

流计算 Oceanus API 文档



腾讯云

【版权声明】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

参数类型

日志相关接口

查询作业实例启动日志

资源相关接口

创建资源接口

删除资源接口

获取资源关联作业信息

描述资源接口

描述系统资源接口

资源配置相关接口

创建资源配置接口

删除资源版本

描述资源配置接口

元数据相关接口

查询元数据表

权限相关接口

创建工作空间

删除工作空间

修改工作空间

授权工作空间列表

集群接口

查询集群

作业接口

检查快照是否可用

复制作业

删除作业

获取指定作业的事件

查找Savepoint列表

更新作业

运行作业

停止作业

触发Savepoint

新建作业

查询作业

执行Statement

查询Statement执行结果

获取作业运行时的信息

作业配置接口

创建作业配置

删除作业配置

查询作业配置

其他接口

创建文件夹

删除文件夹

删除作业表配置

自定义树状结构
拖拽文件夹
查询树状结构资源列表
查询文件夹
数据结构
错误码

API 文档
更新历史

最近更新时间：2025-06-05 01:51:06

第 69 次发布

发布时间：2025-06-05 01:50:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeJobEvents](#)
 - 新增出参：Versions

第 68 次发布

发布时间：2025-05-28 01:23:41

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeJobs](#)
 - 新增入参：ConnectorOptions

新增数据结构：

- [ResourceRefLatest](#)
- [Setats](#)
- [SetatsCvmInfo](#)
- [SetatsDisk](#)
- [Warehouse](#)

修改数据结构：

- [Cluster](#)
 - 新增成员：Setats

第 67 次发布

发布时间：2025-03-12 01:22:00

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)
 - 新增入参：OpenJobDefaultAlarm

修改数据结构：

- [JobV1](#)
 - 新增成员：OpenJobDefaultAlarm, ProgressDesc
- [TreeJobSets](#)
 - 新增成员：DecodeSqlCode

第 66 次发布

发布时间：2024-12-30 01:49:42

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：JobManagerCpu, JobManagerMem, TaskManagerCpu, TaskManagerMem
- [DescribeResources](#)
 - 新增入参：SystemResource
- [DescribeSystemResources](#)
 - 新增入参：WorkSpaceId

修改数据结构：

- [Cluster](#)
 - 新增成员：TotalCpu, TotalMem, RunningCpu, RunningMem
- [ClusterGroupSetItem](#)
 - 新增成员：TotalCpu, TotalMem, RunningCpu, RunningMem
- [JobConfig](#)
 - 新增成员：JobManagerCpu, JobManagerMem, TaskManagerCpu, TaskManagerMem, JobConfigItem
- [ResourceItem](#)
 - 新增成员：Connector, ConnectorVersion, ConnectionMethod, RelatedResourceId, Icon, ConnectorName, ConnectorUrl
 - 修改成员：ResourceId, Name, ResourceType, ResourceLoc, Region, Appld, OwnerUin, CreatorUin, CreateTime, UpdateTime, LatestResourceConfigVersion, Remark, VersionCount, RefJobCount, IsJobRun, FileName, WorkSpaceId, RefJobStatusCountSet
- [ResourceRefDetail](#)
 - 新增成员：Connector
- [SqlGatewayItem](#)
 - 新增成员：Cpu, Mem
- [SubEks](#)
 - 新增成员：TotalCpu, TotalMem, RunningCpu, RunningMem
- [SystemResourceItem](#)
 - 新增成员：SystemProvide
- [TreeJobSets](#)
 - 新增成员：ScalingType, RunningCpu, RunningMem

第 65 次发布

发布时间：2024-11-28 01:23:56

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Cluster](#)
 - 新增成员：CrossTenantEniMode

第 64 次发布

发布时间：2024-11-05 02:01:14

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [SessionClusterRefItem](#)

修改数据结构：

- [ClusterSession](#)
 - 新增成员：ResourceRefs, JobManagerCpu, JobManagerMem, TaskManagerCpu, TaskManagerMem
- [JobV1](#)
 - 新增成员：ScalingType, RunningCpu, RunningMem

第 63 次发布

发布时间：2024-10-14 01:26:06

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeJobRuntimeInfo](#)

新增数据结构：

- [JobRuntimeInfo](#)

第 62 次发布

发布时间：2024-09-04 01:54:37

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeleteJobs](#)
 - 新增入参：JobNames

第 61 次发布

发布时间：2024-09-03 02:13:22

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Cluster](#)
 - 新增成员：MemRatio

第 60 次发布

发布时间：2024-07-23 01:53:12

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RunJobDescription](#)
 - 新增成员：KafkaScanMode

第 59 次发布

发布时间：2024-07-19 02:08:08

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Cluster](#)
 - 新增成员：BillingResourceMode
- [ClusterGroupSetItem](#)
 - 新增成员：BillingResourceMode

第 58 次发布

发布时间：2024-07-15 01:16:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：FlinkVersion

第 57 次发布

发布时间：2024-07-12 01:21:39

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [JobConfig](#)
 - 新增成员：FlinkVersion
- [SlotSharingGroup](#)
 - 新增成员：Configuration

第 56 次发布

发布时间：2024-06-05 02:02:33

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Cluster](#)
 - 新增成员：ResourceType

第 55 次发布

发布时间：2024-05-31 01:58:53

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [SubEks](#)

修改数据结构：

- [Cluster](#)
 - 新增成员：Type, SubEks, AgentSerialId
- [ClusterGroupSetItem](#)
 - 新增成员：SubEks
 - 修改成员：RunningCu, PayMode

第 54 次发布

发布时间：2024-03-08 01:20:43

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)

- 新增入参：Description
- [ModifyJob](#)
 - 新增入参：Description

修改数据结构：

- [JobV1](#)
 - 新增成员：Description

第 53 次发布

发布时间：2024-01-19 01:18:01

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CopyJobs](#)
 - 新增入参：TargetWorkspaceId

修改数据结构：

- [JobConfig](#)
 - 新增成员：IndexName, WorkspaceName

第 52 次发布

发布时间：2024-01-01 00:15:04

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeJobs](#)
 - 新增入参：ExtraResult

新增数据结构：

- [JobEventInfo](#)

修改数据结构：

- [JobV1](#)
 - 新增成员：EventInfo

第 51 次发布

发布时间：2023-12-29 01:21:17

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetMetaTable](#)

第 50 次发布

发布时间：2023-12-20 01:57:04

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeJobEvents](#)

新增数据结构：

- [JobEvent](#)

第 49 次发布

发布时间：2023-12-04 15:28:57

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeFolder](#)

新增数据结构：

- [SubFolderInfo](#)

第 48 次发布

发布时间：2023-12-04 00:15:25

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [JobV1](#)
 - 修改成员：Tags

第 47 次发布

发布时间：2023-11-28 01:20:57

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：EsServerlessIndex, EsServerlessSpace

第 46 次发布

发布时间：2023-11-27 00:14:59

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [JobConfig](#)
 - 新增成员：EsServerlessIndex, EsServerlessSpace

第 45 次发布

发布时间：2023-11-16 01:25:38

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeTreeResources](#)
 - 新增入参：Filters, Offset, Limit

第 44 次发布

发布时间：2023-10-24 01:20:07

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：TraceModeOn, TraceModeConfiguration, CheckpointRetainedNum, JobGraph
- [DescribeJobSavepoint](#)
 - 新增入参：RecordTypes
- [DescribeTreeJobs](#)
 - 新增入参：Filters
 - 新增出参：ParentId, Id, Name, JobSet, Children

新增数据结构：

- [DescribeTreeJobsRsp](#)
- [TraceModeConfiguration](#)
- [TreeJobSets](#)

修改数据结构：

- [Cluster](#)
 - 新增成员：WebUIType
- [JobConfig](#)
 - 新增成员：TraceModeOn, TraceModeConfiguration, CheckpointRetainedNum, JobGraph
 - 修改成员：ClazzLevels, ExpertModeOn, ExpertModeConfiguration

第 43 次发布

发布时间：2023-10-23 00:16:17

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteWorkSpace](#)

修改数据结构：

- [ClusterSession](#)
 - 新增成员：ClusterGroupSerialId, AppId, OwnerUin, CreatorUin, Region, Zone, Status, CuNum, FlinkVersion, WebUIUrl, Properties, JobManagerCuSpec, TaskManagerCuSpec, TaskManagerNum, CreateTime, UpdateTime

第 42 次发布

发布时间：2023-10-19 16:11:52

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateWorkSpace](#)
- [DeleteFolders](#)

- [DeleteJobConfigs](#)
- [ModifyFolder](#)
- [ModifyWorkSpace](#)

第 41 次发布

发布时间：2023-10-18 02:29:23

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [FetchSqlGatewayStatementResult](#)
 - 新增入参：ClusterId, SessionId, OperationHandleId, ResultUri
 - 新增出参：ErrorMessage, ResultType, IsQueryResult, ResultKind, Results, NextResultUri
- [RunSqlGatewayStatement](#)
 - 新增入参：ClusterId, Sql, SessionId
 - 新增出参：ErrorMessage, SessionId, OperationHandleId

新增数据结构：

- [LogicalType](#)
- [ResultColumn](#)
- [ResultData](#)
- [StatementResult](#)

第 40 次发布

发布时间：2023-10-17 01:22:49

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [FetchSqlGatewayStatementResult](#)
- [RunSqlGatewayStatement](#)

第 39 次发布

发布时间：2023-10-12 02:20:57

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [GatewayRefItem](#)
- [SqlGatewayItem](#)

修改数据结构：

- [Cluster](#)
 - 新增成员：SqlGateways

第 38 次发布

发布时间：2023-09-06 02:10:27

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RunJobDescription](#)
 - 新增成员：CustomTimestamp

第 37 次发布

发布时间：2023-08-31 01:20:56

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [Order](#)

修改数据结构：

- [Cluster](#)
 - 新增成员：ArchGeneration, ClusterType, Orders
 - 修改成员：ClusterSessions

第 36 次发布

发布时间：2023-08-01 01:34:42

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：ClazzLevels, ExpertModeOn, ExpertModeConfiguration

新增数据结构：

- [ClazzLevel](#)
- [ExpertModeConfiguration](#)
- [JobGraph](#)
- [JobGraphEdge](#)
- [JobGraphNode](#)
- [NodeConfig](#)
- [SlotSharingGroup](#)
- [SlotSharingGroupSpec](#)

修改数据结构：

- [JobConfig](#)
 - 新增成员：ClazzLevels, ExpertModeOn, ExpertModeConfiguration

第 35 次发布

发布时间：2023-07-17 01:29:44

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeJobSubmissionLog](#)

新增数据结构：

- [JobInstanceForSubmissionLog](#)
- [LogContent](#)

第 34 次发布

发布时间：2023-06-19 01:30:56

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeWorkSpaces](#)

新增数据结构：

- [ClusterGroupSetItem](#)
- [RoleAuth](#)
- [WorkspaceSetItem](#)

第 33 次发布

发布时间：2023-05-29 01:34:54

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)
 - 新增入参：Tags

修改数据结构：

- [JobV1](#)
 - 新增成员：Tags
- [TreeResourceItem](#)
 - 新增成员：RefJobStatusCountSet

第 32 次发布

发布时间：2023-05-12 01:39:08

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [ClusterSession](#)

修改数据结构：

- [Cluster](#)
 - 新增成员：ClusterSessions

第 31 次发布

发布时间：2023-05-10 01:38:48

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [CopyJobItem](#)
 - 新增成员：SourceId, TargetClusterId, SourceName, TargetName, TargetFolderId, JobType
- [CopyJobResult](#)
 - 新增成员：JobId, JobName, TargetJobName, TargetJobId, Message, Result, ClusterName, ClusterId, JobType

第 30 次发布

发布时间：2023-03-09 01:40:49

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [RefJobStatusCountItem](#)

修改数据结构：

- [ResourceConfigItem](#)
 - 新增成员：RefJobStatusCountSet
- [ResourceItem](#)
 - 新增成员：IsJobRun, FileName, WorkSpaceId, RefJobStatusCountSet
 - 修改成员：ResourceId, Name, ResourceType, ResourceLoc, Region, Appld, OwnerUin, CreatorUin, CreateTime, UpdateTime, LatestResourceConfigVersion, Remark, VersionCount, RefJobCount
- [RunJobDescription](#)
 - 新增成员：UseOldSystemConnector

第 29 次发布

发布时间：2023-02-21 01:54:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeTreeResources](#)
 - 新增出参：ParentId, Id, Name, Items, Children, TotalCount

新增数据结构：

- [DescribeTreeResourcesRsp](#)
- [TreeResourceItem](#)

第 28 次发布

发布时间：2023-01-13 01:40:40

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Savepoint](#)
 - 新增成员：TimeConsuming, PathStatus
 - 修改成员：Id, VersionId, Status, CreateTime, UpdateTime, Path, Size, RecordType, JobRuntimeId, Description, Timeout, SerialId

第 27 次发布

发布时间：2022-10-27 06:45:50

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [WorkSpaceClusterItem](#)
 - 新增成员：ProjectId, ProjectIdStr

第 26 次发布

发布时间：2022-09-27 06:36:12

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Cluster](#)
 - 新增成员：IsNeedManageNode

第 25 次发布

发布时间：2022-08-30 06:43:16

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：AutoRecover

修改数据结构：

- [JobConfig](#)
 - 新增成员：AutoRecover, LogLevel

第 24 次发布

发布时间：2022-08-25 06:41:36

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeTreeResources](#)

第 23 次发布

发布时间：2022-07-26 06:19:20

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CopyJobs](#)
 - 新增入参：JobItems, WorkSpaceId
 - 新增出参：SuccessCount, FailCount, CopyJobsResults
- [CreateFolder](#)
 - 新增入参：FolderName, ParentId
 - 新增出参：FolderId

新增数据结构：

- [CopyJobItem](#)
- [CopyJobResult](#)

第 22 次发布

发布时间：2022-07-25 06:12:04

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CopyJobs](#)
- [CreateFolder](#)

第 21 次发布

发布时间：2022-07-13 06:14:33

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteJobs](#)
- [DescribeClusters](#)
- [DescribeTreeJobs](#)

新增数据结构：

- [CCN](#)
- [Cluster](#)
- [ClusterVersion](#)
- [Tag](#)
- [WorkSpaceClusterItem](#)

第 20 次发布

发布时间：2022-07-04 06:14:12

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：LogLevel

第 19 次发布

发布时间：2022-06-15 06:12:20

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ModifyJob](#)

第 18 次发布

发布时间：2022-06-08 06:13:07

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [TriggerJobSavepoint](#)

第 17 次发布

发布时间：2022-05-23 06:11:33

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CheckSavepoint](#)
- [DescribeJobSavepoint](#)

新增数据结构：

- [Savepoint](#)

第 16 次发布

发布时间：2022-03-03 08:09:36

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)
 - 新增入参：WorkSpaceId
- [CreateJobConfig](#)
 - 新增入参：WorkSpaceId
- [CreateResource](#)
 - 新增入参：WorkSpaceId
- [CreateResourceConfig](#)
 - 新增入参：WorkSpaceId
- [DeleteResourceConfigs](#)
 - 新增入参：WorkSpaceId
- [DeleteResources](#)
 - 新增入参：WorkSpaceId
- [DeleteTableConfig](#)
 - 新增入参：WorkSpaceId
- [DescribeJobConfigs](#)
 - 新增入参：WorkSpaceId
- [DescribeJobs](#)
 - 新增入参：WorkSpaceId
- [DescribeResourceConfigs](#)
 - 新增入参：WorkSpaceId
- [DescribeResourceRelatedJobs](#)
 - 新增入参：WorkSpaceId
- [DescribeResources](#)
 - 新增入参：WorkSpaceId
- [RunJobs](#)
 - 新增入参：WorkSpaceId
- [StopJobs](#)
 - 新增入参：WorkSpaceId

修改数据结构：

- [JobV1](#)
 - 新增成员：WorkSpaceId, WorkspaceName

第 15 次发布

发布时间：2022-02-24 08:11:34

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：PythonVersion

修改数据结构：

- [JobConfig](#)
 - 新增成员：PythonVersion

第 14 次发布

发布时间：2022-01-25 08:15:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：LogCollectType

第 13 次发布

发布时间：2021-12-21 08:08:44

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RunJobDescription](#)
 - 新增成员：SavepointPath, SavepointId

第 12 次发布

发布时间：2021-12-07 08:09:58

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：ClsLogsetId, ClsTopicId

修改数据结构：

- [JobConfig](#)
 - 新增成员：ClsLogsetId, ClsTopicId

第 11 次发布

发布时间：2021-11-15 08:11:30

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateResource](#)
 - 新增入参：FolderId
 - 修改入参：Name
- [DescribeResourceRelatedJobs](#)
 - 新增入参：ResourceConfigVersion

第 10 次发布

发布时间：2021-09-24 10:00:20

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)
 - 新增入参：FlinkVersion
- [DescribeSystemResources](#)
 - 新增入参：FlinkVersion

第 9 次发布

发布时间：2021-09-06 08:01:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [JobV1](#)
 - 新增成员：FlinkVersion

第 8 次发布

发布时间：2021-08-12 08:01:27

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)
 - 新增入参：FolderId

第 7 次发布

发布时间：2021-06-24 08:01:29

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：JobManagerSpec, TaskManagerSpec

修改数据结构：

- [JobConfig](#)
 - 新增成员：JobManagerSpec, TaskManagerSpec
- [JobV1](#)
 - 新增成员：RunningCu

第 6 次发布

发布时间：2021-04-15 08:00:16

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [JobConfig](#)
 - 新增成员：MaxParallelism

第 5 次发布

发布时间：2021-04-14 08:00:14

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteResourceConfigs](#)
- [DeleteResources](#)
- [DescribeResourceConfigs](#)
- [DescribeResourceRelatedJobs](#)
- [DescribeResources](#)

新增数据结构：

- [ResourceConfigItem](#)
- [ResourceItem](#)
- [ResourceRefJobInfo](#)

第 4 次发布

发布时间：2021-02-05 14:54:40

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeJobConfigs](#)

新增数据结构：

- [JobConfig](#)
- [ResourceRefDetail](#)

第 3 次发布

发布时间：2021-02-03 08:00:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJobConfig](#)
 - 新增入参：LogCollect

第 2 次发布

发布时间：2021-01-25 08:00:06

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [JobV1](#)
 - 新增成员：ClusterStatus

第 1 次发布

发布时间：2021-01-14 14:22:54

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateJob](#)
- [CreateJobConfig](#)
- [CreateResource](#)
- [CreateResourceConfig](#)
- [DeleteTableConfig](#)
- [DescribeJobs](#)
- [DescribeSystemResources](#)
- [RunJobs](#)
- [StopJobs](#)

新增数据结构：

- [Filter](#)
- [JobV1](#)
- [Property](#)
- [ResourceLoc](#)
- [ResourceLocParam](#)
- [ResourceRef](#)
- [RunJobDescription](#)
- [StopJobDescription](#)
- [SystemResourceItem](#)

简介

最近更新时间：2025-04-25 01:48:30

流计算 Oceanus 是基于 Apache Flink 构建的位于云端的流式数据汇聚、计算服务，提供全托管的云上服务，您无需关注基础设施的运维，即可便捷对接丰富的云上数据源，获得完善的配套支持。

API 概览

最近更新时间：2024-10-14 01:26:16

— · — · — · — · — · —

日志相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeJobSubmissionLog	查询作业实例启动日志	20

资源相关接口

接口名称	接口功能	频率限制（次/秒）
CreateResource	创建资源接口	20
DeleteResources	删除资源接口	20
DescribeResourceRelatedJobs	获取资源关联作业信息	20
DescribeResources	描述资源接口	20
DescribeSystemResources	描述系统资源接口	20

资源配置相关接口

接口名称	接口功能	频率限制（次/秒）
CreateResourceConfig	创建资源配置接口	20
DeleteResourceConfigs	删除资源版本	20
DescribeResourceConfigs	描述资源配置接口	35

元数据相关接口

接口名称	接口功能	频率限制（次/秒）
GetMetaTable	查询元数据表	20

权限相关接口

接口名称	接口功能	频率限制（次/秒）
CreateWorkSpace	创建工作空间	20
DeleteWorkSpace	删除工作空间	20
ModifyWorkSpace	修改工作空间	20
DescribeWorkSpaces	授权工作空间列表	20

集群接口

接口名称	接口功能	频率限制（次/秒）
DescribeClusters	查询集群	20

其他接口

接口名称	接口功能	频率限制（次/秒）
CreateFolder	创建文件夹	20
DeleteFolders	删除文件夹	20
DeleteTableConfig	删除作业表配置	20

接口名称	接口功能	频率限制（次/秒）
DescribeTreeJobs	自定义树状结构	20
ModifyFolder	拖拽文件夹	20
DescribeTreeResources	查询树状结构资源列表	20
DescribeFolder	查询文件夹	20

作业接口

接口名称	接口功能	频率限制（次/秒）
CheckSavepoint	检查快照是否可用	20
CopyJobs	复制作业	20
DeleteJobs	删除作业	20
DescribeJobEvents	获取指定作业的事件	20
DescribeJobSavepoint	查找Savepoint列表	20
ModifyJob	更新作业	20
RunJobs	运行作业	20
StopJobs	停止作业	20
TriggerJobSavepoint	触发Savepoint	20
CreateJob	新建作业	20
DescribeJobs	查询作业	30
DescribeJobRuntimeInfo	获取作业运行时的信息	20
FetchSqlGatewayStatementResult	查询Statement执行结果	20
RunSqlGatewayStatement	执行Statement	20

作业配置接口

接口名称	接口功能	频率限制（次/秒）
CreateJobConfig	创建作业配置	20
DeleteJobConfigs	删除作业配置	20
DescribeJobConfigs	查询作业配置	20

△ 注意：

以上给出的接口频率限制维度为 API + 接入地域 + 子账号，有关限频更多说明参考：[API 频率限制说明](#)

