select several or all columns in the topic to be synchronized to AnalyticDB or ApsaraDB RDS for MySQL. The destination table must have the columns in the same order as selected.
    List<String> columnFields = new ArrayList<String>();
    columnFields.add("f1");
    client.createDataConnector(projectName, topicName, ConnectorType.SINK_ADS, columnFields, desc);
    // Alternative: client.createDataConnector(projectName, topicName, ConnectorType.SINK_MYSQL, columnFields, odpsDesc);
    // You can obtain the DataConnector status periodically, such as every 15 minutes, to check whether all shards are being properly synchronized to the destination table.
    String shard = "0";
    GetDataConnectorShardStatusResult getDataConnectorShardStatusResult =
        client.getDataConnectorShardStatus(projectName, topicName, ConnectorType.SINK_ADS, shard);
    System.out.println(getDataConnectorShardStatusResult.getCurSequence());
    System.out.println(getDataConnectorShardStatusResult.getLastErrorMessage());
}
```

 **Note** You cannot create DataConnectors for services that reside in virtual private clouds (VPCs) by using the SDK. You must create this type of DataConnectors in the DataHub console.

4.1.4. DataHub SDK for Python

4.1.4.1. Install DataHub SDK for Python

Run the following commands to install DataHub SDK for Python.

Quick installation

```
$ sudo pip install pydatahub
```

Install by using source code

```
$ git clone https://github.com/aliyun/aliyun-datahub-sdk-python.git
$ cd aliyun-datahub-sdk-python
$ sudo python setup.py install
```

Verify the installation

```
$ python -c "from datahub import DataHub"
```

4.1.4.2. Preparations

To access DataHub, you must have an account that is authorized to access DataHub, the AccessKey for the account, and an endpoint of DataHub. If you are using Apsara Stack V2, obtain the endpoint from the CMDB system. If you are using Apsara Stack V3, obtain the endpoint from the Apsara Infrastructure Management Framework console.

The following example shows how to initialize DataHub configurations:

```
import sys
import traceback
from datahub import DataHub
from datahub.utils import Config
from datahub.models import Topic, RecordType, FieldType, RecordSchema, BlobRecord, TupleRecord, CursorType
from datahub.errors import DatahubException, ObjectAlreadyExistException
access_id = ***your access id***
access_key = ***your access key***
endpoint = ***your datahub server endpoint***
dh = DataHub(access_id, access_key, endpoint)
```

4.1.4.3. Create topics

Create a tuple topic

You must specify a schema for a tuple topic. The following data types are supported:

Tuple data types

Type	Description	Value range
Bigint	An 8-byte signed integer.  Note Do not use the minimum value (-9223372036854775808) because this is a system reserved value.	-9223372036854775807 to 9223372036854775807

Type	Description	Value range
String	A string. Only UTF-8 encoding is supported.	The size of a string must not exceed 1 MB.
Boolean	One of two possible values.	Valid values: True and False, true and false, or 0 and 1.
Double	An 8-byte double-precision floating point.	$-1.0 \cdot 10^{308}$ to $1.0 \cdot 10^{308}$
TimeStamp	A timestamp.	It is accurate to milliseconds.

The sample code is shown as follows:

```
topic = Topic(name=topic_name)
topic.project_name = project_name
topic.shard_count = 3
topic.life_cycle = 7
topic.record_type = RecordType.TUPLE
topic.record_schema = RecordSchema.from_lists(['bigint_field', 'string_field', 'double_field', 'bool_field', 'time_field'], [FieldType.BIGINT, FieldType.STRING, FieldType.DOUBLE, FieldType.BOOLEAN, FieldType.TIMESTAMP])
try:
    dh.create_topic(topic)
    print "create topic success!"
    print "=====\n\n"
except ObjectAlreadyExistsException, e:
    print "topic already exist!"
    print "=====\n\n"
except Exception, e:
    print traceback.format_exc()
    sys.exit(-1)
```

Create a blob topic

In a blob topic, a chunk of binary data is stored as a record. Records written into DataHub are Base64 encoded.

The sample code is shown as follows:

```

topic = Topic(name=topic_name)
topic.project_name = project_name
topic.shard_count = 3
topic.life_cycle = 7
topic.record_type = RecordType.BLOB
try:
    dh.create_topic(topic)
    print "create topic success!"
    print "=====\n\n"
except ObjectAlreadyExistsException, e:
    print "topic already exist!"
    print "=====\n\n"
except Exception, e:
    print traceback.format_exc()
    sys.exit(-1)

```

4.1.4.4. Write records into DataHub and subscribe applications to records

Obtain the list of shards

You can use this SDK to obtain all shards in the topic.

