

云数据库 MongoDB

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

产品版本

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

实例相关接口

指定云数据库实例的所属项目

获取数据库实例订单详情

获取实例连接的URI。

查询实例的库表。

获取慢日志统计信息

查询云数据库的售卖规格

创建实例询价

续费实例询价

隔离云数据库实例

终止数据库实例特定操作

下线隔离状态的云数据库实例

修改实例名称

续费云数据库实例

重启数据库节点

设置云数据库实例的自动备份规则

包年包月隔离接口

创建云数据库实例（包年包月）

创建云数据库实例（按量计费）

删除账号

获取云数据库实例自动备份配置

查询实例客户端连接信息

查询数据库实例当前正在执行的操作

查询节点属性

查询云数据库实例列表

获取当前实例可修改的参数列表

查询实例绑定的安全组

获取慢日志信息

获取实例透明加密的开启状态

开启实例数据透明加密

变更配置询价

修改云数据库实例网络信息

修改实例安全组

调整云数据库实例配置

修改mongoDB实例参数

实例维护时间设置

查询实例慢日志详情

备份相关接口

- 创建备份下载任务
- 查询备份下载任务信息
- 查询实例备份列表
- 备份实例接口

账号相关接口

- 创建账号
- 全部账号列表
- 修改实例用户的密码
- 设置账户权限

任务相关接口

- 查询异步任务状态接口

按key回档相关接口

- 按 Key 回档

其他接口

- 创建数据库参数模板
- 查询实例参数模板列表
- 参数模板详情
- 删除参数模板
- 刷新路由配置
- 修改数据库参数模板

数据结构

错误码

云数据库 MongoDB API 2018-04-08

更新历史

简介

API 概览

调用方式

- 请求结构
- 公共参数
- 接口鉴权 v3
- 接口鉴权
- 返回结果
- 参数类型

实例相关接口

- 升级云数据库实例（按量计费）
- 升级云数据库实例（包年包月）
- 销毁云数据库实例（按量计费--废弃）
- 创建云数据库实例（按量计费）
- 创建云数据库实例（包年包月）
- 查询实例客户端连接信息
- 设置云数据库实例的续费选项
- 修改实例名称
- 查询云数据库的售卖规格
- 查询实例列表
- 指定云数据库实例的所属项目
- 查询实例慢日志

账号相关接口

- 修改云数据库实例的账户密码

数据结构

错误码

API 文档

产品版本

最近更新时间：2025-06-20 01:48:22

您正在浏览mongodb产品文档的版本: 2019-07-25

最新版本

- [2019-07-25](#) (当前版本)

以往版本

- [2018-04-08](#)

更新历史

最近更新时间：2025-06-30 02:05:32

第 53 次发布

发布时间：2025-06-30 02:04:59

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDetailedSlowLogs](#)

新增数据结构：

- [SlowLogItem](#)

第 52 次发布

发布时间：2025-06-27 02:10:46

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SlowLogPattern](#)
 - 新增成员：QueryHash

第 51 次发布

发布时间：2025-05-12 01:45:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyDBInstanceSpec](#)
 - 废弃入参：OplogSize

第 50 次发布

发布时间：2025-03-14 01:21:25

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDBInstanceNamespace](#)

第 49 次发布

发布时间：2025-02-12 01:46:31

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDBInstanceURL](#)

新增数据结构：

- [DbURL](#)

第 48 次发布

发布时间：2025-01-08 01:23:10

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [BackupDownloadTask](#)
 - 新增成员：Region, Bucket
- [NodeProperty](#)
 - 新增成员：WanServiceAddress

第 47 次发布

发布时间：2024-12-26 01:47:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [KMSInfoDetail](#)
 - 新增成员：KmsRegion

第 46 次发布

发布时间：2024-12-19 01:47:13

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [CurrentOp](#)
 - 新增成员：ExecNode

第 45 次发布

发布时间：2024-08-26 02:06:18

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyDBInstanceSpec](#)
 - 修改入参：Memory, Volume

第 44 次发布

发布时间：2024-07-25 01:56:02

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [RestartNodes](#)

修改接口：

- [CreateDBInstance](#)
 - 新增入参：ParamTemplateId
- [CreateDBInstanceHour](#)
 - 新增入参：ParamTemplateId

第 43 次发布

发布时间：2024-07-24 01:54:54

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateDBInstanceParamTpl](#)
- [DescribeDBInstanceParamTpl](#)
- [DescribeDBInstanceParamTplDetail](#)
- [DropDBInstanceParamTpl](#)
- [ModifyDBInstanceParamTpl](#)

新增数据结构：

- [ParamTpl](#)
- [ParamType](#)

第 42 次发布

发布时间：2024-07-15 01:15:52

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyDBInstanceSpec](#)
 - 新增入参：MongosMemory

第 41 次发布

发布时间：2024-06-21 01:58:00

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ModifyInstanceParams](#)

新增数据结构：

- [ModifyMongoDBParamType](#)

第 40 次发布

发布时间：2024-06-18 01:57:52

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [BackupInfo](#)
 - 新增成员：BackId, DeleteTime, BackupRegion

第 39 次发布

发布时间：2024-06-12 02:08:47

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeBackupRules](#)
- [DescribeTransparentDataEncryptionStatus](#)
- [EnableTransparentDataEncryption](#)
- [SetBackupRules](#)

新增数据结构：

- [KMSInfoDetail](#)

第 38 次发布

发布时间：2024-04-09 01:18:23

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [FlashBackDBInstance](#)

新增数据结构：

- [FBKeyValue](#)
- [FlashbackCollection](#)
- [FlashbackDatabase](#)

第 37 次发布

发布时间：2024-03-18 01:18:19

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyDBInstanceNetworkAddress](#)
 - 新增出参: FlowId

第 36 次发布

发布时间: 2024-03-13 01:19:31

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [SetInstanceMaintenance](#)

第 35 次发布

发布时间: 2024-03-05 01:20:29

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [DescribeDBInstanceDeal](#)
 - 新增出参: InstanceId

第 34 次发布

发布时间: 2024-01-29 01:17:07

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [InstanceDetail](#)
 - 新增成员: ZoneList