调用方式

请求结构

最近更新時間：2025-05-21 01:26:34

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 `oceanus.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `oceanus.ap-guangzhou.tencentcloudapi.com`。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 `oceanus.ap-guangzhou.tencentcloudapi.com` 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的 **地域**。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	<code>oceanus.tencentcloudapi.com</code>
华南地区（广州）	<code>oceanus.ap-guangzhou.tencentcloudapi.com</code>
华东地区（上海）	<code>oceanus.ap-shanghai.tencentcloudapi.com</code>
华东地区（南京）	<code>oceanus.ap-nanjing.tencentcloudapi.com</code>
华北地区（北京）	<code>oceanus.ap-beijing.tencentcloudapi.com</code>
西南地区（成都）	<code>oceanus.ap-chengdu.tencentcloudapi.com</code>
西南地区（重庆）	<code>oceanus.ap-chongqing.tencentcloudapi.com</code>
港澳台地区（中国香港）	<code>oceanus.ap-hongkong.tencentcloudapi.com</code>
亚太东南（新加坡）	<code>oceanus.ap-singapore.tencentcloudapi.com</code>
亚太东南（雅加达）	<code>oceanus.ap-jakarta.tencentcloudapi.com</code>
亚太东南（曼谷）	<code>oceanus.ap-bangkok.tencentcloudapi.com</code>
亚太东北（首尔）	<code>oceanus.ap-seoul.tencentcloudapi.com</code>
亚太东北（东京）	<code>oceanus.ap-tokyo.tencentcloudapi.com</code>
美国东部（弗吉尼亚）	<code>oceanus.na-ashburn.tencentcloudapi.com</code>
美国西部（硅谷）	<code>oceanus.na-siliconvalley.tencentcloudapi.com</code>
南美地区（圣保罗）	<code>oceanus.sa-saopaulo.tencentcloudapi.com</code>
欧洲地区（法兰克福）	<code>oceanus.eu-frankfurt.tencentcloudapi.com</code>

注意：由于**金融区**和非金融区是隔离不通的，因此当访问金融区服务时（公共参数 `Region` 为金融区地域），需要同时指定带金融区地域的域名，最好和 `Region` 的地域保持一致。

金融区接入地域	金融区域名
华东地区（上海金融）	<code>oceanus.ap-shanghai-fsi.tencentcloudapi.com</code>
华南地区（深圳金融）	<code>oceanus.ap-shenzhen-fsi.tencentcloudapi.com</code>

2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型:

- application/json (推荐), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。
- application/x-www-form-urlencoded, 必须使用签名方法 v1 (HmacSHA1 或 HmacSHA256)。
- multipart/form-data (仅部分接口支持), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1 (HmacSHA1、HmacSHA256) 时不得超过1MB。POST 请求使用签名方法 v3 (TC3-HMAC-SHA256) 时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2025-01-09 01:23:32

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见 [签名方法 v3](#)。

注意：出于简化的目的，部分接口文档中的示例使用的是签名方法 v1 GET 请求，而不是更安全的签名方法 v3。

使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下表所示：

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 oceanus；tc3_request 为固定字符串； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST（application/json）请求结构示例：

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

HTTP POST（multipart/form-data）请求结构示例（仅特定的接口支持）：

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

使用签名方法 v1 的公共参数

使用签名方法 v1（有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见[文档](#)

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	—	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意： 某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。

参数名称	类型	必选	描述
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****

Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded

Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****
***
```

地域列表

本产品所有接口 Region 字段的可选值如下表所示。如果接口不支持该表中的所有地域，则会在接口文档中单独说明。

地域	取值
亚太东南（曼谷）	ap-bangkok
华北地区（北京）	ap-beijing
西南地区（成都）	ap-chengdu
西南地区（重庆）	ap-chongqing
华南地区（广州）	ap-guangzhou
港澳台地区（中国香港）	ap-hongkong
亚太东南（雅加达）	ap-jakarta
华东地区（南京）	ap-nanjing
亚太东北（首尔）	ap-seoul
华东地区（上海）	ap-shanghai
华东地区（上海金融）	ap-shanghai-fsi
华南地区（深圳金融）	ap-shenzhen-fsi
亚太东南（新加坡）	ap-singapore

地域	取值
亚太东北（东京）	ap-tokyo
欧洲地区（法兰克福）	eu-frankfurt
美国东部（弗吉尼亚）	na-ashburn
美国西部（硅谷）	na-siliconvalley
南美地区（圣保罗）	sa-saopaulo

签名方法 v3

最近更新时间：2024-12-25 01:51:32

以下文档说明了签名方法 v3 的签名过程，但仅在您编写自己的代码来调用腾讯云 API 时才有用。我们推荐您使用 [腾讯云 API Explorer](#)，[腾讯云 SDK](#) 和 [腾讯云命令行工具（TCCLI）](#) 等开发者工具，从而无需学习如何对 API 请求进行签名。

推荐使用 API Explorer

</> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共参数中指定该签名结果并以指定的方式和格式发送请求。

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 [云 API 密钥](#) 页面获取密钥相关信息。

2. 保护传输中的数据

为了防止请求在传输过程中被篡改，腾讯云 API 会使用请求参数来计算请求的哈希值，并将生成的哈希值加密后作为请求的一部分，发送到腾讯云 API 服务器。服务器会使用收到的请求参数以同样的过程计算哈希值，并验证请求中的哈希值。如果请求被篡改，将导致哈希值不一致，腾讯云 API 将拒绝本次请求。

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 JSON 格式，POST 请求支持传空数组和空字符串，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以对生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- 用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄露，请立刻禁用该安全凭证。

申请安全凭证的具体步骤如下：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云API密钥](#) 的控制台页面。
- 在 [云API密钥](#) 页面，单击【新建密钥】创建一对密钥。

签名版本 v3 签名过程

云 API 支持 GET 和 POST 请求。对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于 POST 方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式绝大多数接口均支持，multipart 格式只有特定接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

- 云服务器默认已开通，该接口很常用；
- 该接口是只读的，不会改变现有资源的状态；
- 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC-前缀）。

假设用户的 SecretId 和 SecretKey 分别是：AKID***** 和 *****。用户想查看广州区云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠 (/)。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中间号 (?) 后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接； 2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号 (;) 分隔。 此示例为 content-type;host;x-tc-action
HashedRequestPayload	请求正文（payload，即 body，此示例为 {"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编

字段名称	解释
	码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com
x-tc-action:describeinstances

content-type;host;x-tc-action
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + "\n" +
RequestTimestamp + "\n" +
CredentialScope + "\n" +
HashedCanonicalRequest
```

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能运行一段时间后，请求失败，返回签名过期错误。

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "*****"  
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)  
SecretService = HMAC_SHA256(SecretDate, Service)  
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：
da98fb70dcf6b112dc21038d1eeeb3a95c74b4dcb12c1131f864f6066bd02be0，
8d70cbefb03939f929db64d32dc2ba89b1095620119fe3e050e2b18c5bd2752f，
b596b923aad85185e2d1f6659d2a062e0a86731226e021e61bfe06f7ed05f5af。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，请以实际情况为准。此处的伪代码密钥参数 key 在前，消息参数 data 在后。通常标准库函数会提供二进制格式的返回值，也可能会提供打印友好的十六进制格式的返回值，此处使用的是二进制格式。

字段名称	解释
SecretKey	原始的 SecretKey，即 *****。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名，伪代码如下：

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =  
Algorithm + ' ' +  
'Credential=' + SecretId + '/' + CredentialScope + ', ' +  
'SignedHeaders=' + SignedHeaders + ', ' +  
'Signature=' + Signature
```

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKID*****。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host;x-tc-action。
Signature	签名值。此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}
```

⚠ 注意:

请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过打印实际请求内容，网络抓包等方式对比排查。

签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚地解释签名过程，下面以实际编程语言为例，将上述的签名过程完整实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
}
```

```
private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
private final static String CT_JSON = "application/json; charset=utf-8";

public static byte[] hmac256(byte[] key, String msg) throws Exception {
    Mac mac = Mac.getInstance("HmacSHA256");
    SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
    mac.init(secretKeySpec);
    return mac.doFinal(msg.getBytes(UTF8));
}

public static String sha256Hex(String s) throws Exception {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] d = md.digest(s.getBytes(UTF8));
    return DatatypeConverter.printHexBinary(d).toLowerCase();
}

public static void main(String[] args) throws Exception {
    String service = "cvm";
    String host = "cvm.tencentcloudapi.com";
    String region = "ap-guangzhou";
    String action = "DescribeInstances";
    String version = "2017-03-12";
    String algorithm = "TC3-HMAC-SHA256";
    String timestamp = "1551113065";
    //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    // 注意时区, 否则容易出错
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
    String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

    // ***** 步骤 1: 拼接规范请求串 *****
    String httpRequestMethod = "POST";
    String canonicalUri = "/";
    String canonicalQueryString = "";
    String canonicalHeaders = "content-type:application/json; charset=utf-8\n"
        + "host:" + host + "\n" + "x-tc-action:" + action.toLowerCase() + "\n";
    String signedHeaders = "content-type;host;x-tc-action";

    String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
    String hashedRequestPayload = sha256Hex(payload);
    String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
        + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
    System.out.println(canonicalRequest);

    // ***** 步骤 2: 拼接待签名字符串 *****
    String credentialScope = date + "/" + service + "/" + "tc3_request";
    String hashedCanonicalRequest = sha256Hex(canonicalRequest);
    String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
    System.out.println(stringToSign);

    // ***** 步骤 3: 计算签名 *****
    byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
    byte[] secretService = hmac256(secretDate, service);
    byte[] secretSigning = hmac256(secretService, "tc3_request");
    String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
    System.out.println(signature);
}
```

```
// ***** 步骤 4: 拼接 Authorization *****

String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;

System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\")")
.append(" -H \"X-TC-Action: ").append(action).append("\")")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\")")
.append(" -H \"X-TC-Version: ").append(version).append("\")")
.append(" -H \"X-TC-Region: ").append(region).append("\")")
.append(" -d ").append(payload).append("");
System.out.println(sb.toString());
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}

# ***** 步骤 1: 拼接规范请求串 *****

http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
```

```

ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

# ***** 步骤 3: 计算签名 *****
# 计算签名摘要函数
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** 步骤 4: 拼接 Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"
+ ' -H "Host: ' + host + '"
+ ' -H "X-TC-Action: ' + action + '"
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"
+ ' -H "X-TC-Version: ' + version + '"
+ ' -H "X-TC-Region: ' + region + '"
+ " -d '" + payload + "'")

```

Golang

```

package main

import (
"crypto/hmac"

```

```
"crypto/sha256"
"encoding/hex"
"fmt"
"os"
"strings"
"time"
)

func sha256hex(s string) string {
    b := sha256.Sum256([]byte(s))
    return hex.EncodeToString(b[:])
}

func hmacsha256(s, key string) string {
    hashed := hmac.New(sha256.New, []byte(key))
    hashed.Write([]byte(s))
    return string(hashed.Sum(nil))
}

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
    canonicalQueryString := ""
    canonicalHeaders := fmt.Sprintf("content-type:%s\nhost:%s\nx-tc-action:%s\n",
        "application/json; charset=utf-8", host, strings.ToLower(action))
    signedHeaders := "content-type;host;x-tc-action"
    payload := `{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}`
    hashedRequestPayload := sha256hex(payload)
    canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
        httpRequestMethod,
        canonicalURI,
        canonicalQueryString,
        canonicalHeaders,
        signedHeaders,
        hashedRequestPayload)
    fmt.Println(canonicalRequest)

    // step 2: build string to sign
    date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
    credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
    hashedCanonicalRequest := sha256hex(canonicalRequest)
    string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
        algorithm,
```

```
timestamp,
credentialScope,
hashedCanonicalRequest)
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = implode("\n", [
"content-type:application/json; charset=utf-8",
"host:".$host,
"x-tc-action:".strtolower($action),
```

```
""
});
$signedHeaders = implode(";", [
    "content-type",
    "host",
    "x-tc-action",
]);
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]'}';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/". $service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=". $secretId."/". $credentialScope
.", SignedHeaders=". $signedHeaders.", Signature=". $signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"
.' -H "Content-Type: application/json; charset=utf-8"
.' -H "Host: '.$host.'"
.' -H "X-TC-Action: '.$action.'"
.' -H "X-TC-Timestamp: '.$timestamp.'"
.' -H "X-TC-Version: '.$version.'"
.' -H "X-TC-Region: '.$region.'"
.' -d "'.$payload.'"";
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
```

```
require 'json'
require 'time'
require 'openssl'

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\nxc-action:#{action.downcase}\n"
signed_headers = 'content-type;host;x-tc-action'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['未命名'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
  http_request_method,
  canonical_uri,
  canonical_querystring,
  canonical_headers,
  signed_headers,
  hashed_request_payload,
].join("\n")

puts canonical_request

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
  algorithm,
  timestamp.to_s,
  credential_scope,
  hashed_request_payload,
].join("\n")
puts string_to_sign

# ***** 步骤 3: 计算签名 *****
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
```

```
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ***** 步骤 4: 拼接 Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signature}"
puts authorization

puts 'curl -X POST ' + endpoint \
+ ' -H "Authorization: ' + authorization + '"' \
+ ' -H "Content-Type: application/json; charset=utf-8"' \
+ ' -H "Host: ' + host + '"' \
+ ' -H "X-TC-Action: ' + action + '"' \
+ ' -H "X-TC-Timestamp: ' + timestamp.to_s + '"' \
+ ' -H "X-TC-Version: ' + version + '"' \
+ ' -H "X-TC-Region: ' + region + '"' \
+ " -d '" + payload + '"'
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

public class Application
{
    public static string SHA256Hex(string s)
    {
        using (SHA256 algo = SHA256.Create())
        {
            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
            StringBuilder builder = new StringBuilder();
            for (int i = 0; i < hashbytes.Length; ++i)
            {
                builder.Append(hashbytes[i].ToString("x2"));
            }
            return builder.ToString();
        }
    }

    public static byte[] HmacSHA256(byte[] key, byte[] msg)
    {
        using (HMACSHA256 mac = new HMACSHA256(key))
        {
            return mac.ComputeHash(msg);
        }
    }

    public static Dictionary<String, String> BuildHeaders(string secretid,
        string secretkey, string service, string endpoint, string region,
        string action, string version, DateTime date, string requestPayload)
    {
        string datestr = date.ToString("yyyy-MM-dd");
```

```
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
// ***** 步骤 1: 拼接规范请求串 *****

string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

// ***** 步骤 2: 拼接待签名字符串 *****

string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n"
+ requestTimestamp.ToString() + "\n"
+ credentialScope + "\n"
+ hashedCanonicalRequest;
Console.WriteLine(stringToSign);

// ***** 步骤 3: 计算签名 *****

byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);

// ***** 步骤 4: 拼接 Authorization *****

string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}

public static void Main(string[] args)
```

```
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

string service = "cvm";
string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";

// 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
// DateTime date = DateTime.UtcNow;
// 注意时区, 建议此时间统一采用UTC时间戳, 则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}";

Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service
, endpoint, region, action, version, date, requestPayload);

Console.WriteLine("POST https://cvm.tencentcloudapi.com");
foreach (KeyValuePair<string, string> kv in headers)
{
Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine();
Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
const hmac = crypto.createHmac('sha256', secret)
return hmac.update(message).digest(encoding)
}

function getHash(message, encoding = 'hex') {
const hash = crypto.createHash('sha256')
return hash.update(message).digest(encoding)
}

function getDate(timestamp) {
const date = new Date(timestamp * 1000)
const year = date.getUTCFullYear()
const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
const day = ('0' + date.getUTCDate()).slice(-2)
return `${year}-${month}-${day}`
}

function main(){
```

```
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
//时间处理, 获取世界时间日期
const date = getDate(timestamp)

// ***** 步骤 1: 拼接规范请求串 *****
const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"

const hashedRequestPayload = getHash(payload);
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8\\n"
+ "host:" + endpoint + "\\n"
+ "x-tc-action:" + action.toLowerCase() + "\\n"
const signedHeaders = "content-type;host;x-tc-action"

const canonicalRequest = httpRequestMethod + "\\n"
+ canonicalUri + "\\n"
+ canonicalQueryString + "\\n"
+ canonicalHeaders + "\\n"
+ signedHeaders + "\\n"
+ hashedRequestPayload
console.log(canonicalRequest)

// ***** 步骤 2: 拼接待签名字符串 *****
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\\n" +
timestamp + "\\n" +
credentialScope + "\\n" +
hashedCanonicalRequest
console.log(stringToSign)

// ***** 步骤 3: 计算签名 *****
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)

// ***** 步骤 4: 拼接 Authorization *****
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
```

```

"Signature=" + signature
console.log(authorization)

const curlcmd = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + '"'
console.log(curlcmd)
}
main()

```

C++

```

#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
    utcDate = string(buff);
    return utcDate;
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);

```

```
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
    snprintf(buf, sizeof(buf), "%02x", hash[i]);
    NewString = NewString + buf;
}
return NewString;
}

string HmacSha256(const string &key, const string &input)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
    #else
    h = HMAC_CTX_new();
    #endif

    HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
    HMAC_Update(h, ( unsigned char* )&input[0], input.length());
    unsigned int len = 32;
    HMAC_Final(h, hash, &len);

    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
    #else
    HMAC_CTX_free(h);
    #endif

    std::stringstream ss;
    ss << std::setfill('0');
    for (int i = 0; i < len; i++)
    {
        ss << hash[i];
    }

    return (ss.str());
}

string HexEncode(const string &input)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = input.length();

    string output;
    output.reserve(2 * len);
    for (size_t i = 0; i < len; ++i)
    {
        const unsigned char c = input[i];
        output.push_back(lut[c >> 4]);
        output.push_back(lut[c & 15]);
    }
}
```

```
}
return output;
}

int main()
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

string service = "cvm";
string host = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
int64_t timestamp = 1551113065;
string date = get_data(timestamp);

// ***** 步骤 1: 拼接规范请求串 *****
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string lower = action;
std::transform(action.begin(), action.end(), lower.begin(), ::tolower);
string canonicalHeaders = string("content-type:application/json; charset=utf-8\n")
+ "host:" + host + "\n"
+ "x-tc-action:" + lower + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
cout << canonicalRequest << endl;

// ***** 步骤 2: 拼接待签名字符串 *****
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;

// ***** 步骤 3: 计算签名 *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;

// ***** 步骤 4: 拼接 Authorization *****
```

```
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;

string curlcmd = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: \" + authorization + "\"\n"
+ " -H \"Content-Type: application/json; charset=utf-8\" + "\n"
+ " -H \"Host: \" + host + "\"\n"
+ " -H \"X-TC-Action: \" + action + "\"\n"
+ " -H \"X-TC-Timestamp: \" + RequestTimestamp + "\"\n"
+ " -H \"X-TC-Version: \" + version + "\"\n"
+ " -H \"X-TC-Region: \" + region + "\"\n"
+ " -d '\" + payload + '\"';
cout << curlcmd << endl;
return 0;
};
```

C

```
#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdint.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

void get_utc_date(int64_t timestamp, char* utc, int len)
{
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(utc, len, "%Y-%m-%d", &sttime);
}

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        strcat(result, buf);
    }
}

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
unsigned char* output, unsigned int* output_len)
{
    HMAC_CTX *h;
```

```
#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, key, key_len, EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )input, input_len);
HMAC_Final(h, output, output_len);

#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif

}

void hex_encode(const char* input, int input_len, char* output)
{
    static const char* const lut = "0123456789abcdef";

    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < input_len; ++i)
    {
        const unsigned char c = input[i];
        temp[0] = lut[c >> 4];
        strcat(add_out, temp);
        temp[0] = lut[c & 15];
        strcat(add_out, temp);
    }
    strncpy(output, add_out, 128);
}

void lowercase(const char * src, char * dst)
{
    for (int i = 0; src[i]; i++)
    {
        dst[i] = tolower(src[i]);
    }
}

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
    const char* service = "cvm";
    const char* host = "cvm.tencentcloudapi.com";
    const char* region = "ap-guangzhou";
    const char* action = "DescribeInstances";
```

```
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

// ***** 步骤 1: 拼接规范请求串 *****
const char* http_request_method = "POST";
const char* canonical_uri = "/";
const char* canonical_query_string = "";
char canonical_headers[100] = {"Content-type:application/json; charset=utf-8\nhost:"};
strcat(canonical_headers, host);
strcat(canonical_headers, "\nx-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]\"}";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

char canonical_request[256] = {0};
sprintf(canonical_request, "%s\n%s\n%s\n%s\n%s\n%s", http_request_method,
canonical_uri, canonical_query_string, canonical_headers,
signed_headers, hashed_request_payload);
printf("%s\n", canonical_request);

// ***** 步骤 2: 拼接待签名字符串 *****
const char* algorithm = "TC3-HMAC-SHA256";
char request_timestamp[16] = {0};
sprintf(request_timestamp, "%d", timestamp);
char credential_scope[64] = {0};
strcat(credential_scope, date);
sprintf(credential_scope, "%s/%s/tc3_request", date, service);
char hashed_canonical_request[100] = {0};
sha256_hex(canonical_request, hashed_canonical_request);
char string_to_sign[256] = {0};
sprintf(string_to_sign, "%s\n%s\n%s\n%s", algorithm, request_timestamp,
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s", "TC3", SECRET_KEY);
unsigned char k_date[64] = {0};
unsigned int output_len = 0;
hmac_sha256(k_key, strlen(k_key), date, strlen(date), k_date, &output_len);
unsigned char k_service[64] = {0};
hmac_sha256(k_date, output_len, service, strlen(service), k_service, &output_len);
unsigned char k_signing[64] = {0};
hmac_sha256(k_service, output_len, "tc3_request", strlen("tc3_request"), k_signing, &output_len);
unsigned char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, output_len, string_to_sign, strlen(string_to_sign), k_hmac_sha_sign, &output_len);

char signature[128] = {0};
```

```
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

// ***** 步骤 4: 拼接 Authorization *****
char authorization[512] = {0};
sprintf(authorization, "%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm, SECRET_ID, credential_scope, signed_headers, signature);
printf("%s\n", authorization);

char curlcmd[10240] = {0};
sprintf(curlcmd, "curl -X POST https://%s\n \
-H \"Authorization: %s\"\n \
-H \"Content-Type: application/json; charset=utf-8\"\n \
-H \"Host: %s\"\n \
-H \"X-TC-Action: %s\"\n \
-H \"X-TC-Timestamp: %s\"\n \
-H \"X-TC-Version: %s\"\n \
-H \"X-TC-Region: %s\"\n \
-d '%s'",
host, authorization, host, action, request_timestamp, version, region, payload);
printf("%s\n", curlcmd);
return 0;
}
```

其他语言

- Lua: [GitHub](#)
- Swift: [GitHub](#)
- Dart: [GitHub](#)
- Shell(Bash): [GitHub](#)

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不相符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2024-12-25 01:51:32

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

推荐使用 API Explorer

</> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。

1. 申请安全凭证

在第一次使用云 API 之前，请前往 [云 API 密钥页面](#) 申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云 API 密钥](#) 的控制台页面
- 在 [云 API 密钥](#) 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。以下是使用签名方法 v1 生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKID*****
- SecretKey: *****

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表（DescribeInstances）请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKID*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数。

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action' : 'DescribeInstances',
  'InstanceIds.0' : 'ins-09dx96dg',
  'Limit' : 20,
  'Nonce' : 11886,
  'Offset' : 0,
  'Region' : 'ap-guangzhou',
  'SecretId' : 'AKID*****',
  'Timestamp' : 1465185768,
  'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 "DescribeInstances"，因此格式化后即为 Action=DescribeInstances。

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****
&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为：cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。
4. 请求字符串: 即上一步生成的请求字符串。

签名原串的拼接规则为：请求方法 + 请求主机 + 请求路径 + ? + 请求字符串。

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-gu
angzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。

首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例：