```
shards = dh.list_shards(project_name, topic_name)
```

 **Note** A list is returned and each entry in the list describes a shard in terms of the ID, status, starting hash key, and ending hash key.

Write records into DataHub

You can use this SDK to write records into a topic in DataHub.

```
failed_indexes = dh.put_records(project_name, topic_name, records)
```

 **Note** The records request parameter is a list and each entry in the list represents a record. All records must be of the same data type, either tuple or blob. The subscript of the records that fail to be written into DataHub are returned.

The following sample code shows how to write tuple records into DataHub:

```
try:
    #Wait until all shards are in Active status.
    dh.wait_shards_ready(project_name, topic_name)
    print "shards all ready!!!"
    print "=====\n\n"
    topic = dh.get_topic(topic_name, project_name)
    print "get topic suc! topic=%s" % str(topic)
    if topic.record_type != RecordType.TUPLE:
        print "topic type illegal!"
        sys.exit(-1)
    print "=====\n\n"
    shards = dh.list_shards(project_name, topic_name)
    for shard in shards:
        print shard
    print "=====\n\n"
    records = []
    record0 = TupleRecord(schema=topic.record_schema, values=[1, 'yc1', 10.01, True, 1455869335000000])
    record0.shard_id = shards[0].shard_id
    record0.put_attribute('AK', '47')
    records.append(record0)
    record1 = TupleRecord(schema=topic.record_schema)
    record1['bigint_field'] = 2
    record1['string_field'] = 'yc2'
    record1['double_field'] = 10.02
    record1['bool_field'] = False
    record1['time_field'] = 1455869335000011
    record1.shard_id = shards[1].shard_id
    records.append(record1)
    record2 = TupleRecord(schema=topic.record_schema)
    record2['bigint_field'] = 3
    record2['string_field'] = 'yc3'
    record2['double_field'] = 10.03
    record2['bool_field'] = False
    record2['time_field'] = 1455869335000013
    record2.shard_id = shards[2].shard_id
    records.append(record2)
    failed_indexes = dh.put_records(project_name, topic_name, records)
    print "put tuple %d records, failed list: %s" %(len(records), failed_indexes)
    #If you specify the failed_indexes parameter, we recommend that you retry writing the failed records into DataHub.
    print "=====\n\n"
except DatahubException, e:
    print traceback.format_exc()
    sys.exit(-1)
else:
    sys.exit(-1)
```

Obtain the cursor of a record

You can use this SDK to obtain the cursor of a record. Valid values: **OLDEST**, **LATEST**, and **SYSTEM_TIME**.

- **OLDEST**: the cursor that points to the earliest record in DataHub.

- LATEST: the cursor that points to the latest record in DataHub.
- SYSTEM_TIME: the cursor that points to the first record ingested after the specified time.

```
cursor = dh.get_cursor(project_name, topic_name, CursorType.OLDEST, shard_id)
```

 **Note** If you select SYSTEM_TIME, this SDK returns the cursor that points to the record ingested at the specified time. If no record is ingested at the specified time, the first record ingested after the time is returned.

Consume records

You can use this SDK to specify the cursor that points to a record where the consumption begins and specify the upper limit of records to be consumed. If fewer records exist than the specified limit, the actual number of records are returned.

```
dh.get_records(topic, shard_id, cursor, 10)
```

The following sample code shows how to consume tuple records:

```
try:
    #Wait until all shards are in Active status.
    dh.wait_shards_ready(project_name, topic_name)
    print "shards all ready!!!"
    print "=====\n\n"
    topic = dh.get_topic(topic_name, project_name)
    print "get topic suc! topic=%s" % str(topic)
    if topic.record_type != RecordType.TUPLE:
        print "topic type illegal!"
        sys.exit(-1)
    print "=====\n\n"
    cursor = dh.get_cursor(project_name, topic_name, CursorType.OLDEST, '0')
    while True:
        (record_list, record_num, next_cursor) = dh.get_records(topic, '0', cursor, 10)
        for record in record_list:
            print record
        if 0 == record_num:
            time.sleep(1)
            cursor = next_cursor
    except DatahubException, e:
        print traceback.format_exc()
        sys.exit(-1)
    else:
        sys.exit(-1)
```

4.1.5. Scalability

4.1.5.1. Scenarios

You can increase or decrease the number of shards in a topic according to the service load.

For example, if the topic throughput cannot handle a surge in the service load during Double 11, you can split existing shards to up to 256 to increase the throughput to 256 MB/s.

As the service load decreases after Double 11, you can reduce the number of shards as needed by performing the merge operation.

4.1.5.2. Obtain shard details

You can call the ListShard operation to obtain details about all shards.

The sample response is shown as follows:

```
{
  "ShardId": "string",
  "State": "string",
  "ClosedTime": uint64,
  "BeginHashKey": "string",
  "EndHashKey": "string",
  "ParentShardIds": [string,string,],
  "LeftShardId": "string",
  "RightShardId": "string"
}
```

4.1.5.3. Split a shard

You can split a shard by using an SDK or in the console. You can specify the shard by setting a shard ID and a 128-bit hash key value. After the shard is split into two child shards, the IDs and key values of the child shards are returned. The status of the parent shard is changed to Deactivated.

For example, before the split operation, the parent shard details are shown as follows:

```
ShardId:0 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:FFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
```

Split the shard by using an SDK:

```
String shardId = "0";
SplitShardRequest req = new SplitShardRequest(projectName, topicName, shardId, "AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA");
SplitShardResult resp = client.splitShard(req);
```

After the split operation, the parent shard turns into the following three shards:

```
ShardId:0 Status:Deactivated BeginHashKey:00000000000000000000000000000000 EndHashKey:FFFFFFF
FFFFFFFFFFFFFFFFF
ShardId:1 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:AAAAAAA
AAAAAAAAAAAAAAAA
ShardId:2 Status:ACTIVE BeginHashKey:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA EndHashKey:FFFFFFF
FFFFFFFFFFFFFFFF
```

4.1.5.4. Merge shards

You can merge two adjacent shards by using an SDK or in the console. Shards are considered adjacent if the union of the hash key ranges for the two shards forms a contiguous set with no gaps. After the two shards are merged, the ID and key value of the new shard are returned. The statuses of the two parent shards change to Deactivated.

For example, before the merge operation, the details of the two parent shards are as follows:

```
ShardId:0 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF
ShardId:1 Status:ACTIVE BeginHashKey:7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF EndHashKey:FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF
```

Merge the two shards by using an SDK:

```
String shardId = "0";
String adjacentShardId = "1";
MergeShardRequest req = new MergeShardRequest(projectName, topicName, shardId, adjacentShardId);
MergeShardResult resp = client.mergeShard(req);
```

After the merge operation, the parent shards turn into the following three shards:

```
ShardId:0 Status:Deactivated BeginHashKey:00000000000000000000000000000000 EndHashKey:7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF
ShardId:1 Status:Deactivated BeginHashKey:7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF EndHashKey:FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF
ShardId:2 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF
```

Note

- After merge or split operations are performed, the statuses of the parent shards change to Deactivated. New data cannot be written into deactivated shards, whereas existing data in deactivated shards can still be consumed. A deactivated shard cannot be split into two shards or merged with any other shards. When the time-to-live of the records in a deactivated shard expires, the shard is deleted.
- If a DataConnector is configured for a deactivated shard, the DataConnector will be stopped after all records in the shard are copied to the destination platform. The DataConnector is automatically deleted after the shard is deleted.
- The new shard can only be used after its status is changed to Active. It requires less than five seconds for the status to change to Active.

4.1.6. Insert a column to a topic schema

4.1.6.1. Insert a column

This section describes how to insert columns to a topic schema.

The sample code is shown as follows:

```
// Specify a DataHub account.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Insert a bigint column to the topic schema.
client.appendField(new AppendFieldRequest(projectName, topicName, new Field("test", FieldType.BIGINT))
);
// If you have created a DataConnector for MaxCompute, call the following operation.
client.appendDataConnectorField(new AppendDataConnectorFieldRequest(projectName, topicName, ConnectorType.SINK_ODPS, "test"));
}
```

 Note

- DataHub supports inserting columns only. Modifying or deleting columns is not supported.
- The operation can only be performed on a DataConnector for MaxCompute: **Insert a column in MaxCompute > Insert a corresponding column in DataHub > Modify the corresponding DataConnector configuration.**