第 33 次发布

发布时间: 2023-12-27 01:26:46

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [ModifyDBInstanceSpec](#)
 - 新增入参: AddNodeList, RemoveNodeList

新增数据结构:

- [AddNodeList](#)
- [RemoveNodeList](#)

第 32 次发布

发布时间: 2023-04-11 01:28:42

本次发布包含了以下内容:

改善已有的文档。

新增接口：

- [DeleteAccountUser](#)

第 31 次发布

发布时间：2023-03-31 01:36:51

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeDBInstances](#)
 - 修改入参：Tags

第 30 次发布

发布时间：2023-03-01 01:35:11

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstanceHour](#)
 - 新增入参：ReadonlyNodeNum, ReadonlyNodeAvailabilityZoneList, HiddenZone

第 29 次发布

发布时间：2023-02-22 01:37:39

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceDetail](#)
 - 新增成员：MongosNodeNum, MongosMemory, MongosCpuNum, ConfigServerNodeNum, ConfigServerMemory, ConfigServerVolume, ConfigServerCpuNum, ReadonlyNodeNum

第 28 次发布

发布时间：2023-01-02 01:31:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAccountUser](#)

第 27 次发布

发布时间：2022-11-25 06:45:10

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstance](#)
 - 新增入参：ReadonlyNodeNum, ReadonlyNodeAvailabilityZoneList, HiddenZone

第 26 次发布

发布时间：2022-10-28 06:36:39

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDBInstanceNodeProperty](#)

新增数据结构：

- [NodeProperty](#)
- [NodeTag](#)
- [ReplicateSetInfo](#)

第 25 次发布

发布时间：2022-10-10 06:31:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [InquirePriceCreateDBInstances](#)
 - 新增入参：InstanceChargeType, MongosCpu, MongosMemory, MongosNum, ConfigServerCpu, ConfigServerMemory, ConfigServerVolume
 - 修改入参：Period

第 24 次发布

发布时间：2022-09-21 06:37:03

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeAccountUsers](#)

新增数据结构：

- [UserInfo](#)

第 23 次发布

发布时间：2022-08-15 06:31:24

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [TerminateDBInstances](#)

第 22 次发布

发布时间：2022-08-11 06:40:51

本次发布包含了以下内容：

改善已有的文档。

删除接口：

- DescribeBackupAccess

删除数据结构：

- BackupFile

第 21 次发布

发布时间：2022-06-07 06:10:21

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SpecificationInfo](#)
 - 新增成员：SupportMultiAZ

第 20 次发布

发布时间：2022-04-01 06:11:50

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ModifyDBInstanceNetworkAddress](#)

新增数据结构：

- [ModifyNetworkAddress](#)

第 19 次发布

发布时间：2022-03-03 08:08:57

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ModifyDBInstanceSecurityGroup](#)

第 18 次发布

发布时间：2022-02-25 08:11:43

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [InquirePriceModifyDBInstanceSpec](#)
 - 新增入参：NodeNum, ReplicateSetNum

修改数据结构：

- [BackupDownloadTask](#)
 - 新增成员：BackupMethod, BackupDesc

第 17 次发布

发布时间：2022-01-05 08:11:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [SetAccountUserPrivilege](#)

新增数据结构：

- [Auth](#)

第 16 次发布

发布时间：2021-12-14 08:12:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeInstanceParams](#)

新增数据结构：

- [InstanceEnumParam](#)
- [InstanceIntegerParam](#)
- [InstanceMultiParam](#)
- [InstanceTextParam](#)

第 15 次发布

发布时间：2021-12-09 08:08:23

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ClientConnection](#)
 - 新增成员：InternalService

第 14 次发布

发布时间：2021-09-27 08:14:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstance](#)
 - 新增入参：RestoreTime, InstanceName, AvailabilityZoneList, MongosCpu, MongosMemory, MongosNodeNum
- [CreateDBInstanceHour](#)
 - 新增入参：RestoreTime, InstanceName, AvailabilityZoneList, MongosCpu, MongosMemory, MongosNodeNum

第 13 次发布

发布时间：2021-08-24 08:07:53

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyDBInstanceSpec](#)
 - 新增入参：NodeNum, ReplicateSetNum, InMaintenance

第 12 次发布

发布时间：2021-05-27 08:06:27

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SecurityGroupBound](#)
 - 新增成员：Id, AddressModule, ServiceModule, Desc

第 11 次发布

发布时间：2021-05-12 08:06:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeSecurityGroup](#)

新增数据结构：

- [SecurityGroup](#)
- [SecurityGroupBound](#)

第 10 次发布

发布时间：2021-03-29 08:05:40

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateBackupDownloadTask](#)

- [DescribeBackupDownloadTask](#)

新增数据结构：

- [BackupDownloadTask](#)
- [BackupDownloadTaskStatus](#)
- [ReplicaSetInfo](#)

第 9 次发布

发布时间：2021-03-03 08:05:46

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeSlowLogPatterns](#)
 - 新增入参：Format
- [DescribeSlowLogs](#)
 - 新增入参：Format

第 8 次发布

发布时间：2020-09-16 08:04:22

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeDBInstances](#)
 - 新增入参：Tags

第 7 次发布

发布时间：2020-09-04 08:04:58

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeCurrentOp](#)
- [KillOps](#)

修改接口：

- [DescribeDBBackups](#)
 - 新增入参：BackupMethod, Limit, Offset

新增数据结构：

- [CurrentOp](#)
- [Operation](#)

第 6 次发布

发布时间：2020-07-22 08:14:25

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateBackupDBInstance](#)
- [DescribeAsyncRequestInfo](#)
- [ResetDBInstancePassword](#)

第 5 次发布

发布时间：2020-07-20 08:13:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstance](#)
 - 新增入参：AutoVoucher, Clone, Father, SecurityGroup
- [CreateDBInstanceHour](#)
 - 新增入参：Clone, Father, SecurityGroup

第 4 次发布

发布时间：2020-06-02 08:13:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [FlushInstanceRouterConfig](#)

第 3 次发布

发布时间：2020-04-22 08:12:39

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDBInstanceDeal](#)
- [InquirePriceCreateDBInstances](#)
- [InquirePriceModifyDBInstanceSpec](#)
- [InquirePriceRenewDBInstances](#)