```
$secretKey = '*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&R
egion=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12';
```

```
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ=，最终得到的签名串请求参数（Signature）为：

7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的网络库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 [RFC 3986](#)。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-`

guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12。

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
        // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
        StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
        // 实际请求的url中对参数顺序没有要求
        for (String k : params.keySet()) {
            // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
        // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：params.put("Timestamp", System.currentTimeMillis() / 1000);
        params.put("Timestamp", 1465185768); // 公共参数
        // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID*****
        params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
    }
}
```

```
params.put("Action", "DescribeInstances"); // 公共参数
params.put("Version", "2017-03-12"); // 公共参数
params.put("Region", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA1")); // 公共参数
System.out.println(getUrl(params));
}
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： `pip install requests` 。

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import os
import time

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
```

```
# resp = requests.get("https://" + endpoint, params=data)
# print(resp.url)
```

Golang

```
package main

import (
    "bytes"
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "fmt"
    "os"
    "sort"
    "strconv"
)

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": strconv.Itoa(1465185768),
        "Region": "ap-guangzhou",
        "SecretId": secretId,
        "Version": "2017-03-12",
        "Action": "DescribeInstances",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": strconv.Itoa(20),
        "Offset": strconv.Itoa(0),
    }

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("cvm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

    // sort keys by ascii asc order
    keys := make([]string, 0, len(params))
    for k, _ := range params {
        keys = append(keys, k)
    }
    sort.Strings(keys)

    for i := range keys {
        k := keys[i]
        buf.WriteString(k)
        buf.WriteString("=")
        buf.WriteString(params[k])
        buf.WriteString("&")
    }
    buf.Truncate(buf.Len() - 1)
```

```
hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $params as $key => $value ) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
```

```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
  'SecretId' => secret_id,
  'Timestamp' => 1465185768, # Time.now.to_i
  'Version' => '2017-03-12',
}
sign = method + endpoint + '/?'
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;

public class Application {
    public static string Sign(string signKey, string secret)
    {
        string signRet = string.Empty;
        using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
        {
            byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
            signRet = Convert.ToBase64String(hash);
        }
        return signRet;
    }

    public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
    {
        string retStr = "";
        retStr += requestMethod;
```

```
retStr += requestHost;
retStr += requestPath;
retStr += "?";
string v = "";
foreach (string key in requestParams.Keys)
{
    v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

    string endpoint = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";
    double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25, 此参数作为示例, 以实际为准
    // long timestamp = ToTimestamp() / 1000;
    // string requestTimestamp = timestamp.ToString();
    Dictionary<string, string> param = new Dictionary<string, string>();
    param.Add("Limit", "20");
    param.Add("Offset", "0");
    param.Add("InstanceIds.0", "ins-09dx96dg");
    param.Add("Action", action);
    param.Add("Nonce", "11886");
    // param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

    param.Add("Timestamp", RequestTimestamp.ToString());
    param.Add("Version", version);

    param.Add("SecretId", SECRET_ID);
    param.Add("Region", region);
    SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
    string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
    string sigOutParam = Sign(SECRET_KEY, sigInParam);
    Console.WriteLine(sigOutParam);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
    params['Signature'] = encodeURIComponent(params['Signature']);
    const url_strParam = sort_params(params)
    return "https://" + endpoint + "?" + url_strParam.slice(1);
}
```

```
function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}

function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
let strParam = "";
let keys = Object.keys(params);
keys.sort();
for (let k in keys) {
//k = k.replace(/_/g, '.');
strParam += ("%&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}

function main(){
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768 // 时间戳 2016-06-06 12:02:48, 此参数作为示例, 以实际为准
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886 // 随机正整数
//const nonce = Math.round(Math.random() * 65535)

let params = {};
params['Action'] = Action;
params['InstanceIds.0'] = 'ins-09dx96dg';
params['Limit'] = 20;
params['Offset'] = 0;
params['Nonce'] = Nonce;
params['Region'] = Region;
params['SecretId'] = SECRET_ID;
params['Timestamp'] = Timestamp;
params['Version'] = Version;

// 1. 对参数排序, 并拼接请求字符串
strParam = sort_params(params)

// 2. 拼接签名原字符串
const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)
```

```
// 3. 生成签名串
params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])

// 4. 进行url编码并拼接请求url
// const req_url = get_req_url(params, endpoint)
// console.log(params['Signature'])
// console.log(req_url)
}
main()
```

返回结果

最近更新时间：2024-03-12 01:42:39

云 API 3.0 接口默认返回 JSON 数据，返回非 JSON 格式的接口会在文档中做出说明。返回 JSON 数据时最大限制为 50 MB，如果返回的数据超过最大限制，请求会失败并返回内部错误。请根据接口文档中给出的过滤功能（例如时间范围）或者分页功能，控制返回数据不要过大。

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为 200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是 200，而不是 401。

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系 [腾讯云客服](#) 或 [提交工单](#)，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系 [腾讯云客服](#) 或 [提交工单](#)，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码。完整的错误码列表请参考本产品“API 文档”目录下的“错误码”页面。

参数类型

最近更新时间：2022-08-10 06:42:31

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

- String: 字符串。
- Integer: 整型，上限为无符号64位整数。SDK 3.0 不同编程语言支持的类型有所差异，建议以所使用编程语言的最大整型定义，例如 Golang 的 `uint64`。
- Boolean: 布尔型。
- Float: 浮点型。
- Double: 双精度浮点型。
- Date: 字符串，日期格式。例如：2022-01-01。
- Timestamp: 字符串，时间格式。例如：2022-01-01 00:00:00。
- Timestamp ISO8601: ISO 8601 是由国际标准化组织（International Organization for Standardization，ISO）发布的关于日期和时间格式的国际标准，对应国标《GB/T 7408-2005数据元和交换格式信息交换日期和时间表示法》。建议以所使用编程语言的标准库进行格式解析。例如：2022-01-01T00:00:00+08:00。
- Binary: 二进制内容，需要以特定协议请求和解析。

日志相关接口

查询作业实例启动日志

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询作业实例启动日志

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeJobSubmissionLog。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID，例如：cql-6v1jkxrn 示例值：cql-ebgioapf
StartTime	是	Integer	起始时间，unix时间戳，毫秒级，例如：1611754219108 示例值：1585749042
EndTime	是	Integer	结束时间，unix时间戳，毫秒级，例如：1611754219108 示例值：1585749042
RunningOrderId	否	Integer	作业运行的实例ID，例如：1,2,3。默认为0，表示未选中任何实例，搜索该时间段内最近的一个实例的日志 示例值：0
Keyword	否	String	日志搜索的关键词，默认为空
Cursor	否	String	日志搜索的游标，可透传上次返回的值，默认为空
OrderType	否	String	时间戳排序规则，asc – 升序，desc – 降序。默认为升序 示例值：asc
Limit	否	Integer	搜索的日志条数上限值，最大为100 示例值：100

3. 输出参数

参数名称	类型	描述
Cursor	String	日志搜索的游标，需要搜索更多时透传这个值 示例值： Y29udGV4dC1hNTI3ZmVhYy01YmE0LTQ5MWMtYjAwZS1kZGQwZWUwZWU2NDAxI
ListOver	Boolean	是否返回了所有的日志记录 示例值：false
JobRequestId	String	作业启动的requestId 注意：此字段可能返回 null，表示取不到有效值。 示例值：dfd73175-f8d4-4981-8cd3-8072e0a52fbd

参数名称	类型	描述
JobInstanceList	Array of JobInstanceForSubmissionLog	该时间段内符合关键字的所有的作业实例列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
LogList	Array of String	废弃，请使用LogContentList 注意：此字段可能返回 null，表示取不到有效值。 示例值：["2020-04-08 11:57:32.703 [queuedThreadPool-57106
LogContentList	Array of LogContent	日志列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会带 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询启动日志

查询启动日志

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobSubmissionLog
<公共请求参数>

{
  "JobId": "cq1-067opv71",
  "RunningOrderId": 1,
  "StartTime": 1698044059122,
  "EndTime": 1698045059131
}
```

输出示例

```
{
  "Response": {
    "Cursor": "",
    "JobInstanceList": null,
    "JobRequestId": "9acab4fc-d101-47ca-96a6-da08fe3fc130",
    "ListOver": true,
    "LogContentList": [
      {
        "ContainerName": "",
        "Log": "2023-10-23 14:54:18.158 [pool-1-thread-14] INFO org.apache.flink.kubernetes.utils.KubernetesUtils - Kubernetes deployment requires a fixed port. Configuration blob.server.port will be set to 6124",
        "PkgId": "",
        "PkgLogId": 0,
        "Time": 1698044059131
      },
      {
        "ContainerName": "",
        "Log": "2023-10-23 14:54:18.158 [pool-1-thread-14] INFO org.apache.flink.kubernetes.utils.KubernetesUtils - Kubernetes deployment requires a fixed port. Configuration taskmanager.rpc.port will be set to 6122",
        "PkgId": "",

```

```
"PkgLogId": 0,
"Time": 1698044059131
},
{
  "ContainerName": "",
  "Log": "2023-10-23 14:54:18.590 [pool-1-thread-14] INFO com.qcloud.oceanus.ca.service.RunJobService - Finish invoking clusterClient.run (run job)",
  "PkgId": "",
  "PkgLogId": 0,
  "Time": 1698044059132
}
],
"LogList": [
  "2020-04-08 11:58:35.048 [queuedThreadPool-57106 dfd73175-f8d4-4981-8cd3-8072e0a52fbd cql-ebgioapf 4 20200408115835048] WARN org.apache.flink.streaming.connectors.kafka.FlinkKafkaProducerBase - Overwriting the key.serializer is not recommended"
],
"RequestId": "9270beec-295b-42dd-836b-11ec47b6bff7"
}
}
```

示例2 查询指定job实例的启动日志

持续加载数据

输入示例

```
https://oceanus.tencentcloudapi.com/?Action=DescribeJobSubmissionLog
&JobId=cql-ebgioapf
&RunningOrderId=1
&StartTime=1586284444
&EndTime=1586327659
&Cursor=Y29udGV4dC0xMzcyM2NmZi03YjAyLTQ2MTgtYWJiOC0zYWQ5ZmU3MzIxMGUxNTg2MzI3OTk0NjEx
&Limit=100
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Cursor": "Y29udGV4dC1hNTI3ZmVhYy01YmE0LTQ5MWMtYjAwZS1kZGQwZWU2NDAxNTg2MzI4MTQ4Nzkw",
    "JobInstanceList": [],
    "JobRequestId": "dfd73175-f8d4-4981-8cd3-8072e0a52fbd",
    "ListOver": false,
    "LogList": [
      "2020-04-08 11:57:32.703 [queuedThreadPool-57106 dfd73175-f8d4-4981-8cd3-8072e0a52fbd cql-ebgioapf 1 20200408115732703] INFO com.tencent.cloud.tstream.flink.toolbox.sql.parse.settings.SettingManager - Registered Setting Handler CHECKPOINT_INTERVAL to CheckpointIntervalHandler for POST_PROCESSING"
    ],
    "LogContentList": [
      {
        "Log": "2023-09-11 15:37:23.094 [jobmanager-future-thread-4] INFO org.apache.flink.runtime.checkpoint.CheckpointCoordinator - Completed checkpoint 2767 for job 00000000000000000000000000000000 (0 bytes in 269 ms).",
        "Time": 1694417844009,
        "PkgId": "11CBB7C1649A91DD-1741",
        "PkgLogId": 131073,
        "ContainerName": "cql-07zmx8tv-318940-64f4dbfcb6-vtf67"
      }
    ]
  }
}
```

```

},
"RequestId": "db79c5af-adba-484b-8cf0-775e0b0b8070"
}
}

```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InternalError.LogicError	内部错误。
InvalidParameter.IllegalSearchKeyword	非法的搜索字符串。
InvalidParameterValue.InvalidLimit	无效限制错误。
InvalidParameterValue.InvalidTime	无效时间错误。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.LogTopic	找不到日志。
UnsupportedOperation.ClsSqlNotEnabled	Cls Sql未启用。

资源相关接口

创建资源接口

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：`oceanus.tencentcloudapi.com`。

创建资源接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateResource。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceLoc	是	ResourceLoc	资源位置 示例值：[]
ResourceType	是	Integer	资源类型。目前只支持 JAR，取值为 1 示例值：1
Remark	否	String	资源描述 示例值：v1
Name	否	String	资源名称 示例值：1
ResourceConfigRemark	否	String	资源版本描述 示例值：1
FolderId	否	String	目录ID 示例值：folder-xxxx
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1237

3. 输出参数

参数名称	类型	描述
ResourceId	String	资源ID 示例值：resource-xxx
Version	Integer	资源版本 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建依赖

创建依赖

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateResource
<公共请求参数>

{
  "ResourceLoc": {
    "StorageType": 1,
    "Param": {
      "Bucket": "oceanus-online-resource01-gz-1257058918",
      "Path": "1257058945/100006386216/upload/20231120142441/flink-hello-world-1.0.0-jar-with-dependencies.jar",
      "Region": "ap-guangzhou"
    }
  },
  "Remark": "test",
  "Name": "flink-hello-world-1.0.0-jar-with-dependencies.jar",
  "ResourceType": 1,
  "WorkSpaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1a7f54d8-fbfa-4367-a572-219c46b18058",
    "ResourceId": "resource-gk2iv5z5",
    "Version": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.COSClient	COS 服务访问错误。
InternalError.DB	数据库异常。
InvalidParameter.InvalidAppId	appid错误。
LimitExceeded	超过配额限制。
ResourceInUse.ResourceNameAlreadyExists	资源名称已存在。
ResourceNotFound	资源不存在。
ResourceUnavailable.CheckResourceLocExists	检查资源位置是否存在错误。
ResourceUnavailable.ResourceLocNotExists	cos上的程序包不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

删除资源接口

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

删除资源接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteResources。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceIds.N	是	Array of String	待删除资源ID列表 示例值：resource-xxx
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除依赖

删除依赖

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteResources
<公共请求参数>

{
  "ResourceIds": [
    "resource-gk2iv5z5"
  ],
  "WorkspaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1218133b-2a74-413d-abef-237bee5803df"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InternalError.LogicError	内部错误。
InvalidParameter.InvalidResourceIds	ResourceIds非法。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

获取资源关联作业信息

最近更新時間：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

获取资源关联作业信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeResourceRelatedJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceId	是	String	资源ID 示例值：resource-xxx
DESCByJobConfigCreateTime	否	Integer	默认0；1：按照作业版本创建时间降序 示例值：0
Offset	否	Integer	偏移量，默认为0 示例值：0
Limit	否	Integer	分页大小，默认为20，最大值为100 示例值：20
ResourceConfigVersion	否	Integer	资源版本号 示例值：1
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总数 示例值：0
RefJobInfos	Array of ResourceRefJobInfo	关联作业信息 示例值：[]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询依赖关联作业

查询特定关联作业

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeResourceRelatedJobs
<公共请求参数>

{
  "ResourceId": "resource-ew6aw29s",
  "Offset": 0,
  "Limit": 100,
  "ResourceConfigVersion": 1,
  "WorkSpaceId": "space-1257057943ap-guangzhou"
}
```

输出示例

```
{
  "Response": {
    "RefJobInfos": [
      {
        "JobConfigVersion": -1,
        "JobId": "cq1-oy8hvg7i",
        "ResourceVersion": 1
      }
    ],
    "RequestId": "5c2f633d-1706-492b-811f-4cdd6ec24810",
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

描述资源接口

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

描述资源接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeResources。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceIds.N	否	Array of String	需要查询的资源ID数组，数量不超过100个。如果填写了该参数则忽略Filters参数。 示例值：["resource-8udfak5k","resource-8udfak5j"]
Offset	否	Integer	偏移量，仅当设置 Limit 参数时有效 示例值：0
Limit	否	Integer	条数限制。如果不填，默认返回 20 条 示例值：20
Filters.N	否	Array of Filter	ResourceName 按照资源名字过滤，支持模糊过滤。传入的过滤名字不超过5个 类型: String 必选: 否 示例值：[]
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx
SystemResource	否	Integer	资源类型，0=用户，1系统connector，2=用户自定义connector 示例值：1

3. 输出参数

参数名称	类型	描述
ResourceSet	Array of ResourceItem	资源详细信息集合
TotalCount	Integer	总数量 示例值：[]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询资源

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeResources
<公共请求参数>

{
  "ResourceIds": [
    "resource-greuon9y"
  ],
  "Offset": 0,
  "Limit": 0,
  "WorkSpaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2f1a8e8b-a5e7-41a1-a96c-f40091d6ffff",
    "ResourceSet": [
      {
        "AppId": 1257058945,
        "CreateTime": "2023-08-16 20:29:58",
        "CreatorUin": "100032489861",
        "FileName": "flink-hello-world-1.0.0.jar",
        "LatestResourceConfigVersion": 1,
        "Name": "flink-hello-world-1.0.0.jar",
        "OwnerUin": "100006386216",
        "RefJobCount": 2,
        "IsJobRun": 0,
        "RefJobStatusCountSet": [
          {
            "Count": 1,
            "JobStatus": 5
          },
          {
            "Count": 3,
            "JobStatus": 4
          }
        ],
        "Region": "ap-guangzhou",
        "Remark": "test",
        "ResourceId": "resource-greuon9y",
        "ResourceLoc": {
          "Param": {
            "Bucket": "oceanus-online-resource01-gz-1257058918",
            "Path": "1257058945/100006386216/upload/20230816202945/flink-hello-world-1.0.0.jar",
            "Region": "ap-guangzhou"
          },
          "StorageType": 1
        }
      }
    ]
  }
}
```

```
},
"ResourceType": 1,
"UpdateTime": "2023-08-16 20:29:58",
"VersionCount": 1,
"WorkSpaceId": 2892
}
},
"TotalCount": 1
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InternalError.LogicError	内部错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceNotFound.ResourceConfig	程序包版本不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

描述系统资源接口

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

描述系统资源接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSystemResources。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceIds.N	否	Array of String	需要查询的资源ID数组 示例值：[]
Offset	否	Integer	偏移量，仅当设置 Limit 参数时有效 示例值：0
Limit	否	Integer	条数限制，默认返回 20 条 示例值：20
Filters.N	否	Array of Filter	查询资源配置列表，如果不填写，返回该 ResourceIds.N 下所有作业配置列表 示例值：[]
ClusterId	否	String	集群ID 示例值：cluster-xxxx
FlinkVersion	否	String	查询对应Flink版本的内置connector 示例值：Flink-1.11
WorkSpaceId	否	String	空间 示例值：space-123qew

3. 输出参数

参数名称	类型	描述
ResourceSet	Array of SystemResourceItem	资源详细信息集合 示例值：[]
TotalCount	Integer	总数量 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询系统资源配置列表示例

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeSystemResources
<公共请求参数>

{
  "ClusterId": "cluster-n8yaia0p"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "09250e9b-d3a1-4d37-b44f-de10e06ed52d",
    "ResourceSet": [
      {
        "ResourceId": "resource-abd503yt",
        "Name": "testjar",
        "ResourceType": 1,
        "Remark": "",
        "Region": "ap-guangzhou",
        "LatestResourceConfigVersion": 1
      }
    ],
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
InternalError	内部错误。
InternalError.DB	数据库异常。
InternalError.FailedToDescribeResources	失败的目标资源。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceNotFound.ClusterId	集群不存在。
ResourceUnavailable.FailedToDescribeResources	失败的目标资源。

资源配置相关接口

创建资源配置接口

最近更新时间：2025-04-25 01:48:24

1. 接口描述

接口请求域名：`oceanus.tencentcloudapi.com`。

创建资源配置接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateResourceConfig。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceId	是	String	资源ID 示例值：resource-xxx
ResourceLoc	是	ResourceLoc	位置信息 示例值：[]
Remark	否	String	资源描述信息 示例值：xxx
AutoDelete	否	Integer	1：资源版本达到上限，自动删除最早可删除的版本 示例值：0
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
Version	Integer	资源版本ID 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建依赖版本

创建依赖版本

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: CreateResourceConfig
<公共请求参数>

{
  "ResourceId": "resource-qreuon9y",
  "ResourceLoc": {
    "StorageType": 1,
    "Param": {
      "Bucket": "oceanus-online-resource01-gz-1257058918",
      "Path": "1257058945/100006386216/upload/20231120150319/flink-hello-world-1.0.0.jar",
      "Region": "ap-guangzhou"
    }
  },
  "Remark": "test",
  "WorkspaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "65812cf1-22bb-4ecd-89bb-01fe5a12ffb2",
    "Version": 3
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。

错误码	描述
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.ResourceNotExist	资源不存在。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceNotFound.ResourceNotExist	资源不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

删除资源版本

最近更新时间：2025-04-25 01:48:24

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

删除资源版本

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteResourceConfigs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceId	是	String	资源ID 示例值：resource-xxx
ResourceConfigVersions.N	是	Array of Integer	资源版本数组 示例值：1
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除资源版本

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteResourceConfigs
<公共请求参数>

{
  "ResourceId": "resource-gk2iv5z5",
  "ResourceConfigVersions": [
    "1"
  ]
}
```

```
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5d5a201f-0a3d-485f-a82f-3c73ccca348a"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InternalError.ResourceConfigCanNotDelete	资源只有一个版本，无法删除。
InvalidParameter.InvalidResourceIds	ResourceIds非法。
InvalidParameterValue.ResourceIdsNotFound	未找到资源ID。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。

错误码	描述
ResourceUnavailable.BeUseBySomeJobConfig	被某个作业配置使用。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

描述资源配置接口

最近更新时间：2025-04-25 01:48:24

- - - - -

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

描述资源配置接口

默认接口请求频率限制：35次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeResourceConfigs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ResourceId	否	String	资源ID 示例值：resource-xxx
Offset	否	Integer	偏移量，仅当设置 Limit 时该参数有效 示例值：0
Limit	否	Integer	返回值大小，不填则返回全量数据 示例值：20
ResourceConfigVersions.N	否	Array of Integer	资源配置Versions集合 示例值：1
JobConfigVersion	否	Integer	作业配置版本
JobId	否	String	作业ID
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
ResourceConfigSet	Array of ResourceConfigItem	资源配置描述数组 示例值：[]
TotalCount	Integer	资源配置数量 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询资源版本

查询资源版本

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: DescribeResourceConfigs
```

<公共请求参数>