新增数据结构：

- [DBInstancePrice](#)

第 2 次发布

发布时间：2020-04-21 08:12:45

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeSlowLogPatterns](#)
- [DescribeSlowLogs](#)
- [RenewDBInstances](#)

修改接口：

- [DescribeClientConnections](#)
 - 新增入参：Limit, Offset
 - 新增出参：TotalCount

新增数据结构：

- [InstanceChargePrepaid](#)
- [SlowLogPattern](#)

第 1 次发布

发布时间：2019-10-18 16:36:58

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AssignProject](#)
- [CreateDBInstance](#)
- [CreateDBInstanceHour](#)
- [DescribeBackupAccess](#)
- [DescribeClientConnections](#)
- [DescribeDBBackups](#)
- [DescribeDBInstances](#)
- [DescribeSpecInfo](#)
- [IsolateDBInstance](#)
- [ModifyDBInstanceSpec](#)
- [OfflineIsolatedDBInstance](#)
- [RenameInstance](#)

新增数据结构：

- [BackupFile](#)
- [BackupInfo](#)
- [ClientConnection](#)
- [DBInstanceInfo](#)
- [InstanceDetail](#)
- [ShardInfo](#)
- [SpecItem](#)
- [SpecificationInfo](#)
- [TagInfo](#)

简介

最近更新时间：2025-04-25 01:45:29

云数据库 MongoDB（TencentDB for MongoDB），是腾讯云基于全球广受欢迎的 MongoDB 打造的高性能 NoSQL 数据库，100% 完全兼容 MongoDB 协议，提供稳定丰富的监控管理，弹性可扩展、自动容灾，适用于文档型数据库场景。

注意：

- 本章节云数据库 MongoDB API 接口均为最新 API 3.0 接口，后续云数据库 MongoDB 相关新增功能都会在此章节更新。我们强烈推荐您使用最新 API3.0 接口。
- 现有旧版 API 接口功能依然保持，未来可能停止维护，如您仍需使用旧版接口可参考：[云数据库 MongoDB API（旧版）简介](#)。

API 概览

最近更新时间：2025-06-30 02:05:31

实例相关接口

接口名称	接口功能	频率限制（次/秒）
AssignProject	指定云数据库实例的所属项目	20
DescribeDBInstanceDeal	获取数据库实例订单详情	5
DescribeDBInstanceURL	获取实例连接的URI。	20
DescribeDBInstanceNamespace	查询实例的库表。	20
DescribeSlowLogPatterns	获取慢日志统计信息	20
DescribeSpecInfo	查询云数据库的售卖规格	10
InquirePriceCreateDBInstances	创建实例询价	20
InquirePriceRenewDBInstances	续费实例询价	20
IsolateDBInstance	隔离云数据库实例	20
KillOps	终止数据库实例特定操作	20
OfflineIsolatedDBInstance	下线隔离状态的云数据库实例	10
RenameInstance	修改实例名称	20
RenewDBInstances	续费云数据库实例	20
RestartNodes	重启数据库节点	20
SetBackupRules	设置云数据库实例的自动备份规则	20
TerminateDBInstances	包年包月隔离接口	20
CreateDBInstance	创建云数据库实例（包年包月）	20
CreateDBInstanceHour	创建云数据库实例（按量计费）	20
DeleteAccountUser	删除账号	20
DescribeBackupRules	获取云数据库实例自动备份配置	20
DescribeClientConnections	查询实例客户端连接信息	5
DescribeCurrentOp	查询数据库实例当前正在执行的操作	20
DescribeDBInstanceNodeProperty	查询节点属性	3
DescribeDBInstances	查询云数据库实例列表	20
DescribeInstanceParams	获取当前实例可修改的参数列表	20
DescribeSecurityGroup	查询实例绑定的安全组	20
DescribeSlowLogs	获取慢日志信息	20
DescribeTransparentDataEncryptionStatus	获取实例透明加密的开启状态	20
EnableTransparentDataEncryption	开启实例数据透明加密	20

接口名称	接口功能	频率限制（次/秒）
InquirePriceModifyDBInstanceSpec	变更配置询价	20
ModifyDBInstanceNetworkAddress	修改云数据库实例网络信息	20
ModifyDBInstanceSecurityGroup	修改实例安全组	20
ModifyDBInstanceSpec	调整云数据库实例配置	20
ModifyInstanceParams	修改mongoDB实例参数	20
SetInstanceMaintenance	实例维护时间设置	20
DescribeDetailedSlowLogs	查询实例慢日志详情	20

备份相关接口

接口名称	接口功能	频率限制（次/秒）
CreateBackupDownloadTask	创建备份下载任务	20
DescribeBackupDownloadTask	查询备份下载任务信息	20
DescribeDBBackups	查询实例备份列表	150
CreateBackupDBInstance	备份实例接口	20

账号相关接口

接口名称	接口功能	频率限制（次/秒）
CreateAccountUser	创建账号	20
DescribeAccountUsers	全部账号列表	20
ResetDBInstancePassword	修改实例用户的密码	20
SetAccountUserPrivilege	设置账户权限	20

其他接口

接口名称	接口功能	频率限制（次/秒）
CreateDBInstanceParamTpl	创建数据库参数模板	20
DescribeDBInstanceParamTpl	查询实例参数模板列表	20
DescribeDBInstanceParamTplDetail	参数模板详情	20
DropDBInstanceParamTpl	删除参数模板	20
FlushInstanceRouterConfig	刷新路由配置	20
ModifyDBInstanceParamTpl	修改数据库参数模板	20

任务相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeAsyncRequestInfo	查询异步任务状态接口	20

按key回档相关接口

接口名称	接口功能	频率限制（次/秒）
FlashBackDBInstance	按 Key 回档	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

最近更新时间：2025-05-21 01:25:05

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 `mongodb.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `mongodb.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 `mongodb.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）	mongodb.tencentcloudapi.com
华南地区（广州）	mongodb.ap-guangzhou.tencentcloudapi.com
华东地区（上海）	mongodb.ap-shanghai.tencentcloudapi.com
华东地区（南京）	mongodb.ap-nanjing.tencentcloudapi.com
华北地区（北京）	mongodb.ap-beijing.tencentcloudapi.com
西南地区（成都）	mongodb.ap-chengdu.tencentcloudapi.com
西南地区（重庆）	mongodb.ap-chongqing.tencentcloudapi.com
港澳台地区（中国香港）	mongodb.ap-hongkong.tencentcloudapi.com
亚太东南（新加坡）	mongodb.ap-singapore.tencentcloudapi.com
亚太东南（雅加达）	mongodb.ap-jakarta.tencentcloudapi.com
亚太东南（曼谷）	mongodb.ap-bangkok.tencentcloudapi.com
亚太东北（首尔）	mongodb.ap-seoul.tencentcloudapi.com
亚太东北（东京）	mongodb.ap-tokyo.tencentcloudapi.com
美国东部（弗吉尼亚）	mongodb.na-ashburn.tencentcloudapi.com
美国西部（硅谷）	mongodb.na-siliconvalley.tencentcloudapi.com
南美地区（圣保罗）	mongodb.sa-saopaulo.tencentcloudapi.com
欧洲地区（法兰克福）	mongodb.eu-frankfurt.tencentcloudapi.com

注意：由于金融区和非金融区是隔离不互通的，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）	mongodb.ap-shanghai-fsi.tencentcloudapi.com

金融区接入地域	金融区域名
华南地区（深圳金融）	mongodb.ap-shenzhen-fsi.tencentcloudapi.com

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：2025-03-28 01:48:10

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入参公共参数 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 mongoddb；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, Sign
```

```
edHeaders=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 。
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
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 01:48:10

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区实例列表作为例子，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；

2. 该接口是只读的，不会改变现有资源的状态；
3. 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。

字段名称	解释
	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 是二进制的的数据，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, 否则容易出错
    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++)  
    {  
        sprintf(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 = "{\n  \"Limit\": 1, \"Filters\": [{\n    \"Values\": [\n      \"\u672a\u547d\u540d\"], \"Name\": \"instance-name\"}]}\n";
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++)
    {
        sprintf(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;
#ifdef 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);

#ifdef 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, "\nxc-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 01:48:11

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：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云 API 密钥](#) 的控制台页面
3. 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

参数名称	中文	参数值
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。

签名原文串的拼接规则为：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-guangzhou&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&Region=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都需要做 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();
```

```
$param["Timestamp"] = 1465185768;//time();
$param["Region"] = "ap-guangzhou";
$param["SecretId"] = $secretId;
$param["Version"] = "2017-03-12";
$param["Action"] = "DescribeInstances";
$param["InstanceIds.0"] = "ins-09dx96dg";
$param["Limit"] = 20;
$param["Offset"] = 0;

ksort($param);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $param 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
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// 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. 拼接签名原文字符串
```

```
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 01:40:21

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 06:40:39

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。

实例相关接口

指定云数据库实例的所属项目

最近更新时间：2025-04-25 01:45:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(AssignProject)用于指定云数据库实例的所属项目。

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：AssignProject。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	是	Array of String	实例 ID 列表，请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：["cmgo-eqmo****"]
ProjectId	是	Integer	项目ID，用户已创建项目的唯一ID,非自定义 示例值：1118****

3. 输出参数

参数名称	类型	描述
FlowIds	Array of Integer	返回的异步任务ID列表 示例值：[1081****]
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 指定云数据库实例的所属项目

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=AssignProject
&InstanceIds.0=cmgo-eqmoa5sf
```

```
&ProjectId=12
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "FlowIds": [
      1081
    ],
    "RequestId": "8e2a339e-5ba9-4da3-8dc2-65a0a6314811"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.ProjectNotFound	项目不存在。

获取数据库实例订单详情

最近更新时间：2025-04-25 01:45:26

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeDBInstanceDeal）用于获取MongoDB购买、续费及变配订单详细。

默认接口请求频率限制：5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：DescribeDBInstanceDeal。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
DealId	是	String	订单ID，通过CreateDBInstance等接口返回 示例值：20200420111635

3. 输出参数

参数名称	类型	描述
Status	Integer	订单状态，1：未支付，2：已支付，3：发货中，4：发货成功，5：发货失败，6：退款，7：订单关闭，8：超时未支付关闭。 示例值：4
OriginalPrice	Float	订单原价。 示例值：1116.67
DiscountPrice	Float	订单折扣价格。 示例值：759.33
Action	String	订单行为，purchase：新购，renew：续费，upgrade：升配，downgrade：降配，refund：退货退款。 示例值：purchase
InstanceId	String	当前订单的资源Id。 示例值：cmgo-**
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 查询订单详情

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeDBInstanceDeal
&DealId=20200420111635
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Action": "purchase",
    "DiscountPrice": 759.33,
    "OriginalPrice": 1116.67,
    "RequestId": "e79836e8-1864-45bf-aa63-0c7af31e7de5",
    "Status": 4
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

获取实例连接的URI。

最近更新时间：2025-04-25 01:45:25

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeDBInstanceURL）用于获取指定实例的 URI 形式的连接串访问地址示例。

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：DescribeDBInstanceURL。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-p8vn****

3. 输出参数

参数名称	类型	描述
Urls	Array of DbURL	实例 URI 形式的连接串访问地址示例。包含：URI 类型及连接串地址。
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 获取访问实例的 URI。

获取指定实例的 URI 形式的连接串访问地址示例，包含 URI 类型、URI 连接串。

输入示例

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

{
```

```
"InstanceId": "cmgo-g5*****"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8e4cb480-9d32-11ec-8dfc-037b69052954",
    "Urls": [
      {
        "Address": "mongodb://mongouser:*****@10.*.0.**:27017,10.*.0.**:27017/test?authSource=admin",
        "URLType": "CLUSTER_ALL"
      },
      {
        "Address": "mongodb://mongouser:*****@10.*.0.**:27017,10.*.0.**:27017/test?authSource=admin&readPreference=secondaryPreferred",
        "URLType": "CLUSTER_READ_SECONDARY"
      }
    ]
  }
}
```

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
InternalServerError	内部错误。

错误码	描述
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

查询实例的库表。

最近更新时间：2025-04-25 01:45:26

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeDBInstanceNamespaces）用于查询数据库的表信息。

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：DescribeDBInstanceNamespaces。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定查询数据库所属的实例 ID，支持批量查询。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值："cmgo-j0i**"
DbName	否	String	指定查询的数据库名。为空时，返回当前实例的全部数据库列表。 示例值：dba

3. 输出参数

参数名称	类型	描述
Databases	Array of String	查询实例的数据库列表。若未使用 DbName 指定具体查询的数据库，则仅返回查询实例所有的数据库列表，而不返回 Collections 集合信息。 示例值：['db-a','db-b']
Collections	Array of String	查询的集合信息。指定 DbName 时，则仅返回该数据库下的集合列表。 示例值：['col-1','col-2',]
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 查询当前实例的DB列表。

输入参数 DbName 为空时，查看当前实例全部数据库列表。

输入示例


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