```
{
  "ResourceId": "resource-qreuon9y",
  "Limit": 1,
  "WorkSpaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d20bc7a4-8382-42d1-8f8b-91ec83eac1d5",
    "ResourceConfigSet": [
      {
        "AppId": 1257058945,
        "CreateTime": "2023-11-20 15:05:56",
        "CreatorUin": "100032489861",
        "OwnerUin": "100006386216",
        "RefJobCount": 0,
        "RefJobStatusCountSet": [],
        "Region": "ap-guangzhou",
        "Remark": "test",
        "ResourceId": "resource-qreuon9y",
        "ResourceLoc": {
          "Param": {
            "Bucket": "oceanus-online-resource01-gz-1257058918",
            "Path": "1257058945/100006386216/upload/20231120150319/flink-hello-world-1.0.0.jar",
            "Region": "ap-guangzhou"
          },
          "StorageType": 1
        },
        "ResourceType": 1,
        "Status": 1,
        "Version": 3
      }
    ],
    "TotalCount": 3
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InvalidParameter.AppldResourceNotMatch	Appld资源不匹配。
InvalidParameter.UinResourceNotMatch	Uin资源不匹配。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.JobIdValueError	作业id的参数无效。
ResourceNotFound	资源不存在。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.ResourceConfig	程序包版本不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

元数据相关接口

查询元数据表

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询元数据表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：GetMetaTable。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Catalog	是	String	目录名 示例值：_dc
Database	是	String	库名 示例值：_db
Table	是	String	表名 示例值：my_table1
WorkSpaceId	是	String	空间唯一标识 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
SerialId	String	元数据表唯一标识 示例值：meta-xxx
Catalog	String	目录名 示例值：_dc
Database	String	库名 示例值：_db
Table	String	表名 示例值：my_table1
DDL	String	建表语句,使用 Base64 编码。 例如 Q1JFQVRFIGRFBQkxFIGRhZGFnZW5fc291cmNIX3RhYmxlICggCiAgICBpZCBJTlQsIAogICAgbmFtZSBTVFJJTkcgCi
CreateTime	String	创建时间
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要

4. 示例

示例1 查询元数据表信息

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: GetMetaTable
<公共请求参数>

{
  "Catalog": "_dc",
  "Database": "_db",
  "Table": "my_table1",
  "WorkSpaceId": "space-xxx"
}
```

输出示例

```
{
  "Response": {
    "SerialId": "mtable-7v16heof",
    "Catalog": "_dc",
    "Database": "_db",
    "Table": "my_table1",
    "DDL": "Q1JFQVRFI FRBQkxFIGRhdGFnZW5fc291cmNlX3RhYmxlICggCiAgICBpZCBJTlQsIAogICAgbmFtZSBTVFJJTkcgCikgV0lUSCAoCidjb25uZWNOb3InPSdkYXRhZ2VuJywKJ3Jvd3MtcGVyLXNlY29uZCcgPSAnMScKKTs=",
    "CreateTime": "2023-11-08 14:45:07",
    "RequestId": "6d1f41df-7c10-4ddd-a6e0-e05bccd996fa"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
ResourceNotFound.TableNotExist	数据库或表不存在。

权限相关接口

创建工作空间

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：`oceanus.tencentcloudapi.com`。

创建工作空间

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateWorkspace。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceName	是	String	工作空间名称 示例值：MySpace
Description	否	String	项目空间备注 示例值：default0

3. 输出参数

参数名称	类型	描述
WorkspaceId	String	工作空间 SerialId 示例值：space-1327
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建项目空间

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateWorkspace
<公共请求参数>

{
  "WorkspaceName": "abc",
```

```
"Description": "abc"
}
```

输出示例

```
{
  "Response": {
    "WorkspaceId": "space-xxxxx",
    "RequestId": "xxx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InvalidItemSpaceName	空间命名不规范。
LimitExceeded.ItemSpaceLimitExceeded	项目空间数量超出限制。
LimitExceeded.WorkSpaceLimitExceeded	工作空间超阈值。
UnsupportedOperation	操作不支持。

删除工作空间

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

删除工作空间

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteWorkSpace。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkSpaceId	是	String	工作空间 SerialId 示例值：space-1327

3. 输出参数

参数名称	类型	描述
Delete	Boolean	是否删除 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除空间

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteWorkSpace
<公共请求参数>

{
  "WorkSpaceId": "space-1327"
}
```

输出示例

```
{
  "Response": {
```

```
"Delete": true,
"RequestId": "a123"
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
FailedOperation.DeleteWedataItemspace	不能删除wedata项目关联的空间
InternalError	内部错误。
InvalidParameter	参数错误。
UnsupportedOperation.CanNotDelete	资源未释放,不能做删除作业。

修改工作空间

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

修改工作空间

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyWorkSpace。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkSpaceId	是	String	工作空间 SerialId 示例值：1327
WorkSpaceName	否	String	待修改的工作空间名称 示例值：MySpace
Description	否	String	待修改的工作空间备注 示例值：default0

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改项目空间

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyWorkSpace
<公共请求参数>

{
  "WorkSpaceId": "workspace-xxx",
  "WorkSpaceName": "MySpace0",
  "Description": "default0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.DuplicatedSpaceName	空间命名重复。
InvalidParameter.InvalidItemSpaceName	空间命名不规范。
UnsupportedOperation	操作不支持。

授权工作空间列表

最近更新时间：2025-04-25 01:48:25

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

授权工作空间列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeWorkSpaces。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	否	Integer	偏移量，默认 0 示例值：0
OrderType	否	Integer	1 按照创建时间降序排序(默认) 2.按照创建时间升序排序，3. 按照状态降序排序 4. 按照状态升序排序 默认为0 示例值：1
Limit	否	Integer	请求的集群数量，默认 20 示例值：20
Filters.N	否	Array of Filter	过滤规则 示例值：[]

3. 输出参数

参数名称	类型	描述
WorkSpaceSetItem	Array of WorkSpaceSetItem	空间详情列表 注意：此字段可能返回 null，表示取不到有效值。
TotalCount	Integer	空间总数
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 授权空间列表

获取用户的空间列表

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeWorkSpaces
```

<公共请求参数>

```
{
  "Offset": 0,
  "Limit": 5,
  "Filters": [
    {
      "Name": "ItemSpaceName",
      "Values": [
        "def"
      ]
    }
  ],
  "OrderType": 1
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 25,
    "WorkspaceSetItem": [
      {
        "SerialId": "spc-1874",
        "WorkspaceId": "spc-1874",
        "AppId": 1257058945,
        "OwnerUin": "100006386216",
        "CreatorUin": "100006386216",
        "WorkspaceName": "default0",
        "Region": "ap-guangzhou",
        "CreateTime": "2021-12-10 02:39:54",
        "UpdateTime": "2021-12-10 02:39:54",
        "Status": 2,
        "Description": "子账号在默认空间具有开发者权限，可以在成员管理模块进行限制",
        "ClusterGroupSetItem": [
          {
            "ClusterId": "cluster-jphs0vtz",
            "Name": "k1",
            "Region": "ap-guangzhou",
            "Zone": "ap-guangzhou-3",
            "AppId": 1257058945,
            "OwnerUin": "100006386216",
            "CreatorUin": "100006386216",
            "CuNum": 24,
            "CuMem": 4,
            "Status": -1,
            "StatusDesc": "unknown",
            "CreateTime": "2020-11-05 09:40:45",
            "UpdateTime": "2020-11-25 02:47:02",
            "Remark": "k1",
            "NetEnvironmentType": 1,
            "FreeCuNum": 24,
            "FreeCu": 23.5,
            "RunningCu": 0,
            "PayMode": 0
          }
        ]
      }
    ]
  }
}
```

```

],
"RoleAuth": [
{
"Id": 1363,
"AppId": 1257058945,
"WorkSpaceId": 1874,
"WorkSpaceSerialId": "workspace-xxx",
"OwnerUin": "100006386216",
"CreatorUin": "100006386216",
"AuthSubAccountUin": "100006386216",
"Permission": 0,
"CreateTime": "2021-12-10 02:40:46",
"UpdateTime": "2021-12-10 02:40:46",
"Status": 2,
"RoleName": "name"
}
],
"RoleAuthCount": 2,
"JobsCount": 5
}
],
"RequestId": "415320BF-0F15-40AB-BB57-8D3B358E0423"
}
}

```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。

集群接口

查询集群

最近更新时间：2025-04-25 01:48:24

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询集群

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeClusters。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ClusterIds.N	否	Array of String	按照一个或者多个集群 ID 查询，每次请求的集群上限为 100 示例值：cluster-xxxx
Offset	否	Integer	偏移量，默认 0 示例值：0
Limit	否	Integer	请求的集群数量，默认 20，最大值 100 示例值：20
OrderType	否	Integer	集群信息结果排序规则，1 按时间降序，2 按照时间升序，3 按照状态排序 示例值：1
Filters.N	否	Array of Filter	过滤规则 - Name 按照集群的名字进行模糊查询。例如：测试 类型：String 必选：否
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1327

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	集群总数 示例值：1
ClusterSet	Array of Cluster	集群列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询集群

查询集群

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeClusters
<公共请求参数>

{
  "ClusterIds": [
    "cluster-5c42n3a5"
  ],
  "Limit": 1,
  "WorkspaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "ClusterSet": [
      {
        "AppId": 1234567890,
        "ArchGeneration": 2,
        "AutoRenewFlag": 1,
        "CCNs": [
          {
            "CcnId": "",
            "SubnetId": "subnet-abcdefghijklmno",
            "VpcId": "vpc-pqrstuvwxyz"
          }
        ],
        "CLSLogName": "",
        "CLSLogSet": "",
        "CLSTopicId": "",
        "CLSTopicName": "",
        "ClusterId": "cluster-abcd1234",
        "ClusterSessions": [
          {
            "AppId": 1234567890,
            "ClusterGroupSerialId": "cluster-abcd1234",
            "CreateTime": "2023-11-14 18:07:52",
            "CreatorUin": "100012345678",
            "CuNum": 3,
            "FlinkVersion": "Flink-1.13",
            "JobManagerCuSpec": 1,
            "OwnerUin": "100098765432",
            "Properties": [
              {
                "Key": "pipeline.max-parallelism",
                "Value": "2048"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
"Region": "ap-guangzhou",
"Status": 1,
"TaskManagerCuSpec": 1,
"TaskManagerNum": 2,
"UpdateTime": "2023-11-14 18:09:02",
"WebUIUrl": "",
"Zone": "ap-guangzhou-7"
},
],
"ClusterType": 0,
"Correlations": [
{
"ClusterGroupId": 1234,
"ClusterGroupSerialId": "cluster-abcd1234",
"ClusterName": "sample",
"ProjectId": 0,
"ProjectIdStr": "0",
"Status": 2,
"WorkspaceId": "space-1234567890ap-guangzhou",
"WorkspaceName": "Default"
}
],
"CreateTime": "2023-08-15 17:18:08",
"CreatorUin": "100012345678",
"CuMem": 4,
"CuNum": 19,
"CustomizedDNSEnabled": 1,
"DefaultCOSBucket": "sample-gz-bucket-1234567890",
"DefaultLogCollectConf": "{\"LogCollectType\":2,\"Conf\":{\"ClsLogsetId\":\"cd9adbb5-6b7d-48d2-9870-77658959c7a4\",\"ClsToPicId\":\"80856bb3-0ef7-4e05-80eb-79ec283bb67d\",\"CosBucket\":\"\"}}",
"ExpireTime": "2023-12-04 15:10:29",
"FreeCu": 9,
"FreeCuNum": 9,
"IsNeedManageNode": 1,
"IsolatedTime": "2023-11-04 01:57:54",
"Name": "sample",
"NetEnvironmentType": 1,
"Orders": [
{
"AutoRenewFlag": 1,
"ComputeCu": 19,
"OperateUin": "100098765432",
"OrderTime": "2023-11-04 01:57:54",
"Type": 2
}
],
"OwnerUin": "100098765432",
"PayMode": 1,
"Region": "ap-guangzhou",
"Remark": "sample",
"RunningCu": 10,
"SecondsUntilExpiry": "1223380",
"SqlGateways": [],
"Status": 2,
"StatusDesc": "running",
"Tags": [
{
```

```
"TagKey": "tag1",
"TagValue": "value1"
},
{
  "TagKey": "tag2",
  "TagValue": "value2"
}
],
"UpdateTime": "2023-11-04 01:57:54",
"Version": {
  "Flink": "Flink-1.13",
  "SupportedFlink": [
    "Flink-1.11",
    "Flink-1.13",
    "Flink-1.14",
    "Flink-1.16"
  ]
},
"WebUIType": 0,
"Zone": "ap-guangzhou-7"
}
],
"RequestId": "24d0cbe8-9bd4-4356-bb95-74c5c4790877",
"TotalCount": 1
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。

错误码	描述
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
FailedOperation.GetResourceTagsByResourceIds	查询资源关联标签失败。
InternalError	内部错误。
InvalidParameterValue.ClusterIds	群集Id报错。
InvalidParameterValue.OrderType	OrderType值错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

作业接口

检查快照是否可用

最近更新时间：2025-04-25 01:48:29

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

检查快照是否可用

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CheckSavepoint。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业 id 示例值：cql-52xkpypm
SerialId	是	String	快照资源 id 示例值：svp-52xkpypm
RecordType	是	Integer	快照类型 1: savepoint; 2: checkpoint; 3: cancelWithSavepoint 示例值：1
SavepointPath	是	String	快照路径，目前只支持 cos 路径 示例值：cosn://xxx/xxx
WorkSpaceId	是	String	工作空间 id 示例值：space-1327

3. 输出参数

参数名称	类型	描述
SerialId	String	资源 id 示例值：svp-6w8eab6f
SavepointStatus	Integer	1=可用, 2=不可用 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 成功

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CheckSavepoint
<公共请求参数>

{
  "SerialId": "cql-52xkpypm",
  "WorkSpaceId": "default-12345ap-beijing",
  "SavepointPath": "cosn://52xkpypm-12345/12345/10000/cql-12345/2/flink-savepoints/savepoint-000000-12334",
  "RecordType": "1",
  "JobId": "cql-52xkpypm"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8e9e8172-17e2-4fc5-b363-1d480a4795ae",
    "SavepointStatus": 2,
    "SerialId": "cql-52xkpypm"
  }
}
```

示例2 检查快照可用性

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CheckSavepoint
<公共请求参数>

{
  "JobId": "cql-52xkpypm",
  "SerialId": "svp-52xkpypm",
  "RecordType": 1,
  "SavepointPath": "cosn://xxxx/xxxx/xxxx/chk-1",
  "WorkSpaceId": "space-12345"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d7b76d5e-ad7d-4abd-b3b2-43b96dd08d16",
    "SerialId": "svp-52xkpypm",
    "SavepointStatus": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
ResourceNotFound.Job	作业不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

复制作业

最近更新时间：2025-04-25 01:48:29

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

单条和批量复制作业

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CopyJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobItems.N	是	Array of CopyJobItem	复制明细列表
WorkSpaceId	否	String	工作空间 SerialId
TargetWorkspaceId	否	String	目标工作空间 SerialId

3. 输出参数

参数名称	类型	描述
SuccessCount	Integer	成功条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
FailCount	Integer	失败条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
CopyJobsResults	Array of CopyJobResult	结果列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 复制作业

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CopyJobs
<公共请求参数>
```

```
{
  "WorkSpaceId": "sapce-xxx",
  "JobItems": [
    {
      "SourceId": "cql-pqx9zh30",
      "SourceName": "datagen_datagen_datagen_datagen_datagen_datagen_50",
      "JobType": 1,
      "TargetClusterId": "cluster-4y035tb0",
      "TargetName": "datagen_datagen_datagen_datagen_datagen_datagen_50_2022-01-12_10_14_52",
      "TargetFolderId": "root"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "SuccessCount": 0,
    "FailCount": 0,
    "CopyJobsResults": [
      {
        "JobId": "abc",
        "JobName": "abc",
        "TargetJobName": "abc",
        "TargetJobId": "abc",
        "Message": "abc",
        "Result": 0,
        "ClusterName": "abc",
        "ClusterId": "abc",
        "JobType": 0
      }
    ],
    "RequestId": "abc"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
InternalError	内部错误。
InternalError.CLS	CLS接口错误。
InternalError.DB	数据库异常。
InternalError.StsNewClient	内部错误。
InternalError.SystemError	系统错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound.Job	作业不存在。
UnsupportedOperation	操作不支持。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

删除作业

最近更新时间：2025-04-25 01:48:29

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

批量删除作业接口，批量操作数量上限20

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobIds.N	是	Array of String	作业Id列表 示例值：[]
WorkSpaceId	否	String	工作空间Id 示例值：space-1327
JobNames.N	否	Array of String	作业名称列表 示例值：["name1","name2"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除作业

删除作业

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteJobs
<公共请求参数>

{
  "JobIds": [
    "cql-6ntmhryo"
  ],
  "JobNames": [
```

```
"name1"
},
"WorkSpaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "12470b27-ef12-4c52-9b79-bddd64ae0a5b"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.DB	数据库异常。
InternalError.LogicError	内部错误。
InvalidParameterValue.JobIdValueError	作业id的参数无效。
ResourceInUse	资源被占用。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.JobId	找不到作业。

错误码	描述
ResourceUnavailable.NotAllowedToBeDeleted	不允许删除错误。
UnsupportedOperation	操作不支持。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

获取指定作业的事件

最近更新时间：2025-06-05 01:51:06

1. 接口描述

接口请求域名：`oceanus.tencentcloudapi.com`。

请求参数不包含 "RunningOrderIds"时，接口获取指定作业的事件，包括作业启动停止、运行失败、快照失败、作业异常等各种事件类型;请求参数不包含 "RunningOrderIds"时，接口为查询作业实例ID接口,获取作业实例

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeJobEvents。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业的 ID 示例值：cql-6w8eab6f
StartTimestamp	是	Integer	筛选条件：起始 Unix 时间戳（秒） 示例值：1630932161
EndTimestamp	是	Integer	筛选条件：结束 Unix 时间戳（秒） 示例值：1631232466
Types.N	否	Array of String	事件类型。如果不传则返回所有类型的数据 示例值：[1,2]
RunningOrderIds.N	否	Array of Integer	运行实例 ID 数组 示例值：[1,2,3]
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
Events	Array of JobEvent	该作业指定范围内的事件列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
RunningOrderIds	Array of Integer	该作业指定范围内运行实例 ID 数组，仅当入参没有传入 RunningOrderIds 参数时才会返回。倒序输出 注意：此字段可能返回 null，表示取不到有效值。 示例值：[3,2,1]
TotalCount	Integer	事件的总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
Versions	Array of Integer	实例对应的版本 注意：此字段可能返回 null，表示取不到有效值。 示例值：[1,2,3]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询作业事件

请求参数不包含 "RunningOrderIds"时，接口为查询作业事件接口

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobEvents
<公共请求参数>

{
  "RunningOrderIds": [
    1
  ],
  "Types": [
    "1",
    "2"
  ],
  "EndTimestamp": 1630933910,
  "StartTimestamp": 1630833910,
  "JobId": "cql-xxx"
}
```

输出示例

```
{
  "Response": {
    "RunningOrderIds": [
      1
    ],
    "Events": [
      {
        "RunningOrderId": 1,
        "Description": "作业停止事件",
        "Timestamp": 1630933810,
        "Type": "11"
      }
    ],
    "TotalCount": 1,
    "RequestId": "123345be-e89b-52f3-a456-426616274040"
  }
}
```

示例2 查询作业运行实例ID

请求参数不包含 "RunningOrderIds"时，接口为查询作业实例ID接口

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobEvents
<公共请求参数>
```

```
{
  "WorkSpaceId": "space-xxxx",
  "JobId": "cql-xxxx",
  "StartTimestamp": 1747994802,
  "EndTimestamp": 1748513202
}
```

输出示例

```
{
  "Response": {
    "Events": null,
    "RequestId": "a908a65e-aa94-4406-acc4-2aa64edb40c6",
    "RunningOrderIds": [
      10,
      9,
      8,
      7
    ],
    "Versions": [
      1,
      2,
      3
    ],
    "TotalCount": 0
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InvalidAppId	appid错误。
InvalidParameterValue.JobIdValueError	作业id的参数无效。
InvalidParameterValue.Timestamp	时间戳范围有误。
InvalidParameterValue.Type	非法的类型字段。
ResourceNotFound.Job	作业不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

查找Savepoint列表

最近更新时间：2025-04-25 01:48:28

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查找Savepoint列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeJobSavepoint。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业 SerialId 示例值：cql-asdf5678
Limit	是	Integer	分页参数，单页总数 示例值：1
Offset	是	Integer	分页参数，偏移量 示例值：1
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx
RecordTypes.N	否	Array of Integer	2 是checkpoint 1 是触发savepoint 3 停止触发的savepoint 示例值：[]

3. 输出参数

参数名称	类型	描述
TotalNumber	Integer	快照列表总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Savepoint	Array of Savepoint	快照列表 注意：此字段可能返回 null，表示取不到有效值。
RunningSavepoint	Array of Savepoint	进行中的快照列表 注意：此字段可能返回 null，表示取不到有效值。
RunningTotalNumber	Integer	进行中的快照列表总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例11

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobSavepoint
<公共请求参数>

{
  "Offset": 0,
  "Limit": 20,
  "JobId": "cql-asdf5678"
}
```

输出示例

```
{
  "Response": {
    "TotalNumber": 0,
    "Savepoint": [
      {
        "Id": 0,
        "VersionId": 0,
        "Status": 0,
        "CreateTime": 0,
        "UpdateTime": 0,
        "Path": "abc",
        "Size": 0,
        "RecordType": 0,
        "JobRuntimeId": 0,
        "Description": "abc",
        "Timeout": 0,
        "SerialId": "abc",
        "TimeConsuming": 0,
        "PathStatus": 0
      }
    ],
    "RunningSavepoint": [
      {
        "Id": 0,
        "VersionId": 0,
        "Status": 0,
        "CreateTime": 0,
        "UpdateTime": 0,
        "Path": "abc",
        "Size": 0,
        "RecordType": 0,
        "JobRuntimeId": 0,
        "Description": "abc",
        "Timeout": 0,
        "SerialId": "abc",
        "TimeConsuming": 0,
        "PathStatus": 0
      }
    ]
  }
}
```

```

    }
  ],
  "RunningTotalNumber": 0,
  "RequestId": "abc"
}
}

```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InternalError	内部错误。
ResourceNotFound.JobId	找不到作业。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

更新作业

最近更新时间：2025-04-25 01:48:28

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

更新作业属性，仅允许以下3种操作，不支持组合操作：

- (1) 更新作业名称
- (2) 更新作业备注
- (3) 更新作业最大并行度

变更前提：WorkerCuNum<=MaxParallelism

如果MaxParallelism变小，不重启作业，待下一次重启生效

如果MaxParallelism变大，则要求入参RestartAllowed必须为True

假设作业运行状态，则先停止作业，再启动作业，中间状态丢失

假设作业暂停状态，则将作业更改为停止状态，中间状态丢失

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyJob。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业Id 示例值：cql-6iagingqp
Name	否	String	作业名称，支持长度小于50的中文/英文/数字/" -" /" _" /" ."，不能重名 示例值：test
Remark	否	String	描述 示例值：测试
TargetFolderId	否	String	拖拽文件需传入此参数 示例值：folder-xxxxxxx
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1237
Description	否	String	作业描述 示例值：description