```
{
  "InstanceId": "cmgo-qny2****"
}
```

输出示例

```
{
  "Response": {
    "Databases": [
      "demo_db7",
      "demo_db8",
      "demo_db9"
    ],
    "RequestId": "18cfe2bf-6934-4f17-a089-8d2e5d6cfc53"
  }
}
```

示例2 查询指定 DB 下的所有集合

指定 DbName，查询该数据库下的所有集合信息。

输入示例

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

```
{
  "InstanceId": "cmgo-qny2****",
  "DbName": "demo_db0"
}
```

输出示例

```
{
  "Response": {
    "Collections": [
      "col148",
      "col108",
      "col120"
    ],
    "RequestId": "21b503b5-6606-4efb-b250-63071c2fe565"
  }
}
```

```
}  
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。

获取慢日志统计信息

最近更新时间：2025-04-25 01:45:25

1. 接口描述

接口请求域名： mongoddb.tencentcloudapi.com 。

本接口（DescribeSlowLogPatterns）用于获取数据库实例慢日志的统计信息。

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：DescribeSlowLogPatterns。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-a3bm93hf
StartTime	是	Timestamp	慢日志起始时间，格式：yyyy-mm-dd hh:mm:ss，如：2019-06-01 10:00:00。查询起止时间间隔不能超过24小时，只允许查询最近7天内慢日志。 示例值：2020-04-20 00:00:00
EndTime	是	Timestamp	慢日志终止时间，格式：yyyy-mm-dd hh:mm:ss，如：2019-06-02 12:00:00。查询起止时间间隔不能超过24小时，只允许查询最近7天内慢日志。 示例值：2020-04-20 22:00:00
SlowMS	是	Integer	慢日志执行时间阈值，返回执行时间超过该阈值的慢日志，单位为毫秒(ms)，最小为100毫秒。 示例值：100
Offset	否	Integer	偏移量，最小值为0，最大值为10000，默认值为0。 示例值：0
Limit	否	Integer	分页大小，最小值为1，最大值为100，默认值为20。 示例值：100
Format	否	String	慢日志返回格式，可设置为json，不传默认返回原生慢日志格式。 示例值：json

3. 输出参数

参数名称	类型	描述
Count	Integer	慢日志统计信息总数 示例值：2
SlowLogPatterns	Array of	慢日志统计信息

参数名称	类型	描述
	SlowLogPattern	
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 查询数据库慢日志统计信息

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeSlowLogPatterns
&InstanceId=cmgo-a3bm93hf
&StartTime=2020-04-20 00:00:00
&EndTime=2020-04-20 22:00:00
&SlowMS=100
&Offset=0
&Limit=100
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Count": 2,
    "RequestId": "895de939-f726-4cb0-bb02-3890289479e3",
    "SlowLogPatterns": [
      {
        "AverageTime": 699,
        "MaxTime": 5335,
        "Pattern": "admin.$cmd.command",
        "Total": 12
      },
      {
        "AverageTime": 271,
        "MaxTime": 423,
        "Pattern": "local.system.replset.command",
        "Total": 8
      }
    ]
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.QueryTimeOutOfRange	只能查询7天内的慢日志。
InvalidParameterValue.QueryTimeRangeBeyondLimit	只能查询7天内的慢日志。
InvalidParameterValue.SlowMSBelowLimit	slowMS参数取值范围[100,65536]。
InvalidParameterValue.StartTimeNotBeforeThanEndTime	起始时间晚于结束时间。

查询云数据库的售卖规格

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeSpecInfo)用于查询实例的售卖规格。

默认接口请求频率限制：10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：DescribeSpecInfo。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Zone	否	String	待查询可用区 示例值：ap-guangzhou-3

3. 输出参数

参数名称	类型	描述
SpecInfoList	Array of SpecificationInfo	实例售卖规格信息列表
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 查询云数据库售卖

输入示例

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

{
```

```
"Zone": "ap-guangzhou-3"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "a8c7ade4-2d22-4f5e-9e71-f546f15e6d67",
    "SpecInfoList": [
      {
        "Region": "ap-guangzhou",
        "SpecItems": [
          {
            "ClusterType": 0,
            "Conns": 1500,
            "Cpu": 2,
            "DefaultStorage": 25600,
            "EngineName": "WiredTiger",
            "MachineType": "HIO10G",
            "MaxNodeNum": 3,
            "MaxReplicateSetNodeNum": 5,
            "MaxReplicateSetNum": 20,
            "MaxStorage": 512000,
            "Memory": 4096,
            "MinNodeNum": 3,
            "MinReplicateSetNodeNum": 3,
            "MinReplicateSetNum": 2,
            "MinStorage": 102400,
            "MongoVersionCode": "MONGO_36_WT",
            "MongoVersionValue": 4,
            "Qps": 5000,
            "SpecCode": "mongo.HIO10G.4g",
            "Status": 1,
            "Version": "3.6"
          },
          {
            "ClusterType": 0,
            "Conns": 1500,
            "Cpu": 2,
            "DefaultStorage": 25600,
            "EngineName": "WiredTiger",
            "MachineType": "HIO10G",
            "MaxNodeNum": 3,
            "MaxReplicateSetNodeNum": 5,
            "MaxReplicateSetNum": 20,
            "MaxStorage": 512000,
            "Memory": 4096,
            "MinNodeNum": 3,
            "MinReplicateSetNodeNum": 3,
            "MinReplicateSetNum": 2,
            "MinStorage": 102400,
```