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新作业

更新作业

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyJob
<公共请求参数>

{
  "JobId": "cql-glegwraz",
  "Name": "modify_test",
  "WorkSpaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "db5c46b4-b927-4a24-963d-a24e333cc78f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。

错误码	描述
FailedOperation	操作失败。
FailedOperation.DuplicatedJobName	重复的作业名。
InternalError.JobInstanceNotFound	作业实例没找到。
InternalError.SystemError	系统错误。
InvalidParameterValue.JobName	Illegal JobName。
ResourceNotFound.Job	作业不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

运行作业

最近更新时间：2025-04-25 01:48:28

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

批量启动或者恢复作业，批量操作数量上限20

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：RunJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
RunJobDescriptions.N	是	Array of RunJobDescription	批量启动作业的描述信息
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1237

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 运行作业示例

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: RunJobs
<公共请求参数>

{
  "RunJobDescriptions": [
    {
      "RunType": 0,
      "JobConfigVersion": 1,
      "StartMode": "xx",
      "JobId": "xx",
      "SavepointId": "cql-asdf5678",
      "SavepointPath": "cosn://xxxx"
```

```
}
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "18e0bde4-3922-42ca-8bfd-a36eaa035da4"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalServerError.COSClient	COS 服务访问错误。
InternalServerError.DB	数据库异常。
InternalServerError.FailedToUpdateJob	无法更新作业错误。
InternalServerError.LogicError	内部错误。
InternalServerError.StsNewClient	内部错误。
InvalidParameter.InvalidClusterId	无效集群id。
InvalidParameter.InvalidRegion	无效Region。

错误码	描述
InvalidParameterValue.InvalidStartMode	无效启动模式。
InvalidParameterValue.JobIdValueError	作业id的参数无效。
InvalidParameterValue.RunJobDescriptionsCount	批量运行作业个数超过上限。
InvalidParameterValue.RunType	RunType错误。
ResourceInUse.InstanceInProcess	集群有其他操作。
ResourceInsufficient.CU	CU资源不足。
ResourceNotFound	资源不存在。
ResourceNotFound.COSBucket	COS Bucket 未找到或无权限访问。
ResourceNotFound.ClusterId	集群不存在。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.JobConfig	作业配置版本不存在。
ResourceNotFound.Resource	程序包不存在。
ResourceUnavailable	资源不可用。
ResourceUnavailable.JobResourceConfigNotReady	作业资源配置未就绪。
ResourceUnavailable.NoRunningJobInstancesFoundForJobId	找不到JobId的正在运行的作业实例。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。
UnsupportedOperation.UnsupportedStartMode	不支持的启动模式。

停止作业

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

批量停止作业，批量操作数量上限为20

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StopJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
StopJobDescriptions.N	是	Array of StopJobDescription	批量停止作业的描述信息
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1237

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 批量停止作业示例

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StopJobs
<公共请求参数>

{
  "StopJobDescriptions": [
    {
      "StopType": 0,
      "JobId": "cql-glegwraz"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d7b76d5e-ad7d-4abd-b3b2-43b96dd08d16"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.LogicError	内部错误。
InternalError.ResourceNotExist	资源不存在。
InvalidParameterValue.UnknownStopType	未知停止类型错误。
ResourceInUse	资源被占用。
ResourceNotFound.COSBucket	COS Bucket 未找到或无权限访问。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.ResourceNotExist	资源不存在。
ResourceUnavailable.NoRunningJobInstancesFoundForJobId	找不到JobId的正在运行的作业实例。
ResourceUnavailable.NotAllowedToBeStopOrPause	不允许停止或暂停错误。
UnsupportedOperation	操作不支持。

错误码	描述
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

触发Savepoint

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

触发Savepoint

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：TriggerJobSavepoint。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业 SerialId 示例值：cqj-12345
Description	否	String	描述 示例值：Description
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1327

3. 输出参数

参数名称	类型	描述
SavepointTrigger	Boolean	是否成功 示例值：true
ErrorMsg	String	错误消息 注意：此字段可能返回 null，表示取不到有效值。 示例值：errmsg
FinalSavepointPath	String	快照路径 注意：此字段可能返回 null，表示取不到有效值。 示例值：cos
SavepointId	String	快照 ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：cqj-xxx
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 触发快照

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: TriggerJobSavepoint
<公共请求参数>
```

```
{
  "JobId": "cql-52xkpymp"
}
```

输出示例

```
{
  "Response": {
    "ErrorMsg": "",
    "FinalSavepointPath": "",
    "SavepointTrigger": true,
    "RequestId": "cc75a49d-a99b-4840-b3cb-7d3bdd9a0e33",
    "SavepointId": "svp-asdf5678"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

新建作业

最近更新时间：2025-04-25 01:48:29

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

新建作业接口，一个 AppId 最多允许创建1000个作业

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateJob。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Name	是	String	作业名称，允许输入长度小于50个字符的中文、英文、数字、-（横线）、_（下划线）、.（点），且符号必须半角字符。注意作业名不能和现有作业同名 示例值：job_new
JobType	是	Integer	作业的类型，1 表示 SQL 作业，2 表示 JAR 作业 示例值：2
ClusterType	是	Integer	集群的类型，1 表示共享集群，2 表示独享集群 示例值：2
ClusterId	否	String	当 ClusterType=2 时，必选，用来指定该作业提交的独享集群 ID 示例值：cluster-xxxxxxxx
CuMem	否	Integer	设置每 CU 的内存规格，单位为 GB，支持 2、4、8、16（需申请开通白名单后使用）。默认为 4，即 1 CU 对应 4 GB 的运行内存 示例值：4
Remark	否	String	作业的备注信息，可以随意设置 示例值：测试作业
FolderId	否	String	作业名所属文件夹ID，根目录为"root" 示例值：folder-xxxxxxxx
FlinkVersion	否	String	作业运行的Flink版本 示例值：Flink-1.11
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1327
Tags.N	否	Array of Tag	作业标签 示例值：1
Description	否	String	作业描述 示例值：description of job
OpenJobDefaultAlarm	否	Integer	开启默认告警 示例值：1

3. 输出参数

参数名称	类型	描述
JobId	String	作业Id 示例值: cql-qlpn5o2a
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建作业

创建作业

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateJob
<公共请求参数>

{
  "Name": "create_job_test",
  "JobType": 1,
  "ClusterType": 2,
  "ClusterId": "cluster-5c42n3a5",
  "WorkSpaceId": "space-53rqk422",
  "Description": "some description"
}
```

输出示例

```
{
  "Response": {
    "JobId": "cql-e92rhppb",
    "RequestId": "9159e720-d61c-459f-862b-d630de3209dd"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
FailedOperation.DuplicatedJobName	重复的作业名。
FailedOperation.UserNotAuthenticated	用户未实名认证。
InternalServerError	内部错误。
InternalServerError.DB	数据库异常。
InternalServerError.LogicError	内部错误。
InvalidParameter.InvalidName	名字不符合规范。
InvalidParameterValue.ClusterId	集群Id指定错误，为空或者不存在。
InvalidParameterValue.CuMem	CU内存规格不匹配。
InvalidParameterValue.JobName	Illegal JobName。
InvalidParameterValue.JobNameExisted	作业名称已存在。
InvalidParameterValue.JobTypeCombineWithClusterType	集群模式与作业类型不匹配。
InvalidParameterValue.UnSupportedComposite	不支持的复合类型。
LimitExceeded	超过配额限制。
LimitExceeded.Job	Job个数超过限额。
ResourceNotFound.ClusterId	集群不存在。
ResourceUnavailable.Cluster	集群处于非运行状态。
ResourceUnavailable.ClusterGroupStatus	群集组状态错误。
ResourceUnavailable.RegCuMem	共享群集只允许CuMem=4。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

查询作业

最近更新时间：2025-05-28 01:23:45

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询作业

默认接口请求频率限制：30次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobIds.N	否	Array of String	按照一个或者多个作业ID查询。作业ID形如：cql-11112222，每次请求的作业上限为100。参数不支持同时指定 JobIds和Filters。 示例值：["cql-123"]
Filters.N	否	Array of Filter	过滤条件，支持的 Filter.Name 为：作业名 Name、作业状态 Status、所属集群 ClusterId、作业id JobId、集群名称 ClusterName。每次请求的 Filters 个数的上限为 5，Filter.Values 的个数上限为 5。参数不支持同时指定 JobIds 和 Filters。
Offset	否	Integer	偏移量，默认为0 示例值：0
Limit	否	Integer	分页大小，默认为20，最大值为100 示例值：20
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1327
ExtraResult.N	否	Array of String	查询额外的作业信息,例如 JobEventInfo 示例值：["JobEventInfo"]
ConnectorOptions	否	String	查询引用connector 示例值：setats

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	作业总数 示例值：1
JobSet	Array of JobV1	作业列表 示例值：[{}]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询作业

查询作业

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobs
<公共请求参数>

{
  "Limit": 1,
  "WorkspaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "JobSet": [
      {
        "AppId": 1257048945,
        "ClusterId": "cluster-5c43n3a5",
        "ClusterName": "cluster-test",
        "ClusterStatus": 2,
        "CreateTime": "2023-11-20 15:19:57",
        "CreatorUin": "100032489761",
        "CuMem": 4,
        "CurrentRunMillis": 0,
        "FlinkVersion": "Flink-1.13",
        "JobId": "cq1-e92rh2pb",
        "JobType": 1,
        "LastOpResult": "",
        "LatestJobConfigVersion": 0,
        "Name": "create_job_test",
        "OwnerUin": "100006376216",
        "PublishedJobConfigVersion": -1,
        "Region": "ap-guangzhou",
        "Remark": "",
        "RunningCu": 0,
        "RunningCuNum": 0,
        "SchedulerType": 2,
        "StartTime": "2025-11-11 11:11:11",
        "Status": 1,
        "StatusDesc": "create",
        "StopTime": "2025-11-11 11:11:11",
        "Tags": null,
        "TotalRunMillis": 0,
        "UpdateTime": "2023-11-20 15:19:57",
        "WebUIUrl": "",
        "WorkspaceId": "space-53rek422",
        "WorkspaceName": "name",
        "Zone": "ap-guangzhou-7"
      }
    ]
  }
}
```

```
"RequestId": "ac6a62a7-45f1-4066-963d-eabea0f2824d",
"TotalCount": 5
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.CamCgwError	CAM 网关错误。
InternalError.DB	数据库异常。
ResourceNotFound.JobId	找不到作业。
ResourceUnavailable.GetJobPublishedJobConfig	获取发布的作业配置错误。
UnsupportedOperation	操作不支持。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

执行Statement

最近更新时间：2025-04-25 01:48:28

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

通过Sql gateway执行satement

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：RunSqlGatewayStatement。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ClusterId	是	String	集群ID 示例值：cluster-lq6dtnk5
Sql	是	String	需要执行的sql，该sql会被Sql Gateway执行，当前支持的是paimon修改需求，因此主要是DDL语句 示例值：select 1
SessionId	否	String	Sql Gateway会话ID，可不填，如果不填则会自动创建一个会话ID，每个会话ID都有一个存活时间，测试环境为10分钟，线上默认是30分钟 示例值：b523f113-aa32-40cc-9665-5884a1dd480d

3. 输出参数

参数名称	类型	描述
ErrorMessage	Array of String	错误信息 注意：此字段可能返回 null，表示取不到有效值。
SessionId	String	会话id，若入参未传，则返回自动创建的会话id，若入参已经传递，则返回值与原传入值一致 示例值：b523f113-aa32-40cc-9665-5884a1dd480d
OperationHandleId	String	返回执行id，可以根据该执行id和会话id获取执行结果 示例值：操作ID
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 执行Statement

执行Statement

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: RunSqlGatewayStatement
```

<公共请求参数>

```
{
  "ClusterId": "abc",
  "SessionId": "abc",
  "Sql": "abc"
}
```

输出示例

```
{
  "Response": {
    "ErrorMessage": [
      "abc"
    ],
    "SessionId": "abc",
    "OperationHandleId": "abc",
    "RequestId": "abc"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.FailedOperationCode_RunSqlGateway	运行SqlGateway失败，异常

查询Statement执行结果

最近更新时间：2025-04-25 01:48:28

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询Sql Gateway的Statement执行结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：FetchSqlGatewayStatementResult。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ClusterId	是	String	集群ID 示例值：cluster-lq6dtnk5
SessionId	否	String	Sql Gateway会话ID 示例值：b523f113-aa32-40cc-9665-5884a1dd480d
OperationHandleId	否	String	sql的查询id 示例值：f613c399-e1c7-4ab9-a07f-ee08a782b546
ResultUri	否	String	下一条结果的获取url，首次获取执行结果时可以为空，当获取下一批查询结果时需要传递 示例值：/v1/sessions/b523f113-aa32-40cc-9665-5884a1dd480d/operations/f613c399-e1c7-4ab9-a07f-ee08a782b546/result/1

3. 输出参数

参数名称	类型	描述
ErrorMessage	Array of String	错误信息 注意：此字段可能返回 null，表示取不到有效值。
ResultType	String	返回类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：PAYLOAD
IsQueryResult	Boolean	是否DQL结果 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
ResultKind	String	结果类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：SUCCESS
Results	StatementResult	结果 注意：此字段可能返回 null，表示取不到有效值。
NextResultUri	String	下一次请求的uri 注意：此字段可能返回 null，表示取不到有效值。

参数名称	类型	描述
		示例值: /v1/sessions/1c7a1105-4759-46d4-b35f-3a21f7a7cf8d/operations/a675de79-58c7-44cf-910f-bdae095041bf/result/1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询Statement执行结果

查询Statement执行结果

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: FetchSqlGatewayStatementResult
<公共请求参数>

{
  "ClusterId": "abc",
  "SessionId": "abc",
  "OperationHandleId": "abc",
  "ResultUri": "abc"
}
```

输出示例

```
{
  "Response": {
    "ErrorMessage": [
      "abc"
    ],
    "ResultType": "abc",
    "IsQueryResult": true,
    "ResultKind": "abc",
    "Results": {
      "Columns": [
        {
          "Name": "abc",
          "LogicalType": {
            "Type": "abc",
            "Nullable": true,
            "Length": 0
          },
          "Comment": "abc"
        }
      ],
      "RowFormat": "abc",
      "Data": [
        {
          "Kind": "abc",
          "Fields": [
            "abc"
          ]
        }
      ]
    }
  }
}
```

```
}  
]  
,  
"NextResultUri": "abc",  
"RequestId": "abc"  
}  
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.FailedOperationCode_FetchSqlGatewayStatement	获取SqlGateway结果失败，异常

获取作业运行时的信息

最近更新时间：2025-04-25 01:48:29

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

获取作业运行时的信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeJobRuntimeInfo。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID 示例值：cql-xxx
WorkSpaceId	是	String	工作空间ID 示例值：space-xxx
IncludeInfo.N	是	Array of String	作业运行信息 key 示例值：["TaskManagers","SubTasks"]

3. 输出参数

参数名称	类型	描述
JobRuntimeInfo	Array of JobRuntimeInfo	作业运行时信息
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询作业的taskmanger和subtask

查询作业的taskmanger和subtask

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobRuntimeInfo
<公共请求参数>

{
  "JobId": "cql-xxx",
```

```
"WorkspaceId": "space-xxx",
"IncludeInfo": [
  "TaskManagers",
  "SubTasks"
]
}
```

输出示例

```
{
  "Response": {
    "JobRuntimeInfo": [
      {
        "Key": "TaskManagers",
        "Value": "WyJjcWwtbDkzZzJ3eGotNDI0MS10YXNrZWVud1ci0xLTEiXQ=="
      },
      {
        "Key": "SubTasks",
        "Value": "W3siVGFza0lkIjoY2JjMzU3Y2NiNzYzZGYyODUyZmVlOGM0ZmM3ZDU1ZjIiLCJUYXNrRGVzY3JpcHRpb24iOiJTb3VyY2U6IFRhYmx1U291cmNlU2Nhbih0YWJsZT1bW2RlZmF1bHRfY2F0YWxvZywgZGVmYXVsdF9kYXRhYmFzZSsgZGF0YWdlb19zb3VyY2VfdGFibGVdXSwgZml1bGRzPVtpZCwgYmFtZV0pIC1cdTAWMjZndDsgU2luazogU2luayh0YWJsZT1bZGVmYXVsdF9jYXRhbG9nLmRlZmF1bHRfZGF0YWJhc2UubG9nZ2VyX3NpbmtfdGFibGVdLCBmaWVsZHM9W2lkLzCBuYW1lXSlfMCIsIlN1Y1RhczRlZmRleCI6MH1d"
      }
    ],
    "RequestId": "51964833-6e61-4909-b93b-3d4996c9360d"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

作业配置接口

创建作业配置

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：`oceanus.tencentcloudapi.com`。

创建作业配置，一个作业最多有100个配置版本

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateJobConfig。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业Id 示例值：cql-abcdefg
EntrypointClass	否	String	主类 示例值：com.tencent.cloud.test.WordCount
ProgramArgs	否	String	主类入参，需要区分下Sql作业配置，Jar作业配置，Python作业配置，具体参考下面的示例值 示例值：--conf Key=Value
Remark	否	String	备注 示例值：example
ResourceRefs.N	否	Array of ResourceRef	资源引用数组 示例值：[{ "ResourceId": "resource-lbi8h92t", "Type": 1, "Version": 1 }]
DefaultParallelism	否	Integer	作业默认并行度 示例值：2
Properties.N	否	Array of Property	系统参数 示例值：[{ "Key": "pipeline.max-parallelism", "Value": "2048" }]
AutoDelete	否	Integer	1: 作业配置达到上限之后，自动删除可删除的最早版本 示例值：1
COSBucket	否	String	作业使用的 COS 存储桶名 示例值：example-oceanus-1257058918
LogCollect	否	Boolean	是否采集作业日志 示例值：true
JobManagerSpec	否	Float	JobManager规格 示例值：0.5
TaskManagerSpec	否	Float	TaskManager规格 示例值：0.5
ClsLogsetId	否	String	CLS日志集ID

参数名称	必选	类型	描述
			示例值：544002e3-299b-41d2-b18f-19194fb0660b
ClsTopicId	否	String	CLS日志主题ID 示例值：544002e3-299b-41d2-b18f-19194fb0660b
LogCollectType	否	Integer	日志采集类型 2：CLS；3：COS 示例值：2
PythonVersion	否	String	pyflink作业运行时使用的python版本 示例值：Python-3.7
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1327
LogLevel	否	String	日志级别 示例值：INFO
AutoRecover	否	Integer	Oceanus 平台恢复作业开关 1:开启 -1: 关闭 示例值：1
ClazzLevels.N	否	Array of ClazzLevel	类日志级别 示例值：[]
ExpertModeOn	否	Boolean	是否打开专家模式 示例值：false
ExpertModeConfiguration	否	ExpertModeConfiguration	专家模式的配置 示例值：{ }
TraceModeOn	否	Boolean	trace链路 示例值：false
TraceModeConfiguration	否	TraceModeConfiguration	trace链路配置 示例值：{ }
CheckpointRetainedNum	否	Integer	checkpoint保留个数 示例值：1
JobGraph	否	JobGraph	算子拓扑图 示例值：{ "Nodes": [], "Edges": [] }
EsServerlessIndex	否	String	es索引名称
EsServerlessSpace	否	String	es索引空间
FlinkVersion	否	String	flink版本 示例值：Flink-1.13
JobManagerCpu	否	Float	JobManager cpu 示例值：1
JobManagerMem	否	Float	JobManager 内存 示例值：1
TaskManagerCpu	否	Float	TaskManager cpu 示例值：1
TaskManagerMem	否	Float	TaskManager 内存 示例值：1

3. 输出参数

参数名称	类型	描述
Version	Integer	作业配置版本号 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

示例1 新建Jar作业配置示例

输入示例

输出示例

示例2 创建SQL作业配置示例

其中ProgramArgs参数包括了大量的入参，SqlCode、Metadata都采用了base64加密

输入示例

第155 共220页

```
"ResourceRefs": [],
"DefaultParallelism": 1,
"Properties": [
{
"Key": "pipeline.max-parallelism",
"Value": "2048"
}
],
"LogCollect": true,
"ClsLogsetId": "10981e21-ae9-4a01-9ed9-eadf5956fe3c",
"ClsTopicId": "544002e3-299b-41d2-b18f-19194fb0660b",
"LogCollectType": 2,
"WorkSpaceId": "space-1257058945ap-guangzhou",
"LogLevel": "INFO",
"AutoRecover": 1,
"ClazzLevels": [],
"ExpertModeOn": false,
"ExpertModeConfiguration": {},
"TraceModeOn": false,
"TraceModeConfiguration": {},
"CheckpointRetainedNum": 0,
"JobGraph": {
"Nodes": [],
"Edges": []
},
"EsServerlessIndex": "",
"EsServerlessSpace": "",
"FlinkVersion": "Flink-1.16",
"JobManagerCpu": 1,
"JobManagerMem": 4,
"TaskManagerCpu": 1,
"TaskManagerMem": 4
}
```

输出示例

```
{
"Response": {
"RequestId": "8f152cf4-9a43-40fa-a2ea-f28e74cd24ff",
"Version": 3
}
}
```

示例3 创建Python作业配置示例

创建python作业配置示例

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateJobConfig
<公共请求参数>

{
"JobId": "cql-68w5vlip",
```

```
"EntrypointClass": "com.tencent.cloud.oceanus.wordcount.WordCount",
"ProgramArgs": "",
"Remark": "v1",
"ResourceRefs": [
{
"ResourceId": "resource-lbi8h92t",
"Version": 1,
"Type": 1
}
],
"DefaultParallelism": 1,
"Properties": [
{
"Key": "pipeline.max-parallelism",
"Value": "2048"
}
],
"LogCollect": true,
"ClsLogsetId": "10981e21-ae9-4a01-9ed9-eadf5956fe3c",
"ClsTopicId": "544002e3-299b-41d2-b18f-19194fb0660b",
"LogCollectType": 2,
"PythonVersion": "Python-3.7",
"WorkSpaceId": "space-1257058945ap-guangzhou",
"LogLevel": "INFO",
"AutoRecover": 1,
"ClazzLevels": [],
"ExpertModeOn": false,
"ExpertModeConfiguration": {},
"TraceModeOn": false,
"TraceModeConfiguration": {},
"CheckpointRetainedNum": 0,
"JobGraph": {
"Nodes": [],
"Edges": []
},
"EsServerlessIndex": "",
"EsServerlessSpace": "",
"FlinkVersion": "Flink-1.13",
"JobManagerCpu": 0,
"JobManagerMem": 0,
"TaskManagerCpu": 0,
"TaskManagerMem": 0
}
```

输出示例

```
{
"Response": {
"RequestId": "8b4433c9-3efc-4afc-a2bd-78d682f5c3dd",
"Version": 2
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
FailedOperation.GrammarCheckFailure	语法检查失败。
FailedOperation.ParseSql	SQL解析失败。
InternalError.COSClient	COS 服务访问错误。
InternalError.DB	数据库异常。
InternalError.LogicError	内部错误。
InternalError.SqlCodeNotFound	未找到sqlcode错误。
InternalError.StsNewClient	内部错误。
InvalidParameter.IllegalMaxParallelism	非法的 MaxParallelism 参数。
InvalidParameter.InvalidRegion	无效Region。
InvalidParameter.JobConfigLogCollectParamError	创建作业配置错误。
InvalidParameter.MaxParallelismTooLarge	MaxParallelism 过大。
InvalidParameter.MaxParallelismTooSmall	MaxParallelism 不允许小于算子默认并行度。
InvalidParameter.UnsupportedFlinkConf	Flink参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.JobTypeCombineWithEntrypointClass	SQL作业不能指定EntrypointClass，JAR作业则必须指定。
LimitExceeded	超过配额限制。
LimitExceeded.JobConfig	作业配置超出限制。
ResourceNotFound	资源不存在。

错误码	描述
ResourceNotFound.COSBucket	COS Bucket 未找到或无权限访问。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.JobId	找不到作业。
UnsupportedOperation.InvalidCheckpointIntervalError	Checkpoint 时间间隔，错误。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

删除作业配置

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

删除作业配置

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteJobConfigs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID 示例值：cql-ses24dtd
JobConfigVersions.N	是	Array of Integer	作业配置版本数组 示例值：["1\n","2\n"]
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除作业配置样例

输入示例

```
https://oceanus.tencentcloudapi.com/?Action=DeleteJobConfigs
&JobId=cql-ses24dtd
&JobConfigVersions.0=1
&JobConfigVersions.1=2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "d7b76d5e-ad7d-4abd-b3b2-43b96dd08d16"
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation.DraftConfigCanNotDelete	作业的草稿不允许删除。
FailedOperation.JobConfigOnPublish	作业配置在使用中。
InternalError	内部错误。
InternalError.DB	数据库异常。
ResourceNotFound	资源不存在。
ResourceNotFound.Job	作业不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

查询作业配置

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询作业配置列表，一次最多查询100个

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeJobConfigs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业Id 示例值：cql-xxxx
JobConfigVersions.N	否	Array of Integer	作业配置版本 示例值：[]
Offset	否	Integer	偏移量，默认0 示例值：0
Limit	否	Integer	分页大小，默认20，最大100 示例值：20
Filters.N	否	Array of Filter	过滤条件 示例值：[]
OnlyDraft	否	Boolean	true 表示只展示草稿 示例值：false
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总的配置版本数量 示例值：1
JobConfigSet	Array of JobConfig	作业配置列表 示例值：[]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询作业配置

查询指定的作业配置

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeJobConfigs
<公共请求参数>

{
  "JobId": "cql-ahnkqtay",
  "WorkSpaceId": "space-1257058945ap-guangzhou"
}
```

输出示例

```
{
  "Response": {
    "JobConfigSet": [
      {
        "AutoRecover": 1,
        "COSBucket": "autotest-gz-bucket-1257058945",
        "CheckpointRetainedNum": 0,
        "ClazzLevels": [],
        "ClsLogsetId": "",
        "ClsTopicId": "",
        "CreateTime": "2023-10-20 10:55:12",
        "CreatorUin": "100032959577",
        "DefaultParallelism": 1,
        "EntrypointClass": "",
        "ExpertModeOn": false,
        "JobId": "cql-ahnkqtay",
        "JobManagerSpec": 1,
        "LogCollect": 4,
        "LogLevel": "INFO",
        "MaxParallelism": 2048,
        "ProgramArgs": "{\"CheckpointInterval\":600,\"Metadata\":{\"Metadata\":{\"ReferenceTables\":[],\"Variables\":[]}}}\",\"SqlCode\":\"-- Datagen Connector 可以随机生成一些数据用于测试\\n-- 参见 https://cloud.tencent.com/document/product/849/58713\\nCREATE TABLE datagen_source_table_b ( \\n id INT, \\n name STRING \\n) WITH ( \\n 'connector' = 'datagen', \\n 'rows-per-second'='1' -- 每秒产生的数据条数\\n);\\n\\n\\n-- 输入到 Blackhole Sink 的数据, 会被全部丢弃。这个 Sink 适合做性能测试。\\n-- 参见 https://ci.apache.org/projects/flink/flink-docs-release-1.11/zh/dev/table/connectors/blackhole.html\\nCREATE TABLE blackhole_sink_b ( \\n id INT, \\n name STRING \\n) WITH ( \\n 'connector' = 'blackhole'\\n);\\n\\n\\nINSERT INTO blackhole_sink_b SELECT * FROM datagen_source_table_b;\\n\\n\\n\\n\"}",
        "Properties": [
          {
            "Key": "pipeline.max-parallelism",
            "Value": "2048"
          }
        ],
        "PythonVersion": "",
        "Remark": "v1",
        "ResourceRefDetails": null,
        "TaskManagerSpec": 1,
        "TraceModeConfiguration": null,
        "TraceModeOn": false,
        "UpdateTime": "2023-11-13 17:19:04",
      }
    ]
  }
}
```

```
"Version": 1
}
],
"RequestId": "21ac3206-2b1a-4095-93cd-ceed11cf5379",
"TotalCount": 1
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
InternalError.DB	数据库异常。
InternalError.LogicError	内部错误。
InvalidParameter.IllegalMaxParallelism	非法的 MaxParallelism 参数。
InvalidParameter.MaxParallelismTooLarge	MaxParallelism 过大。
InvalidParameter.MaxParallelismTooSmall	MaxParallelism 不允许小于算子默认并行度。
ResourceNotFound	资源不存在。
ResourceNotFound.Job	作业不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

其他接口

创建文件夹

最近更新时间：2025-04-25 01:48:27

1. 接口描述

接口请求域名：`oceanus.tencentcloudapi.com`。

作业列表页面新建文件夹请求

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateFolder。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FolderName	是	String	新建文件夹名 示例值：joylyu-test
ParentId	是	String	新建文件夹的父目录ID（根目录为"root"） 示例值：folder-xxxxxxx
FolderType	否	Integer	文件夹类型，0是任务文件夹，1是依赖文件夹 示例值：0
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
FolderId	String	新建文件夹的唯一ID 示例值：folder-xxxxxxx
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建文件夹

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateFolder
<公共请求参数>
```

```
{
  "FolderName": "test",
  "WorkSpaceId": "space-1257058945ap-guangzhou",
  "FolderType": 0,
  "ParentId": "root"
}
```

输出示例

```
{
  "Response": {
    "FolderId": "folder-dhq5omr1",
    "RequestId": "a6030a9d-d791-4e7f-8f2c-7d908b6449a0"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

删除文件夹

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

作业列表删除文件夹

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteFolders。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FolderIds.N	是	Array of String	需删除的文件夹唯一ID 示例值：["folder-xxxxxxx"]
FolderType	否	Integer	文件夹类型，0是任务文件夹，1是依赖文件夹 示例值：0
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除文件夹

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action:DeleteFolders
<公共请求参数>

{
  "FolderIds": [
    "joylyu-test"
  ],
```

```
"FolderType": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5d5a201f-0a3d-485f-a82f-3c73ccca348a"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalServerError	内部错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

删除作业表配置

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

删除作业表配置

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteTableConfig。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID 示例值：cql-xxx
DebugId	是	Integer	调试作业ID 示例值：123
TableName	是	String	表名 示例值：a
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除作业表配置

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action:DeleteFolders
<公共请求参数>

{
  "JobId": "cql-6ntmhryo",
  "DebugId": 1,
```

```
"TableName": "tabale_test"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3998cdae-e624-4e23-b497-17dbf8c90b0f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
InternalError	内部错误。
InternalError.DB	数据库异常。

自定义树状结构

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

生成树状作业显示结构

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTreeJobs。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	筛选条件字段
WorkSpaceId	否	String	工作空间 Serialid 示例值：space-1327

3. 输出参数

参数名称	类型	描述
ParentId	String	父节点ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：root-xxx
Id	String	当前文件夹ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：folder-xxx
Name	String	当前文件夹名 注意：此字段可能返回 null，表示取不到有效值。 示例值：New Folder
JobSet	Array of TreeJobSets	当前文件夹下的作业列表 注意：此字段可能返回 null，表示取不到有效值。
Children	Array of DescribeTreeJobsRsp	迭代子目录 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 分类中的自定义树状结构并筛选

用户点击作业管理页面中的分类的自定义树状结构并点击分类旁的筛选按钮选择作业类型

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeTreeJobs
<公共请求参数>
```

```
{
  "Filters": [
    {
      "Name": "JobType",
      "Values": [
        "1",
        "2"
      ]
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Children": [
      {
        "ParentId": "root",
        "Id": "folder-xxx",
        "Name": "test",
        "JobSet": [
          {
            "JobId": "cql-xxxx",
            "Name": "test_xxxx",
            "JobType": 1,
            "RunningCu": 0,
            "Status": 5
          },
          {
            "JobId": "cql-xxxx",
            "Name": "test_xxxx",
            "JobType": 1,
            "RunningCu": 0,
            "Status": 5
          }
        ],
        "Children": null,
        "RequestId": "015e7c7e-dad7-4bdb-9a16-b348ef8c1457"
      }
    ],
    "Id": "root",
    "JobSet": [],
    "Name": "作业列表",
    "ParentId": "",
    "RequestId": "015e7c7e-dad7-4bdb-9a16-b348ef8c1457"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

拖拽文件夹

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

自定义树状结构页面拖拽文件夹

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyFolder。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SourceFolderId	否	String	文件夹ID（必填） 示例值：folder-aaaaaaa
TargetFolderId	否	String	如需拖拽文件夹，需传入目标文件夹ID 示例值：folder-bbbbbbbb
FolderName	否	String	如需修改文件夹名，需传入FolderName字段 示例值：xxxxxxx
FolderType	否	Integer	文件夹类型，0是任务文件夹，1是依赖文件夹 示例值：0
SourceJobIds.N	否	Array of String	批量移动的作业serial id 列表
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-1327

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 拖拽文件夹

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action:ModifyFolder
<公共请求参数>
```

```
{
  "SourceFolderId": "folder-aaaaaaaa",
  "TargetFolderId": "folder-bbbbbbbb",
  "FolderName": "",
  "FolderType": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5d5a201f-0a3d-485f-a82f-3c73ccca348a"
  }
}
```

示例2 修改文件夹名

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action:ModifyFolder
<公共请求参数>

{
  "SourceFolderId": "folder-aaaaaaaa",
  "TargetFolderId": "",
  "FolderName": "joylyutest",
  "FolderType": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5d5a201f-0a3d-485f-a82f-3c73ccca348a"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalServerError	内部错误。
InvalidParameterValue	参数取值错误。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

查询树状结构资源列表

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询树状结构资源列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTreeResources。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	筛选条件字段
WorkSpaceId	否	String	工作空间 SerialId 示例值：space-xxx
Offset	否	Integer	分页游标 示例值：0
Limit	否	Integer	单页显示数 示例值：200

3. 输出参数

参数名称	类型	描述
ParentId	String	父节点ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：root
Id	String	文件夹ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：folder-id
Name	String	文件夹名 注意：此字段可能返回 null，表示取不到有效值。 示例值：folder-xxx
Items	Array of TreeResourceItem	文件列表 注意：此字段可能返回 null，表示取不到有效值。
Children	Array of DescribeTreeResourcesRsp	子目录列表 注意：此字段可能返回 null，表示取不到有效值。
TotalCount	Integer	资源总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：200
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询指定依赖

查询指定依赖

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeTreeResources
<公共请求参数>

{
  "Filters": [
    {
      "Name": "ResourceName",
      "Values": [
        "normal"
      ]
    }
  ],
  "WorkspaceId": "space-53rqk422"
}
```

输出示例

```
{
  "Response": {
    "Children": null,
    "Id": "root",
    "Items": [
      {
        "FileName": "1317959670/deep_normal_flink-hello-world-1.0.0.jar",
        "FolderId": "root",
        "Name": "1317959670/deep_normal_flink-hello-world-1.0.0.jar",
        "RefJobStatusCountSet": [],
        "Remark": "deep normal size",
        "ResourceId": "resource-er9pkk0o",
        "ResourceType": 1
      },
      {
        "FileName": "normal_flink-hello-world-1.0.0.jar",
        "FolderId": "root",
        "Name": "normal_flink-hello-world-1.0.0.jar",
        "RefJobStatusCountSet": [],
        "Remark": "normal size",
        "ResourceId": "resource-jvwtg594",
        "ResourceType": 1
      }
    ],
    "Name": "依赖目录",
    "ParentId": "",
    "RequestId": "78d06002-44d2-46fa-8f63-14f51b0237d3",
    "TotalCount": 2
  }
}
```

```
}  
}
```

示例2 分类中的自定义树状结构并筛选

用户点击作业管理页面中的分类的自定义树状结构并点击分类旁的筛选按钮选择作业类型

输入示例

```
POST / HTTP/1.1  
Host: oceanus.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: DescribeTreeResources  
<公共请求参数>
```

```
{  
  "Filters": [  
    {  
      "Name": "ResourceName",  
      "Values": [  
        "111"  
      ]  
    }  
  ]  
}
```

输出示例

```
{  
  "Response": {  
    "Children": [  
      {  
        "ParentId": "root",  
        "Id": "folder-xxxxx",  
        "Name": "folder-name",  
        "Items": null,  
        "Children": null,  
        "TotalCount": 0  
      }  
    ],  
    "Id": "root",  
    "Items": [  
      {  
        "ResourceId": "resource-xxxxxxx",  
        "Name": "flink-connector-jdbc.jar",  
        "ResourceType": 1,  
        "Remark": "pgsql1",  
        "FileName": "flink-connector-jdbc.jar",  
        "FolderId": "root",  
        "RefJobStatusCountSet": [  
          {  
            "JobStatus": 5,  
            "Count": 2  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
"Name": "依赖目录",
"ParentId": "",
"RequestId": "4aafc302-4fc5-44f5-9e90-2e0a2702ab81",
"TotalCount": 333
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	未授权操作。
FailedOperation	操作失败。
FailedOperation.DataSourceConnectionFailed	数据库连接失败，请检查参数是否填写正确。
InternalError	内部错误。
InvalidParameter	参数错误。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。

查询文件夹

最近更新时间：2025-04-25 01:48:26

1. 接口描述

接口请求域名：oceanus.tencentcloudapi.com。

查询指定文件夹及其相应的子文件夹信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeFolder。
Version	是	String	公共参数 ，本接口取值：2019-04-22。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FolderId	是	String	folder id 示例值：folder-lfqkt10s
WorkSpaceId	是	String	workspace id 示例值：space-53rqk422
FolderType	是	Integer	1:资源文件夹 其他:作业文件夹 示例值：1

3. 输出参数

参数名称	类型	描述
FolderId	String	folder id 示例值：folder-lfqkt10s
FolderName	String	folder name 示例值：test
ParentId	String	父文件夹id 示例值：folder-lfqkt10i
FolderType	Integer	文件夹类型 示例值：1
WorkSpaceId	String	workspace id
SubFolderInfo	Array of SubFolderInfo	子文件夹信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询文件夹

查询文件夹

输入示例

```
POST / HTTP/1.1
Host: oceanus.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeFolder
<公共请求参数>

{
  "FolderId": "folder-bnjsm2sb",
  "WorkspaceId": "space-7rfcns1f",
  "FolderType": 1
}
```

输出示例

```
{
  "Response": {
    "FolderId": "folder-bnjsm2sb",
    "FolderName": "6.9_connector",
    "FolderType": 1,
    "ParentId": "root",
    "RequestId": "3ca249e8-b742-4e97-856d-1fca5a95d6a5",
    "SubFolderInfo": [
      {
        "FolderId": "folder-gbtcpgh0",
        "FolderName": "hive_catalog_config_311"
      },
      {
        "FolderId": "folder-79b2dxuc",
        "FolderName": "sql_server"
      },
      {
        "FolderId": "folder-oc406jen",
        "FolderName": "doris"
      },
      {
        "FolderId": "folder-3ldo2w3g",
        "FolderName": "mysql_cdc"
      },
      {
        "FolderId": "folder-nodi86tq",
        "FolderName": "kudu"
      }
    ],
    "WorkspaceId": "space-7rfcns1f"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

数据结构

最近更新时间：2025-05-28 01:23:45

CCN

云联网描述信息

被如下接口引用：DescribeClusters。

名称	类型	描述
VpcId	String	私有网络 ID
SubnetId	String	子网 ID
CcnId	String	云联网 ID，如 ccn-rahigzjd

ClazzLevel