```
"MongoVersionCode": "MONGO_3_WT",
"MongoVersionValue": 2,
"Qps": 5000,
"SpecCode": "mongo.HIO10G.4g",
"Status": 1,
"Version": "3.2"
},
{
"ClusterType": 1,
"Conns": 15000,
"Cpu": 48,
"DefaultStorage": 4096000,
"EngineName": "WiredTiger",
"MachineType": "HIO10G",
"MaxNodeNum": 3,
"MaxReplicateSetNodeNum": 5,
"MaxReplicateSetNum": 20,
"MaxStorage": 6144000,
"Memory": 524288,
"MinNodeNum": 3,
"MinReplicateSetNodeNum": 3,
"MinReplicateSetNum": 2,
"MinStorage": 1536000,
"MongoVersionCode": "MONGO_36_WT",
"MongoVersionValue": 4,
"Qps": 42000,
"SpecCode": "mongo.HIO10G.512g",
"Status": 1,
"Version": "3.6"
}
],
"SupportMultiAZ": 1,
"Zone": "ap-guangzhou-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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.ZoneError	无效的可用区。

创建实例询价

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（InquirePriceCreateDBInstances）用于创建数据库实例询价。本接口参数中必须传入region参数，否则无法通过校验。本接口仅允许针对购买限制范围内的实例配置进行询价。

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：InquirePriceCreateDBInstances。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Zone	是	String	实例所属区域及可用区信息。格式：ap-guangzhou-2。 示例值：ap-guangzhou-4
NodeNum	是	Integer	– 创建副本集实例，指每个副本集内主从节点数量。每个副本集所支持的的最大节点数与最小节点数，请通过接口 DescribeSpecInfo 获取。 – 创建分片集群实例，指每个分片的主从节点数量。每个分片所支持的最大节点数与最小节点数，请通过接口 DescribeSpecInfo 获取。 示例值：3
Memory	是	Integer	实例内存大小。 – 单位：GB。 – 取值范围：请通过接口 DescribeSpecInfo 查询，其返回的数据结构SpecItems中的参数CPU与Memory分别对应CPU核数与内存规格。 示例值：4
Volume	是	Integer	实例硬盘大小。 – 单位：GB。 – 取值范围：请通过接口 DescribeSpecInfo 查询，其返回的数据结构SpecItems中的参数MinStorage与MaxStorage分别对应其最小磁盘规格与最大磁盘规格。 示例值：100
MongoVersion	是	String	实例版本信息。具体支持的版本，请通过接口 DescribeSpecInfo 查询，其返回的数据结构SpecItems中的参数MongoVersionCode为实例所支持的版本信息。版本信息与版本号对应关系如下： – MONGO_36_WT：MongoDB 3.6 WiredTiger存储引擎版本。 – MONGO_40_WT：MongoDB 4.0 WiredTiger存储引擎版本。 – MONGO_42_WT：MongoDB 4.2 WiredTiger存储引擎版本。 – MONGO_44_WT：MongoDB 4.4 WiredTiger存储引擎版本。 – MONGO_50_WT：MongoDB 5.0 WiredTiger存储引擎版本。

参数名称	必选	类型	描述
			– MONGO_60_WT: MongoDB 6.0 WiredTiger存储引擎版本。 示例值: MONGO_36_WT
MachineCode	是	String	产品规格类型。 – HIO10G: 通用高HIO万兆型。 – HCD: 云盘版。 示例值: HIO10G
GoodsNum	是	Integer	实例数量, 取值范围为[1,10]。 示例值: 2
ClusterType	是	String	实例类型。 – REPLSET: 副本集。 – SHARD: 分片集群。 示例值: REPLSET
ReplicateSetNum	是	Integer	– 创建副本集实例, 指副本集数量, 该参数只能为1。 – 创建分片集群实例, 指分片的数量。请通过接口 DescribeSpecInfo 查询分片数量的取值范围, 其返回的数据结构SpecItems中的参数MinReplicateSetNum与MaxReplicateSetNum分别对应其最小值与最大值。 示例值: 1
Period	否	Integer	– 选择包年包月计费模式, 即 InstanceChargeType 设定为PREPAID时, 必须设置该参数, 指定购买实例的购买时长。取值可选: [1,2,3,4,5,6,7,8,9,10,11,12,24,36]; 单位: 月。 – 选择按量计费, 即 InstanceChargeType 设定为 POSTPAID_BY_HOUR 时, 该参数仅可配置为 1。 示例值: 1
InstanceChargeType	否	String	实例付费方式。 – PREPAID: 包年包月计费。 – POSTPAID_BY_HOUR: 按量计费。 示例值: PREPAID
MongosCpu	否	Integer	Mongos CPU 核数, 支持1、2、4、8、16。购买分片集群时, 必须填写。注意为空时取默认取值为2C。 示例值: 2
MongosMemory	否	Integer	Mongos 内存大小。– 购买分片集群时, 必须填写。– 单位: GB, 支持1核2GB、2核4GB、4核8GB、8核16GB、16核32GB。注意为空时取默认取值为4G。 示例值: 4
MongosNum	否	Integer	指 Mongos 个数, 取值范围为[3,32]。若为分片集群实例询价, 则该参数必须设置。注意为空时取默认取值为3个节点。 示例值: 3
ConfigServerCpu	否	Integer	指 ConfigServer CPU核数, 固定取值为 1, 单位: GB, 可不配置该参数。 示例值: 1
ConfigServerMemory	否	Integer	指 ConfigServer 内存大小, 固定取值为 2, 单位: GB, 可不配置该参数。 示例值: 2
ConfigServerVolume	否	Integer	指 ConfigServer 磁盘大小, 固定取值为 20, 单位: GB, 可不配置该参数。 示例值: 20

3. 输出参数

参数名称	类型	描述
Price	DBInstancePrice	价格

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 查询新购数据库实例价格

输入示例

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

{
  "GoodsNum": 2,
  "Zone": "ap-guangzhou-4",
  "Memory": 4,
  "Period": 1,
  "ClusterType": "REPLSET",
  "Volume": 100,
  "NodeNum": 3,
  "ReplicateSetNum": 1,
  "MachineCode": "HIO10G",
  "MongoVersion": "MONGO_50_WT"
}
```

输出示例