```
{
  "Clazz": "c1", // java类全路径
  "Level": "WARN" // 日志级别 TRACE, DEBUG、INFO、WARN、ERROR
}
```

被如下接口引用：CreateJobConfig, DescribeJobConfigs。

名称	类型	必选	描述
Clazz	String	是	java类全路径 注意：此字段可能返回 null，表示取不到有效值。 示例值：org.x.x.x
Level	String	是	日志级别 TRACE, DEBUG、INFO、WARN、ERROR 注意：此字段可能返回 null，表示取不到有效值。 示例值：INFO

Cluster

描述用户创建的集群信息

被如下接口引用：DescribeClusters。

名称	类型	描述
ClusterId	String	集群 ID 示例值：cluster-xxxx
Name	String	集群名称 示例值：cluster
Region	String	地域 示例值：ap-guangzhou
AppId	Integer	用户 AppID 示例值：12314214
OwnerUin	String	主账号 UIN 示例值：12314214
CreatorUin	String	创建者 UIN 示例值：12314214
Status	Integer	集群状态, 1 未初始化, 3 初始化中, 2 运行中 示例值：1
Remark	String	描述 示例值：test

名称	类型	描述
CreateTime	String	集群创建时间 示例值：2020-01-21 00:00:10
UpdateTime	String	最后一次操作集群的时间 示例值：2020-01-21 01:00:10
CuNum	Integer	CU 数量 示例值：12
CuMem	Integer	CU 内存规格 示例值：4
Zone	String	可用区 示例值：ap-guangzhou-1
StatusDesc	String	状态描述 示例值：1
CCNs	Array of CCN	网络 示例值：ccn-a
NetEnvironmentType	Integer	网络 示例值：2
FreeCuNum	Integer	空闲 CU 示例值：2
Tags	Array of Tag	集群绑定的标签 注意：此字段可能返回 null，表示取不到有效值。 示例值：tag
IsolatedTime	String	集群隔离时间; 没隔离时间，则为 - 注意：此字段可能返回 null，表示取不到有效值。 示例值：2024-12-12 10:00:00
ExpireTime	String	集群过期时间; 没过期概念，则为 - 注意：此字段可能返回 null，表示取不到有效值。 示例值：2024-12-12 10:00:00
SecondsUntilExpiry	String	距离过期还有多少秒; 没过期概念，则为 - 注意：此字段可能返回 null，表示取不到有效值。 示例值：2024-12-12 10:00:00
AutoRenewFlag	Integer	自动续费标记，0 表示默认状态 (用户未设置，即初始状态，用户开通了预付费不停服特权会进行自动续费)，1 表示自动续费，2表示明确不自动续费(用户设置) 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
DefaultCOSBucket	String	集群的默认 COS 存储桶 注意：此字段可能返回 null，表示取不到有效值。 示例值：test-12314214
CLSLogSet	String	集群的CLS 日志集 LogSet 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxxx-xx-xx-xxxx
CLSTopicId	String	集群的CLS 日志主题 TopicId 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxxx-xx-xx-xxxx
CLSLogName	String	集群的CLS 日志集 名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：日志集名字
CLSTopicName	String	集群的CLS 日志主题 名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：日志主题名字
Version	ClusterVersion	集群的版本信息 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：2019
FreeCu	Float	细粒度资源下的空闲CU 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.5
DefaultLogCollectConf	String	集群的默认日志采集配置 注意：此字段可能返回 null，表示取不到有效值。 示例值：log
CustomizedDNSEnabled	Integer	取值：0-没有设置，1-已设置，2-不允许设置 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Correlations	Array of WorkSpaceClusterItem	空间信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：space-aa
RunningCu	Float	运行CU 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.5
PayMode	Integer	0 后付费,1 预付费 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
IsNeedManageNode	Integer	前端区分 集群是否需要2CU逻辑 因为历史集群 变配不需要, default 1 新集群都需要 注意：此字段可能返回 null，表示取不到有效值。 示例值：0 / 1
ClusterSessions	Array of ClusterSession	session集群信息 注意：此字段可能返回 null，表示取不到有效值。
ArchGeneration	Integer	V3版本 = 2 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ClusterType	Integer	0:TKE, 1:EKS 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Orders	Array of Order	订单信息 注意：此字段可能返回 null，表示取不到有效值。
SqlGateways	Array of SqlGatewayItem	Gateway信息 注意：此字段可能返回 null，表示取不到有效值。
WebUIType	Integer	0 公网访问 // 1 内网访问 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Type	Integer	2 独享集群 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
SubEks	SubEks	子eks集群 注意：此字段可能返回 null，表示取不到有效值。
AgentSerialId	String	上级集群 注意：此字段可能返回 null，表示取不到有效值。
ResourceType	Integer	资源类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
BillingResourceMode	String	集群类型 注意：此字段可能返回 null，表示取不到有效值。 示例值："1"

名称	类型	描述
MemRatio	Integer	Cu比例 注意：此字段可能返回 null，表示取不到有效值。 示例值：4
CrossTenantEniMode	Integer	是否开启跨租户弹性网卡 示例值：0
TotalCpu	Float	总的CPU 示例值：0
TotalMem	Float	总的内存 示例值：0
RunningCpu	Float	运行的CPU 示例值：0
RunningMem	Float	运行的内存 示例值：0
Setats	Setats	setats集群 注意：此字段可能返回 null，表示取不到有效值。

ClusterGroupSetItem

工作空间集群组信息

被如下接口引用：DescribeWorkSpaces。

名称	类型	描述
ClusterId	String	clusterGroup 的 SerialId 示例值：cluster-sdhd0vtz
Name	String	集群名称 示例值：MyClusterName
Region	String	地域 示例值：ap-beijing
Zone	String	区 示例值：ap-beijing-2
AppId	Integer	账号 APPID 示例值：3435324
OwnerUin	String	主账号 UIN 示例值：123456
CreatorUin	String	创建账号 UIN 示例值：123456
CuNum	Integer	CU 数量 示例值：2
CuMem	Integer	CU 内存规格 示例值：2
Status	Integer	集群状态, 1 未初始化,, 3 初始化中, 2 运行中 示例值：2
StatusDesc	String	状态描述 示例值：description
CreateTime	String	集群创建时间 示例值：xxxx-xx-xx-xxxx
UpdateTime	String	最后一次操作集群的时间 示例值：xxxx-xx-xx-xxxx

名称	类型	描述
Remark	String	描述 示例值：remark
NetEnvironmentType	Integer	网络 示例值：2
FreeCuNum	Integer	空闲 CU 示例值：1
FreeCu	Float	细粒度资源下的空闲CU 示例值：2.5
RunningCu	Float	运行中CU 示例值：2.5
PayMode	Integer	付费模式 示例值：0
SubEks	SubEks	弹性 注意：此字段可能返回 null，表示取不到有效值。 示例值：{}
BillingResourceMode	String	默认 "" 包销模式 "exclusiveSale" 注意：此字段可能返回 null，表示取不到有效值。
TotalCpu	Float	TotalCpu 示例值：0
TotalMem	Float	TotalMem 示例值：0
RunningCpu	Float	RunningCpu 示例值：0
RunningMem	Float	RunningMem 示例值：0

ClusterSession

session集群信息

被如下接口引用：DescribeClusters。

名称	类型	描述
ClusterGroupSerialId	String	集群SerialId 示例值：cluster-nbcjavkn
Appld	Integer	创建者appld 示例值：1257058945
OwnerUin	String	创建者主账号 示例值：100006386216
CreatorUin	String	创建者账号 示例值：100006386216
Region	String	区域 示例值：ap-guangzhou
Zone	String	zone 示例值：ap-guangzhou-3
Status	Integer	Session集群状态 示例值：1
CuNum	Float	Session集群消耗的cu数量 示例值：2

名称	类型	描述
FlinkVersion	String	Session集群的Flink版本 示例值: Flink-1.13
WebUIUrl	String	session集群FlinkUI地址
Properties	Array of Property	session集群高级参数 注意: 此字段可能返回 null, 表示取不到有效值。
ResourceRefs	Array of SessionClusterRefItem	引用资源 注意: 此字段可能返回 null, 表示取不到有效值。
JobManagerCuSpec	Float	JobManager的规格 示例值: 1
TaskManagerCuSpec	Float	TaskManager的规格 示例值: 0.5
TaskManagerNum	Integer	TaskManager启动的数量 示例值: 2
CreateTime	String	创建时间
UpdateTime	String	更新时间
JobManagerCpu	Float	JobManagerCpu 示例值: 0
JobManagerMem	Float	JobManagerMem 示例值: 0
TaskManagerCpu	Float	TaskManagerCpu 示例值: 0
TaskManagerMem	Float	TaskManagerMem 示例值: 0

ClusterVersion

集群的版本相关信息

被如下接口引用: DescribeClusters。

名称	类型	描述
Flink	String	集群的Flink版本 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: Flink-1.11
SupportedFlink	Array of String	集群支持的Flink版本 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["Flink-1.13", "Flink-1.11"]

CopyJobItem

复制作业单条明细

被如下接口引用: CopyJobs。

名称	类型	必选	描述
SourceId	String	是	需要复制的作业serial id 示例值: cql-pqx9zh30
TargetClusterId	String	是	目标集群的cluster serial id 示例值: cluster-l0elpxde
SourceName	String	否	需要复制的作业名称 示例值: job1

名称	类型	必选	描述
TargetName	String	否	新作业的名称 示例值：job2
TargetFolderId	String	否	新作业的目录id 示例值：folder-2ue9i8ny
JobType	Integer	否	源作业类型 示例值：1

CopyJobResult

复制作业单条明细结果

被如下接口引用：CopyJobs。

名称	类型	描述
JobId	String	原作业id 注意：此字段可能返回 null，表示取不到有效值。 示例值：cql-pqx9zh30
JobName	String	原作业名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：job1
TargetJobName	String	新作业名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：job2
TargetJobId	String	新作业id 注意：此字段可能返回 null，表示取不到有效值。 示例值：cql-pqx9zh30
Message	String	失败时候的信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：失败
Result	Integer	0 成功 -1 失败 注意：此字段可能返回 null，表示取不到有效值。 示例值：结果
ClusterName	String	目标集群名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：集群1
ClusterId	String	目标集群id 注意：此字段可能返回 null，表示取不到有效值。 示例值：cluster-l0elpxde
JobType	Integer	作业类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

DescribeTreeJobsRsp

自定义树结构遍历子节点

被如下接口引用：DescribeTreeJobs。

名称	类型	描述
ParentId	String	父节点ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：root-xxx
Id	String	当前文件夹ID 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：folder-xxx
Name	String	当前文件夹名 注意：此字段可能返回 null，表示取不到有效值。 示例值：New Folder
JobSet	Array of TreeJobSets	当前文件夹下的作业集合 注意：此字段可能返回 null，表示取不到有效值。 示例值：JobSet
Children	Array of DescribeTreeJobsRsp	迭代子目录 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	请求ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx

DescribeTreeResourcesRsp

树状结构资源列表对象

被如下接口引用：DescribeTreeResources。

名称	类型	描述
ParentId	String	父节点ID 示例值：folder-xxxx
Id	String	文件夹ID 示例值：folder-xxxx
Name	String	文件夹名称 示例值：xxxx
Items	Array of TreeResourceItem	文件夹下资源数字 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
Children	Array of DescribeTreeResourcesRsp	子节点 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
TotalCount	Integer	资源总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：200

ExpertModeConfiguration

作业配置 -- 专家模式的详细配置

被如下接口引用：CreateJobConfig, DescribeJobConfigs。

名称	类型	必选	描述
JobGraph	JobGraph	否	Job graph 注意：此字段可能返回 null，表示取不到有效值。
NodeConfig	Array of NodeConfig	否	Node configuration 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
SlotSharingGroups	Array of SlotSharingGroup	否	Slot sharing groups 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]

Filter

查询作业列表时的过滤器

被如下接口引用：DescribeClusters, DescribeJobConfigs, DescribeJobs, DescribeResources, DescribeSystemResources, DescribeTreeJobs, DescribeTreeResources, DescribeWorkSpaces。

名称	类型	必选	描述
Name	String	是	要过滤的字段
Values	Array of String	是	字段的过滤值

GatewayRefItem

Gateway引用资源信息

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
WorkspaceId	String	是	空间唯一标识 注意：此字段可能返回 null，表示取不到有效值。
ResourceId	String	是	资源唯一标识 注意：此字段可能返回 null，表示取不到有效值。
Version	Integer	是	版本号 注意：此字段可能返回 null，表示取不到有效值。
Type	Integer	是	引用类型，0:用户资源 注意：此字段可能返回 null，表示取不到有效值。

JobConfig

作业配置详情

被如下接口引用：DescribeJobConfigs。

名称	类型	描述
JobId	String	作业Id 示例值：cql-xxx
EntrypointClass	String	主类 注意：此字段可能返回 null，表示取不到有效值。 示例值：hello
ProgramArgs	String	主类入参 注意：此字段可能返回 null，表示取不到有效值。 示例值：--config key=value
Remark	String	备注 注意：此字段可能返回 null，表示取不到有效值。 示例值：hello
CreateTime	String	作业配置创建时间 示例值：2021-01-22 20:06:19
Version	Integer	作业配置的版本号 示例值：1
DefaultParallelism	Integer	作业默认并行度 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Properties	Array of Property	系统参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
ResourceRefDetails	Array of ResourceRefDetail	引用资源 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：[]
CreatorUin	String	创建者uin 注意：此字段可能返回 null，表示取不到有效值。 示例值：1132
UpdateTime	String	作业配置上次启动时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-01-22 20:06:19
COSBucket	String	作业绑定的存储桶 注意：此字段可能返回 null，表示取不到有效值。 示例值：oceanus-example
LogCollect	Integer	是否启用日志收集，0-未启用，1-已启用，2-历史集群未设置日志集，3-历史集群已开启 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
MaxParallelism	Integer	作业的最大并行度 注意：此字段可能返回 null，表示取不到有效值。 示例值：2048
JobManagerSpec	Float	JobManager规格 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.5
TaskManagerSpec	Float	TaskManager规格 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.5
ClsLogsetId	String	CLS日志集ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx
ClsTopicId	String	CLS日志主题ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx
PythonVersion	String	pyflink作业运行的python版本 注意：此字段可能返回 null，表示取不到有效值。 示例值：Python-3.7
AutoRecover	Integer	Oceanus 平台恢复作业开关 1:开启 -1: 关闭 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
LogLevel	String	日志级别 注意：此字段可能返回 null，表示取不到有效值。 示例值：INFO
ClazzLevels	Array of ClazzLevel	类日志级别 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
ExpertModeOn	Boolean	是否开启专家模式 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
ExpertModeConfiguration	ExpertModeConfiguration	专家模式的配置 注意：此字段可能返回 null，表示取不到有效值。 示例值：{}
TraceModeOn	Boolean	trace链路 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
TraceModeConfiguration	TraceModeConfiguration	trace链路配置 注意：此字段可能返回 null，表示取不到有效值。 示例值：{}

名称	类型	描述
CheckpointRetainedNum	Integer	checkpoint保留个数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
JobGraph	JobGraph	算子拓扑图 注意：此字段可能返回 null，表示取不到有效值。 示例值：{}
EsServerlessIndex	String	es索引 注意：此字段可能返回 null，表示取不到有效值。 示例值：index-**
EsServerlessSpace	String	es空间 注意：此字段可能返回 null，表示取不到有效值。 示例值：space-**
IndexName	String	es索引中文 注意：此字段可能返回 null，表示取不到有效值。 示例值：index-**
WorkspaceName	String	es空间中文 注意：此字段可能返回 null，表示取不到有效值。 示例值：space-**
FlinkVersion	String	flink 版本 注意：此字段可能返回 null，表示取不到有效值。 示例值：Flink-1.13
JobManagerCpu	Float	jm使用cpu数目 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
JobManagerMem	Float	jm使用内存数目 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
TaskManagerCpu	Float	tm使用cpu数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TaskManagerMem	Float	tm使用mem数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
JobConfigItem	JobConfig	运行中配置 注意：此字段可能返回 null，表示取不到有效值。

JobEvent

描述作业发生的一个事件

被如下接口引用：DescribeJobEvents。

名称	类型	描述
Type	String	内部定义的事件类型 示例值：11
Description	String	事件类型的说明文字 示例值：作业停止
Timestamp	Integer	事件发生的 Unix 时间戳（秒） 示例值：1630932161
RunningOrderId	Integer	事件发生时的运行 ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

名称	类型	描述
Message	String	事件的一些可选说明 注意：此字段可能返回 null，表示取不到有效值。 示例值：作业异常停止，原因是 TaskManager 重启次数达到阈值
SolutionLink	String	异常事件的排查手册链接 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://www.example.org/link?id=12357

JobEventInfo

事件信息

被如下接口引用：DescribeJobs。

名称	类型	描述
ErrorEventTotal	Integer	异常事件总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

JobGraph

作业运行图

被如下接口引用：CreateJobConfig, DescribeJobConfigs。

名称	类型	必选	描述
Nodes	Array of JobGraphNode	否	运行图的点集合 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
Edges	Array of JobGraphEdge	否	运行图的边集合 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]

JobGraphEdge

Flink Job 运行图的边信息

被如下接口引用：CreateJobConfig。

名称	类型	必选	描述
Source	Integer	是	边的起始节点ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Target	Integer	是	边的目标节点ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：2

JobGraphNode

Flink Job 运行图的点信息

被如下接口引用：CreateJobConfig。

名称	类型	必选	描述
Id	Integer	是	节点ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

名称	类型	必选	描述
Description	String	是	节点描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：This is a node description
Name	String	是	节点名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：Node1
Parallelism	Integer	是	节点并行度 注意：此字段可能返回 null，表示取不到有效值。 示例值：2

JobInstanceForSubmissionLog

搜索启动日志时返回的作业实例

被如下接口引用：DescribeJobSubmissionLog。

名称	类型	描述
RunningOrderId	Integer	实例的Id, 按照启动的时间顺序, 从1开始
JobInstanceStartTime	String	作业实例的启动时间
StartingMillis	Integer	作业实例启动的时间（毫秒）

JobRuntimeInfo

作业运行时信息

被如下接口引用：DescribeJobRuntimeInfo。

名称	类型	描述
Key	String	运行信息的key，目前支持：TaskManagers：taskmanager pod 列表；StreamGraph：作业对应的 StreamGraph；SubTasks：作业的 subtask 列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：TaskManagers
Value	String	运行信息 注意：此字段可能返回 null，表示取不到有效值。

JobV1

Job详细信息

被如下接口引用：DescribeJobs。

名称	类型	描述
JobId	String	作业ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：cql-xxx
Region	String	地域 注意：此字段可能返回 null，表示取不到有效值。 示例值：ap-guangzhou
Zone	String	可用区 注意：此字段可能返回 null，表示取不到有效值。 示例值：ap-guangzhou-1
AppId	Integer	用户AppId 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

名称	类型	描述
OwnerUin	String	用户UIN 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
CreatorUin	String	创建者UIN 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
Name	String	作业名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：test
JobType	Integer	作业类型，1：sql作业，2：Jar作业 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Status	Integer	作业状态，1：未初始化，2：未发布，3：操作中，4：运行中，5：停止，6：暂停，-1：故障 注意：此字段可能返回 null，表示取不到有效值。 示例值：4
CreateTime	String	作业创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-06-11 16:44:44
StartTime	String	作业启动时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-06-11 16:44:44
StopTime	String	作业停止时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-06-11 16:44:44
UpdateTime	String	作业更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-06-11 16:44:44
TotalRunMillis	Integer	作业累计运行时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：13864986
Remark	String	备注信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：test
LastOpResult	String	操作错误提示信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：success
ClusterName	String	集群名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：test
LatestJobConfigVersion	Integer	最新配置版本号 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
PublishedJobConfigVersion	Integer	已发布的配置版本 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
RunningCuNum	Integer	运行的CU数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
CuMem	Integer	作业内存规格 注意：此字段可能返回 null，表示取不到有效值。 示例值：4

名称	类型	描述
StatusDesc	String	作业状态描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：running
CurrentRunMillis	Integer	运行状态时表示单次运行时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：124
ClusterId	String	作业所在的集群ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：cluster-xxx
WebUIUrl	String	作业管理WEB UI 入口 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://175.24.255.193//cql-czvy480r-78/?defaultToken=0
SchedulerType	Integer	作业所在集群类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
ClusterStatus	Integer	作业所在集群状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
RunningCu	Float	细粒度下的运行的CU数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.5
FlinkVersion	String	作业运行的 Flink 版本 注意：此字段可能返回 null，表示取不到有效值。 示例值：Flink-1.11
WorkSpaceId	String	工作空间 SerialId 注意：此字段可能返回 null，表示取不到有效值。 示例值：space-xxx
WorkSpaceName	String	工作空间名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：mysqlItemSpace
Tags	Array of Tag	作业标签 注意：此字段可能返回 null，表示取不到有效值。
EventInfo	JobEventInfo	作业异常事件信息 注意：此字段可能返回 null，表示取不到有效值。
Description	String	描述信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：desc
ScalingType	Integer	0:代表没开启调优任务，1:开启智能调优，2:代表定时调优 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RunningCpu	Float	使用CPU数目 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RunningMem	Float	使用内存数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
OpenJobDefaultAlarm	Integer	是否开了默认告警 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ProgressDesc	String	操作中描述 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：操作中

LogContent

日志查询的每行日志信息

被如下接口引用：DescribeJobSubmissionLog。

名称	类型	描述
Log	String	日志内容 注意：此字段可能返回 null，表示取不到有效值。 示例值：abc
Time	Integer	毫秒级时间戳 注意：此字段可能返回 null，表示取不到有效值。 示例值：1611816200076
PkgId	String	日志组Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：2014BF16012637D-6
PkgLogId	Integer	日志Id，在日志组范围里唯一 示例值：131072
ContainerName	String	日志所属的容器名 注意：此字段可能返回 null，表示取不到有效值。 示例值：flink-job-manager

LogicalType

SqlGateway返回LogicalType类型

被如下接口引用：FetchSqlGatewayStatementResult。

名称	类型	描述
Type	String	类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：VARCHAR
NullAble	Boolean	是否允许为空 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
Length	Integer	长度 注意：此字段可能返回 null，表示取不到有效值。 示例值：2147483647

NodeConfig

专家模式 计算节点的配置信息

被如下接口引用：CreateJobConfig。

名称	类型	必选	描述
Id	Integer	是	Node ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Parallelism	Integer	否	Node parallelism 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
SlotSharingGroup	String	否	Slot sharing group 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	必选	描述
			示例值：group1
Configuration	Array of Property	否	Configuration properties 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
StateTTL	String	否	节点的状态ttl配置, 多个用；分割 注意：此字段可能返回 null，表示取不到有效值。 示例值：""

Order

集群购买、扩缩容、续费订单信息

被如下接口引用：DescribeClusters。

名称	类型	描述
Type	Integer	创建、续费、扩缩容 1 2 3 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
AutoRenewFlag	Integer	自动续费 1 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
OperateUin	String	操作人的UIN 注意：此字段可能返回 null，表示取不到有效值。 示例值：1201111
ComputeCu	Integer	最终集群的CU数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：12
OrderTime	String	订单的时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2020-12-08 14:12:40

Property

系统配置属性

被如下接口引用：CreateJobConfig, DescribeClusters, DescribeJobConfigs。

名称	类型	必选	描述
Key	String	是	系统配置的Key
Value	String	是	系统配置的Value

RefJobStatusCountItem

依赖作业分状态计数信息

被如下接口引用：DescribeResourceConfigs, DescribeResources, DescribeTreeResources。

名称	类型	描述
JobStatus	Integer	作业状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Count	Integer	作业数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

ResourceConfigItem

描述资源配置的返回参数

被如下接口引用：DescribeResourceConfigs。

名称	类型	描述
ResourceId	String	资源ID
ResourceType	Integer	资源类型
Region	String	资源所属地域
AppId	Integer	资源所属AppId
OwnerUin	String	主账号Uin
CreatorUin	String	子账号Uin
ResourceLoc	ResourceLoc	资源位置描述
CreateTime	String	资源创建时间
Version	Integer	资源版本
Remark	String	资源描述
Status	Integer	资源状态：0: 资源同步中，1:资源已就绪 注意：此字段可能返回 null，表示取不到有效值。
RefJobCount	Integer	关联作业个数 注意：此字段可能返回 null，表示取不到有效值。
RefJobStatusCountSet	Array of RefJobStatusCountItem	分状态统计关联作业数 注意：此字段可能返回 null，表示取不到有效值。

ResourceItem

资源详细描述

被如下接口引用：DescribeResources。

名称	类型	描述
ResourceId	String	资源ID 示例值："test"
Name	String	资源名称 示例值："test"
ResourceType	Integer	资源类型 示例值：0
ResourceLoc	ResourceLoc	资源位置 示例值：0
Region	String	资源地域 示例值："test"
AppId	Integer	应用ID 示例值：0
OwnerUin	String	主账号Uin 示例值："test"
CreatorUin	String	子账号Uin 示例值："test"
CreateTime	String	资源创建时间 示例值："test"

名称	类型	描述
UpdateTime	String	资源最后更新时间 示例值: "test"
LatestResourceConfigVersion	Integer	资源的资源版本ID 示例值: 0
Remark	String	资源备注 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "test"
VersionCount	Integer	版本个数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
RefJobCount	Integer	关联作业数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
IsJobRun	Integer	作业运行状态 示例值: 0
FileName	String	文件名 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "test"
WorkSpaceId	Integer	工作空间ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
RefJobStatusCountSet	Array of RefJobStatusCountItem	分状态统计关联作业数 注意: 此字段可能返回 null, 表示取不到有效值。
Connector	String	连接器名称 示例值: mysql
ConnectorVersion	String	连接器版本 示例值: 1.1
ConnectionMethod	String	连接方式 示例值: 1
RelatedResourceId	String	connector关联的资源id 示例值: "test"
Icon	String	图标 示例值: "test"
ConnectorName	String	连接器中文名 示例值: "Kafka"
ConnectorUrl	String	连接器官网链接 示例值: "test"

ResourceLoc

资源位置描述

被如下接口引用: CreateResource, CreateResourceConfig, DescribeResourceConfigs, DescribeResources。

名称	类型	必选	描述
StorageType	Integer	是	资源位置的存储类型, 目前只支持1: COS 示例值: 1
Param	ResourceLocParam	是	描述资源位置的json 示例值: {"Bucket": "bucket1", "Path": "your/path", "Region": "ap-guangzhou"}

ResourceLocParam

资源参数描述

被如下接口引用：CreateResource, CreateResourceConfig。

名称	类型	必选	描述
Bucket	String	是	资源bucket 示例值：scs-online-1257058945
Path	String	是	资源路径 示例值：251008563/100000006047/flink-cos-fs-hadoop-oceanus-1-20210304112
Region	String	否	资源所在地域，如果不填，则使用Resource的Region 注意：此字段可能返回 null，表示取不到有效值。 示例值：ap-chengdu

ResourceRef

资源引用参数

被如下接口引用：CreateJobConfig。

名称	类型	必选	描述
ResourceId	String	是	资源ID
Version	Integer	是	资源版本ID，-1表示使用最新版本
Type	Integer	是	引用资源类型，例如主资源设置为1，代表main class所在的jar包

ResourceRefDetail

JobConfig引用资源信息

被如下接口引用：DescribeJobConfigs。

名称	类型	描述
ResourceId	String	资源id
Version	Integer	资源版本，-1表示使用最新版本
Name	String	资源名称
Type	Integer	1: 主资源
SystemProvide	Integer	1: 系统内置资源
Connector	String	Connector 示例值：kafka

ResourceRefJobInfo

资源被Job 引用信息

被如下接口引用：DescribeResourceRelatedJobs。

名称	类型	描述
JobId	String	Job id
JobConfigVersion	Integer	Job配置版本
ResourceVersion	Integer	资源版本

ResourceRefLatest

资源引用

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
ResourceId	String	否	资源id 注意：此字段可能返回 null，表示取不到有效值。 示例值：resource-123
Version	Integer	否	版本号 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Type	Integer	否	资源类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Status	Integer	否	状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
WorkspaceId	String	否	空间id 注意：此字段可能返回 null，表示取不到有效值。 示例值：space-g7pfiemf
Name	String	否	资源名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：c.conf

ResultColumn

Sql Gateway返回Column类型

被如下接口引用：FetchSqlGatewayStatementResult。

名称	类型	描述
Name	String	名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：result
LogicalType	LogicalType	本地类型描述 注意：此字段可能返回 null，表示取不到有效值。
Comment	String	备注 注意：此字段可能返回 null，表示取不到有效值。

ResultData

Sql Gateway返回数据

被如下接口引用：FetchSqlGatewayStatementResult。

名称	类型	描述
Kind	String	操作类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：INSERT
Fields	Array of String	结果 注意：此字段可能返回 null，表示取不到有效值。 示例值：[OK]

RoleAuth

角色授权信息

被如下接口引用：DescribeWorkSpaces。

名称	类型	描述
AppId	Integer	用户 AppID 示例值：1234567
WorkSpaceSerialId	String	工作空间 SerialId 示例值：1327
OwnerUin	String	主账号 UIN 示例值：34553634
CreatorUin	String	创建者 UIN 示例值：34553634
AuthSubAccountUin	String	绑定授权的 UIN 示例值：34553634
Permission	Integer	对应 role表的id 示例值：2
CreateTime	String	创建时间 示例值：2020-01-21 01:00:10
UpdateTime	String	最后一次操作时间 示例值：2020-01-21 01:00:10
Status	Integer	2 启用 1 停用 示例值：2
Id	Integer	id 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
WorkSpaceId	Integer	工作空间id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RoleName	String	权限名称 注意：此字段可能返回 null，表示取不到有效值。

RunJobDescription

作业启动详情

被如下接口引用：RunJobs。

名称	类型	必选	描述
JobId	String	是	作业Id 示例值：cql-6w8eab6f
RunType	Integer	是	运行类型，1：启动，2：恢复 示例值：1
StartMode	String	否	兼容旧版 SQL 类型作业启动参数：指定数据源消费起始时间点（建议传值） 保证参数为 LATEST、EARLIEST、T+Timestamp（例：T1557394288000） 示例值：LATEST
JobConfigVersion	Integer	否	当前作业的某个版本 （不传默认为非草稿的作业版本） 示例值：1
SavepointPath	String	否	Savepoint路径 示例值：Savepoint
SavepointId	String	否	Savepoint的Id 示例值：cql-6w8eab6f
UseOldSystemConnector	Boolean	否	使用历史版本系统依赖 示例值：false

名称	类型	必选	描述
CustomTimestamp	Integer	否	自定义时间戳 示例值：1690531378626
KafkaScanMode	String	否	timestamp; latest-offset; earliest-offset; 任选一种 示例值：earliest-offset

Savepoint

描述Savepoint信息

被如下接口引用：DescribeJobSavepoint。

名称	类型	描述
Id	Integer	主键 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
VersionId	Integer	版本号 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Status	Integer	状态 1: Active; 2: Expired; 3: InProgress; 4: Failed; 5: Timeout 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
CreateTime	Integer	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1652962710
UpdateTime	Integer	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1652962710
Path	String	路径 注意：此字段可能返回 null，表示取不到有效值。 示例值：cosn://kkingtest-123/123/123/cql-i7985eg9/27/flink-checkpoints/0
Size	Integer	大小 注意：此字段可能返回 null，表示取不到有效值。 示例值：89446
RecordType	Integer	快照类型 1: savepoint; 2: checkpoint; 3: cancelWithSavepoint 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
JobRuntimeId	Integer	运行作业实例的顺序 ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Description	String	描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：Description
Timeout	Integer	固定超时时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
SerialId	String	快照 serialId 注意：此字段可能返回 null，表示取不到有效值。 示例值：cql-6w8eab6f
TimeConsuming	Integer	耗时 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PathStatus	Integer	快照路径状态 1: 可用; 2: 不可用; 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：1

SessionClusterRefItem

session集群引用资源信息

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
WorkspaceId	String	是	空间唯一标识 注意：此字段可能返回 null，表示取不到有效值。
ResourceId	String	是	资源唯一标识 注意：此字段可能返回 null，表示取不到有效值。
Version	Integer	是	版本号 注意：此字段可能返回 null，表示取不到有效值。
Type	Integer	是	引用类型，0:用户资源 注意：此字段可能返回 null，表示取不到有效值。

Setats

setats类型

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
SetatsSerialId	String	否	setats serialId 注意：此字段可能返回 null，表示取不到有效值。 示例值：setats-dx1
Status	Integer	否	1 // 停止 2 // 运行中 3 // 初始化中 4 // 扩容中 5 // Warehouse未配置 6 // Warehouse配置中 7 // 重启中 -2 // 已删除(集群被销毁时更新为此状态) 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
Warehouse	Warehouse	否	setats warehouse 注意：此字段可能返回 null，表示取不到有效值。 示例值：{""}
MasterInfo	SetatsCvmInfo	否	setats master 机器规格 注意：此字段可能返回 null，表示取不到有效值。 示例值：{""}
WorkerInfo	SetatsCvmInfo	否	setats worker规格 注意：此字段可能返回 null，表示取不到有效值。 示例值：{""}
Tags	Array of Tag	否	标签 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{}]
AutoRenewFlag	Integer	否	自动续费 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ExpireTime	String	否	过期时间 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	必选	描述
			示例值：2025-11-11 11:11:11
SecondsUntilExpiry	String	否	过期时间 秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：123
CreateTime	String	否	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-11-11 11:11:11
ManagerUrl	String	否	manager url 注意：此字段可能返回 null，表示取不到有效值。 示例值：127.0.0.1:6072
IsolatedTime	String	否	隔离时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-11-11 11:11:11

SetatsCvmlInfo

setats 机器规格

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
Cpu	Float	否	setats机器cpu 注意：此字段可能返回 null，表示取不到有效值。 示例值：8
Mem	Float	否	setats机器内存 注意：此字段可能返回 null，表示取不到有效值。 示例值：32
DefaultParallelism	Integer	否	setats worker 并行度 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
Disk	SetatsDisk	否	setats 机器磁盘 注意：此字段可能返回 null，表示取不到有效值。

SetatsDisk

setats disk

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
DiskType	String	否	磁盘类型 CLOUD_BSSD CLOUD_SSD CLOUD_HSSD CLOUD_PREMIUM 注意：此字段可能返回 null，表示取不到有效值。 示例值：CLOUD_PREMIUM
DiskSize	Integer	否	磁盘大小 注意：此字段可能返回 null，表示取不到有效值。 示例值：160

SlotSharingGroup

SlotSharingGroup 描述

被如下接口引用：CreateJobConfig。

名称	类型	必选	描述
Name	String	是	SlotSharingGroup的名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：ssgA
Spec	SlotSharingGroupSpec	是	SlotSharingGroup的规格 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]
Description	String	否	SlotSharingGroup的描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：ssgA
Configuration	Array of Property	否	SlotSharingGroup的配置 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]

SlotSharingGroupSpec

SlotSharingGroup的规格描述

被如下接口引用：CreateJobConfig。

名称	类型	必选	描述
CPU	Float	是	适用的cpu 注意：此字段可能返回 null，表示取不到有效值。 示例值：1.0
HeapMemory	String	是	默认为b, 支持单位有 b, kb, mb, gb 注意：此字段可能返回 null，表示取不到有效值。 示例值：1gb
OffHeapMemory	String	否	默认为b, 支持单位有 b, kb, mb, gb 注意：此字段可能返回 null，表示取不到有效值。 示例值：1gb
ManagedMemory	String	否	默认为b, 支持单位有 b, kb, mb, gb 注意：此字段可能返回 null，表示取不到有效值。 示例值：1gb

SqlGatewayItem

SqlGateway配置信息

被如下接口引用：DescribeClusters。

名称	类型	描述
SerialId	String	唯一标识 注意：此字段可能返回 null，表示取不到有效值。
FlinkVersion	String	Flink内核版本 注意：此字段可能返回 null，表示取不到有效值。
Status	Integer	状态，1.停止 2. 开启中 3. 开启 4. 开启失败 5. 停止中 注意：此字段可能返回 null，表示取不到有效值。
CreatorUin	String	创建人 注意：此字段可能返回 null，表示取不到有效值。
ResourceRefs	Array of GatewayRefItem	引用资源 注意：此字段可能返回 null，表示取不到有效值。
CuSpec	Float	Cu规格 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	创建时间

名称	类型	描述
		注意：此字段可能返回 null，表示取不到有效值。
UpdateTime	String	更新时间 注意：此字段可能返回 null，表示取不到有效值。
Properties	Array of Property	配置参数 注意：此字段可能返回 null，表示取不到有效值。
Cpu	Float	Cpu 示例值：0
Mem	Float	Mem 示例值：0

StatementResult

Sql Gateway 返回Result结构类型

被如下接口引用：FetchSqlGatewayStatementResult。

名称	类型	描述
Columns	Array of ResultColumn	返回结果列 注意：此字段可能返回 null，表示取不到有效值。
RowFormat	String	格式 注意：此字段可能返回 null，表示取不到有效值。 示例值：JSON
Data	Array of ResultData	结果值 注意：此字段可能返回 null，表示取不到有效值。

StopJobDescription

停止作业的描述信息

被如下接口引用：StopJobs。

名称	类型	必选	描述
JobId	String	是	作业Id 示例值：1
StopType	Integer	是	停止类型，1 停止 2 暂停 示例值：1

SubEks

混合计费

被如下接口引用：DescribeClusters, DescribeWorkSpaces。

名称	类型	必选	描述
SerialId	String	否	集群id 注意：此字段可能返回 null，表示取不到有效值。 示例值：cluster-xxx
CuNum	Integer	否	cu数 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
Status	Integer	否	状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：2

名称	类型	必选	描述
StatusDesc	String	否	状态描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：running
RunningCu	Float	否	运行cu 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.5
TotalCpu	Float	否	总的CPU 示例值：0
TotalMem	Float	否	总的内存 示例值：0
RunningCpu	Float	否	运行的CPU 示例值：0
RunningMem	Float	否	运行的内存 示例值：0

SubFolderInfo

子目录信息

被如下接口引用：DescribeFolder。

名称	类型	描述
FolderId	String	folder id 示例值：folder-lfqkt10s
FolderName	String	folder name 示例值：test

SystemResourceItem

系统资源返回值

被如下接口引用：DescribeSystemResources。

名称	类型	描述
ResourceId	String	资源ID 示例值：resource-9ds0xim0
Name	String	资源名称 示例值：SampleProgram
ResourceType	Integer	资源类型。1 表示 JAR 包，目前只支持该值。 示例值：1
Remark	String	资源备注 示例值：我的测试程序包
Region	String	资源所属地域 示例值：ap-guangzhou
LatestResourceConfigVersion	Integer	资源的最新版本 示例值：2
SystemProvide	Integer	1 是系统提供资源 2 用户提供CONNECTOR 示例值：1

Tag

标签

被如下接口引用：CreateJob, DescribeClusters, DescribeJobs。

名称	类型	必选	描述
TagKey	String	否	标签键 注意：此字段可能返回 null，表示取不到有效值。
TagValue	String	否	标签值 注意：此字段可能返回 null，表示取不到有效值。

TraceModeConfiguration