```
{
  "Response": {
    "Price": {
      "DiscountPrice": 1340,
      "OriginalPrice": 1340,
      "UnitPrice": 670
    },
    "RequestId": "fb8b4646-389d-44cc-ab8c-98c081cbdad3"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.ReplicaSetNumError	副本集（分片）数量错误。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.ZoneError	无效的可用区。

续费实例询价

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口 (InquiryPriceRenewDBInstances) 用于续费包年包月实例询价。

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：InquirePriceRenewDBInstances。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	是	Array of String	实例ID。请登录 MongoDB 控制台 在实例列表复制实例 ID，且单次最多同时查询5个实例。 示例值：["cmgo-8ojj****"]
InstanceChargePrepaid	是	InstanceChargePrepaid	预付费模式（即包年包月）相关参数设置。通过该参数可以指定包年包月实例的续费时长、是否设置自动续费等属性。

3. 输出参数

参数名称	类型	描述
Price	DBInstancePrice	价格
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 获取续费数据库实例价格

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=InquirePriceRenewDBInstances
&InstanceIds.0=cmgo-8ojj5631
&InstanceChargePrepaid.Period=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Price": {
      "DiscountPrice": 455.6,
      "OriginalPrice": 670,
      "UnitPrice": 0
    },
    "RequestId": "d08f2a0f-4a69-451e-8ddb-753340d5fb34"
  }
}
```

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
InternalServerError	内部错误。
InternalServerError.FindInstanceFailed	实例查询失败。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.InvalidTradeOperation	计费相关错误，不允许对当前实例进行对应的新购/续费/配置变更操作。
InvalidParameterValue.ZoneClosed	可用区已关闭售卖。

隔离云数据库实例

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(IsolateDBInstance)用于隔离MongoDB云数据库按量计费实例。隔离后实例保留在回收站中，不能再写入数据。隔离一定时间后，实例会彻底删除，回收站保存时间请参考按量计费的服务条款。在隔离中的按量计费实例无法恢复，请谨慎操作。

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：IsolateDBInstance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-f555zzzz

3. 输出参数

参数名称	类型	描述
AsyncRequestId	String	异步任务的请求 ID，可使用此 ID 查询异步任务的执行结果。 示例值：28920
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 销毁按量计费的云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=IsolateDBInstance
&InstanceId=cmgo-f555zzzz
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "RequestId": "6EF60BEC-0242-43AF-BB20-270359FB54A7",
    "AsyncRequestId": "28920"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.InstanceHasBeenDeleted	实例已删除。
InvalidParameterValue.InstanceHasBeenIsolated	实例已隔离。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.PrePaidInstanceUnableToIsolate	预付费实例不支持销毁。

终止数据库实例特定操作

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(KillOps)用于终止MongoDB云数据库实例上执行的特定操作。

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：KillOps。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-p8vnipr5
Operations.N	是	Array of Operation	待终止的操作

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 终止数据库实例特定操作

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=KillOps
&InstanceId=cmgo-p8vnipr5
&Operations.0.ReplicaSetName=cmgo-p8vnipr5_0
&Operations.0.NodeName=10.108.93.135:7017:39920912
&Operations.0.OpId=61
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "302530d2-ee57-495e-89d0-51e03b11815e"
  }
}
```

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。
InvalidParameterValue.NodeNotFoundInReplica	当前副本集/分片未找到该节点。

下线隔离状态的云数据库实例

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(OfflineIsolatedDBInstance)用于立即下线隔离状态的云数据库实例。进行操作的实例状态必须为隔离状态。

默认接口请求频率限制：10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：OfflineIsolatedDBInstance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-f555zzzz

3. 输出参数

参数名称	类型	描述
AsyncRequestId	String	异步任务的请求 ID，可使用此 ID 查询异步任务的执行结果。 示例值：28920
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 下线隔离状态的云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=OfflineIsolatedDBInstance
&InstanceId=cmgo-f555zzzz
&<公共请求参数>
```

输出示例

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

```
"RequestId": "6EF60BEC-0242-43AF-BB20-270359FB54A7",
"AsyncRequestId": "28920"
}
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalStatusToOffline	实例状态不允许进行下线，进行下线操作的实例状态必须为隔离状态。
InvalidParameterValue.InstanceHasBeenDeleted	实例已删除。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

修改实例名称

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(RenameInstance)用于修改云数据库实例的名称。

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：RenameInstance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-eqmoa5sf
NewName	是	String	自定义实例名称，要求为1~128 长度的任意字符。 示例值：test_name

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 重命名云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=RenameInstance
&InstanceId=cmgo-eqmoa5sf
&NewName=test_name
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8e2a339e-5ba9-4da3-8dc2-65a0a6314811"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalInstanceName	非法的实例名。
InvalidParameterValue.InstanceHasBeenDeleted	实例已删除。
InvalidParameterValue.InstanceHasBeenIsolated	实例已隔离。
InvalidParameterValue.NotFoundInstance	未找到实例。

续费云数据库实例

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(RenewDBInstance)用于续费云数据库实例，仅支持付费模式为包年包月的实例。按量计费实例不需要续费。

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：RenewDBInstances。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	是	Array of String	指定续费的一个或多个待操作的实例ID。 – 可通过 DescribeDBInstances 接口返回值中的InstanceIds获取。 – 每次续费请求的实例数量上限为100。 示例值：["cmgo-gzo0****"]
InstanceChargePrepaid	是	InstanceChargePrepaid	预付费模式，即包年包月相关参数设置。通过该参数可以指定包年包月实例的续费时长、是否设置自动续费等属性。包年包月实例该参数为必传参数。

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 续费包年包月实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=RenewDBInstances
&InstanceIds.0=cmgo-gzo03o75
&InstanceChargePrepaid.Period=1
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "RequestId": "b1886334-acfe-4445-8429-e11a6e7b3851"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.InvalidTradeOperation	计费相关错误，不允许对当前实例进行对应的新购/续费/配置变更操作。
InvalidParameterValue.NotFoundInstance	未找到实例。