```
{  
"Rate": "0.01", ///如1%转换为0.01  
"Operator": "1:OUT,2:IN_AND_OUT,3:IN" ///如1%转换为0.01  
}
```

Operator

算子ID顺序配置，可以对每个算子配置IN、OUT、IN_AND_OUT三个值，分别表示采集输入数据、采集输出数据、同时采集输入和输出数据，配置示例：

被如下接口引用：CreateJobConfig, DescribeJobConfigs。

名称	类型	必选	描述
Rate	String	否	如1%转换为0.01 注意：此字段可能返回 null，表示取不到有效值。 示例值：0.01
Operator	String	否	按照算子ID顺序配置，可以对每个算子配置IN、OUT、IN_AND_OUT三个值，分别表示采集输入数据、采集输出数据、同时采集输入和输出数据 注意：此字段可能返回 null，表示取不到有效值。 示例值：1:OUT,2:IN_AND_OUT,3:IN

TreeJobSets

自定义树结构出参作业列表

被如下接口引用：DescribeTreeJobs。

名称	类型	描述
JobId	String	作业Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：cql-xxx
Name	String	作业名 注意：此字段可能返回 null，表示取不到有效值。 示例值：sql1
JobType	Integer	作业类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RunningCu	Float	作业占用资源 注意：此字段可能返回 null，表示取不到有效值。 示例值：2.00
Status	Integer	作业状态 启动或者停止或者暂停 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ScalingType	Integer	0:代表没开启调优任务，1:开启智能调优，2:代表定时调优 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RunningCpu	Float	RunningCpu 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

名称	类型	描述
RunningMem	Float	RunningMem 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
DecodeSqlCode	String	sql 注意：此字段可能返回 null，表示取不到有效值。 示例值：select * from table

TreeResourceItem

树状结构资源对象

被如下接口引用：DescribeTreeResources。

名称	类型	描述
ResourceId	String	资源ID 示例值：resource-××
Name	String	资源名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：××
ResourceType	Integer	资源类型 示例值：1
Remark	String	备注 注意：此字段可能返回 null，表示取不到有效值。 示例值：××
FileName	String	文件名 注意：此字段可能返回 null，表示取不到有效值。 示例值：a.txt
FolderId	String	目录ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：folder-xxxx
RefJobStatusCountSet	Array of RefJobStatusCountItem	分状态统计关联作业数 注意：此字段可能返回 null，表示取不到有效值。

Warehouse

Setats Warehouse结构

被如下接口引用：DescribeClusters。

名称	类型	必选	描述
Status	Integer	否	状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
Location	String	否	location 注意：此字段可能返回 null，表示取不到有效值。 示例值：cos
CatalogType	String	否	catalogtype 注意：此字段可能返回 null，表示取不到有效值。 示例值：hadoop
Uri	String	否	uri 注意：此字段可能返回 null，表示取不到有效值。 示例值：uri
WarehouseUrl	String	否	warehouse url 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	必选	描述
			示例值：hdfs://HDFS1018835/usr/setats
Authentication	String	否	认证方式 注意：此字段可能返回 null，表示取不到有效值。 示例值：kerberos
ResourceRefs	Array of ResourceRefLatest	否	资源 注意：此字段可能返回 null，表示取不到有效值。
HiveUri	String	否	hive warehouse uri 注意：此字段可能返回 null，表示取不到有效值。 示例值：thrift://172.28.22.40:7004
Properties	Array of Property	否	高级参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：{""}

WorkspaceClusterItem

空间和集群绑定关系

被如下接口引用：DescribeClusters。

名称	类型	描述
ClusterGroupId	Integer	集群 ID 示例值：1234
ClusterGroupSerialId	String	集群 SerialId 示例值：cluster-xxxx
ClusterName	String	集群名称 示例值：MyCluster
WorkSpaceId	String	工作空间 SerialId 示例值：spc-xxx
WorkSpaceName	String	工作空间名称 示例值：Myspace
Status	Integer	绑定状态 2 绑定 1 解除绑定 示例值：1
ProjectId	Integer	项目ID 示例值：1
ProjectIdStr	String	项目ID string类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

WorkspaceSetItem

工作空间详情

被如下接口引用：DescribeWorkSpaces。

名称	类型	描述
SerialId	String	工作空间 SerialId 示例值：spc-xxx
AppId	Integer	用户 APPID 示例值：32442423
OwnerUin	String	主账号 UIN 示例值：424244

名称	类型	描述
CreatorUin	String	创建者 UIN 示例值：5345343
WorkSpaceName	String	工作空间名称 示例值：myspace
Region	String	区域 示例值：ap-beijing
CreateTime	String	创建时间 示例值：2020-01-21 00:00:10
UpdateTime	String	更新时间 示例值：2020-01-21 00:00:10
Status	Integer	1 未初始化 2 可用 -1 已删除 示例值：2
Description	String	工作空间描述 示例值：description
ClusterGroupSetItem	Array of ClusterGroupSetItem	工作空间包含集群信息
RoleAuth	Array of RoleAuth	工作空间角色的信息
RoleAuthCount	Integer	工作空间成员数量 示例值：12
WorkSpaceId	String	工作空间 SerialId 示例值：spc_xxx
JobsCount	Integer	1 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

错误码

最近更新时间：2024-12-30 01:49:52

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct.",
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
ActionOffline	接口已下线。
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在 控制台 检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP 地址在黑名单中。
IpNotInWhitelist	IP 地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。

错误码	说明
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP 限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
AuthFailure	CAM签名/鉴权错误。
FailedOperation.DataSourceConnectionFailed	数据库连接失败，请检查参数是否填写正确。
FailedOperation.DeleteWedataItemspace	不能删除wedata项目关联的空间
FailedOperation.DraftConfigCanNotDelete	作业的草稿不允许删除。
FailedOperation.DuplicatedJobName	重复的作业名。
FailedOperation.FailedOperationCode_FetchSqlGatewayStatement	获取SqlGateway结果失败，异常
FailedOperation.FailedOperationCode_RunSqlGateway	运行SqlGateway失败，异常
FailedOperation.GetResourceTagsByResourceIds	查询资源关联标签失败。
FailedOperation.GrammarCheckFailure	语法检查失败。
FailedOperation.JobConfigOnPublish	作业配置在使用中。
FailedOperation.ParseSql	SQL解析失败。
FailedOperation.UserNotAuthenticated	用户未实名验证。
InternalError.CLS	CLS接口错误。
InternalError.COSClient	COS 服务访问错误。
InternalError.CamCgwError	CAM 网关错误。
InternalError.DB	数据库异常。

错误码	说明
InternalError.FailedToDescribeResources	失败的目标资源。
InternalError.FailedToUpdateJob	无法更新作业错误。
InternalError.JobInstanceNotFound	作业实例没找到。
InternalError.LogicError	内部错误。
InternalError.ResourceConfigCanNotDelete	资源只有一个版本，无法删除。
InternalError.ResourceNotExist	资源不存在。
InternalError.SqlCodeNotFound	未找到sqlcode错误。
InternalError.StsNewClient	内部错误。
InternalError.SystemError	系统错误。
InvalidParameter.AppIdResourceNotMatch	AppId资源不匹配。
InvalidParameter.DuplicatedSpaceName	空间命名重复。
InvalidParameter.IllegalMaxParallelism	非法的 MaxParallelism 参数。
InvalidParameter.IllegalSearchKeyword	非法的搜索字符串。
InvalidParameter.InvalidAppId	appid错误。
InvalidParameter.InvalidClusterId	无效集群id。
InvalidParameter.InvalidItemSpaceName	空间命名不规范。
InvalidParameter.InvalidName	名字不符合规范。
InvalidParameter.InvalidRegion	无效Region。
InvalidParameter.InvalidResourceIds	ResourceIds非法。
InvalidParameter.JobConfigLogCollectParamError	创建作业配置错误。
InvalidParameter.MaxParallelismTooLarge	MaxParallelism 过大。
InvalidParameter.MaxParallelismTooSmall	MaxParallelism 不允许小于算子默认并行度。
InvalidParameter.UinResourceNotMatch	Uin资源不匹配。
InvalidParameter.UnsupportedFlinkConf	Flink参数非法。
InvalidParameterValue.ClusterId	集群Id指定错误，为空或者不存在。
InvalidParameterValue.ClusterIds	群集Id报错。
InvalidParameterValue.CuMem	CU内存规格不匹配。
InvalidParameterValue.InvalidLimit	无效限制错误。
InvalidParameterValue.InvalidStartMode	无效启动模式。
InvalidParameterValue.InvalidTime	无效时间错误。
InvalidParameterValue.JobIdValueError	作业id的参数无效。
InvalidParameterValue.JobName	Illegal JobName。
InvalidParameterValue.JobNameExisted	作业名称已存在。
InvalidParameterValue.JobTypeCombineWithClusterType	集群模式与作业类型不匹配。
InvalidParameterValue.JobTypeCombineWithEntrypointClass	SQL作业不能指定EntrypointClass，JAR作业则必须指定。
InvalidParameterValue.OrderType	OrderType值错误。
InvalidParameterValue.ResourceIdsNotFound	未找到资源ID。

错误码	说明
InvalidParameterValue.RunJobDescriptionsCount	批量运行作业个数超过上限。
InvalidParameterValue.RunType	RunType错误。
InvalidParameterValue.Timestamp	时间戳范围有误。
InvalidParameterValue.Type	非法的类型字段。
InvalidParameterValue.UnSupportedComposite	不支持的复合类型。
InvalidParameterValue.UnknownStopType	未知停止类型错误。
LimitExceeded.ItemSpaceLimitExceeded	项目空间数量超出限制。
LimitExceeded.Job	Job个数超过限额。
LimitExceeded.JobConfig	作业配置超出限制。
LimitExceeded.WorkSpaceLimitExceeded	工作空间超阈值。
ResourceInUse.InstanceInProcess	集群有其他操作。
ResourceInUse.ResourceNameAlreadyExists	资源名称已存在。
ResourceInsufficient.CU	CU资源不足。
ResourceNotFound.COSBucket	COS Bucket 未找到或无权限访问。
ResourceNotFound.ClusterId	集群不存在。
ResourceNotFound.Job	作业不存在。
ResourceNotFound.JobConfig	作业配置版本不存在。
ResourceNotFound.JobId	找不到作业。
ResourceNotFound.LogTopic	找不到日志。
ResourceNotFound.Resource	程序包不存在。
ResourceNotFound.ResourceConfig	程序包版本不存在。
ResourceNotFound.ResourceNotExist	资源不存在。
ResourceNotFound.TableNotExist	数据库或表不存在。
ResourceUnavailable.BeUseBySomeJobConfig	被某个作业配置使用。
ResourceUnavailable.CheckResourceLocExists	检查资源位置是否存在错误。
ResourceUnavailable.Cluster	集群处于非运行状态。
ResourceUnavailable.ClusterGroupStatus	群集组状态错误。
ResourceUnavailable.FailedToDescribeResources	失败的目标资源。
ResourceUnavailable.GetJobPublishedJobConfig	获取发布的作业配置错误。
ResourceUnavailable.JobResourceConfigNotReady	作业资源配置未就绪。
ResourceUnavailable.NoRunningJobInstancesFoundForJobId	找不到JobId的正在运行的作业实例。
ResourceUnavailable.NotAllowedToBeDeleted	不允许删除错误。
ResourceUnavailable.NotAllowedToBeStopOrPause	不允许停止或暂停错误。
ResourceUnavailable.RegCuMem	共享群集只允许CuMem=4。
ResourceUnavailable.ResourceLocNotExists	cos上的程序包不存在。
UnsupportedOperation.CanNotDelete	资源未释放,不能做删除作业。
UnsupportedOperation.ClsSqlNotEnabled	Cls Sql未启用。

错误码	说明
UnsupportedOperation.InvalidCheckpointIntervalError	Checkpoint 时间间隔，错误。
UnsupportedOperation.NoPermissionAccess	权限拦截,没有进入权限。
UnsupportedOperation.UnsupportedStartMode	不支持的启动模式。