重启数据库节点

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口用于重启数据库节点。

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：RestartNodes。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同。 示例值：cmgo-p8vnipr5
NodeIds.N	是	Array of String	节点Id。 示例值：["cmgo-6heje4jp_0-node-slave0","cmgo-6heje4jp_1-node-slave0"]

3. 输出参数

参数名称	类型	描述
FlowId	Integer	流程Id。 示例值：2313431
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 重启Mongod节点

输入示例

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

```
{
  "InstanceId": "cmgo-pvwp23v1",
  "NodeIds": [
    "cmgo-pvwp23v1_0-node-slave0",
    "cmgo-pvwp23v1_0-node-slave1"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2efcf780-9970-11ec-8975-6bc44e3bfad7",
    "FlowId": 19677
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

设置云数据库实例的自动备份规则

最近更新时间：2025-04-25 01:45:22

- - - - -

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(SetBackupRules)用于设置 MongoDB 云数据库的自动备份规则。

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：SetBackupRules。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID，例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-p8vnipr5
BackupMethod	是	Integer	设置自动备份方式。- 0：逻辑备份。- 1：物理备份。-3：快照备份(仅云盘版支持)。 示例值：1
BackupTime	是	Integer	设置自动备份开始时间。取值范围为：[0,23]，例如：该参数设置为2，表示02:00开始备份。 示例值：2
Notify	否	Boolean	设置自动备份发生错误时，是否发送失败告警。 - true：发送。 - false：不发送。 示例值：true
BackupRetentionPeriod	否	Integer	指定备份数据保存天数。默认为 7 天，支持设置为7、30、90、180、365。 示例值：7

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 设置实例的自动备份规则

输入示例

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

<公共请求参数>

```
{
  "InstanceId": "cmgo-abcde",
  "BackupTime": 22,
  "BackupMethod": 1,
  "BackupRetentionPeriod": 7,
  "Notify": true
}
```

输出示例

```
{
  "Response": {
    "RequestId": "58f3a6a0-2330-11ef-ad75-7139c2c3f68c"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.CheckAppIdFailed	appid校验失败。

包年包月隔离接口

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（TerminateDBInstances）可将包年包月实例退还隔离。

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：TerminateDBInstances。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定预隔离实例ID。格式如：cmgo-p8vnipr5。 示例值：cmgo-7pje****

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 包年包月实例隔离接口

输入示例

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

{
  "InstanceId": "cmgo-7pje****"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ccfe11c0-b1d3-11ea-8b51-552abe435100"
  }
}
```

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
InternalServerError	内部错误。
InternalServerError.CheckAppIdFailed	appid校验失败。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.CheckAppIdFailed	appid校验失败。
InvalidParameterValue.IllegalInstanceName	非法的实例名。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

创建云数据库实例（包年包月）

最近更新时间：2025-04-25 01:45:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(CreateDBInstance)用于创建包年包月的MongoDB云数据库实例。接口支持的售卖规格，可从查询云数据库的售卖规格（DescribeSpecInfo）获取。

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：CreateDBInstance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
NodeNum	是	Integer	– 创建副本集实例，指每个副本集内主从节点数量。每个副本集所支持的最大节点数与最小节点数，请通过接口 DescribeSpecInfo 获取。 – 创建分片集群实例，指每个分片的主从节点数量。每个分片所支持的最大节点数与最小节点数，请通过接口 DescribeSpecInfo 获取。 示例值：3
Memory	是	Integer	实例内存大小，单位：GB。具体售卖的内存规格，请通过接口 DescribeSpecInfo 获取。 示例值：4
Volume	是	Integer	实例硬盘大小，单位：GB。每一个 CPU 规格对应的最大磁盘与最小磁盘范围，请通过接口 DescribeSpecInfo 获取。 示例值：100
MongoVersion	是	String	指版本信息。具体支持的版本信息，请通过接口 DescribeSpecInfo 获取。 – MONGO_36_WT：MongoDB 3.6 WiredTiger存储引擎版本。 – MONGO_40_WT：MongoDB 4.0 WiredTiger存储引擎版本。 – MONGO_42_WT：MongoDB 4.2 WiredTiger存储引擎版本。 – MONGO_44_WT：MongoDB 4.4 WiredTiger存储引擎版本。 – MONGO_50_WT：MongoDB 5.0 WiredTiger存储引擎版本。 – MONGO_60_WT：MongoDB 6.0 WiredTiger存储引擎版本。 示例值：MONGO_44_WT
GoodsNum	是	Integer	实例数量，最小值1，最大值为10。 示例值：1
Zone	是	String	可用区信息，输入格式如：ap-guangzhou-2。 – 具体信息，请通过接口 DescribeSpecInfo 获取。 – 该参数为主可用区，如果多可用区部署，Zone必须是AvailabilityZoneList中的一个。 示例值：ap-guangzhou-2

参数名称	必选	类型	描述
Period	是	Integer	指定购买实例的购买时长。取值可选：[1,2,3,4,5,6,7,8,9,10,11,12,24,36]；单位：月。 示例值：1
MachineCode	是	String	产品规格类型。 - HIO10G：通用高HIO万兆型。 - HCD：云盘版类型。 示例值：HIO10G
ClusterType	是	String	实例架构类型。 - REPLSET：副本集。 - SHARD：分片集群。 示例值：REPLSET
ReplicateSetNum	是	Integer	- 创建副本集实例，指副本集数量，该参数只能为1。 - 创建分片集群实例，指分片数量。请通过接口 DescribeSpecInfo 查询分片数量的取值范围，其返回的数据结构SpecItems中的参数MinReplicateSetNum与MaxReplicateSetNum分别对应其最小值与最大值。 示例值：1
ProjectId	否	Integer	项目ID。 - 若不设置该参数，则为默认项目。 - 在 MongoDB 控制台项目管理 页面，可获取项目ID。 示例值：0
VpcId	否	String	私有网络ID。请登录 私有网络控制台 查询确认正确的ID。示例值：vpc-pxyzim13 示例值：vpc-0akb****
SubnetId	否	String	私有网络VPC的子网。请登录 私有网络控制台 查询子网列表确认正确的 ID。 示例值：subnet-7jbabcbe 示例值：subnet-fyrt****
Password	否	String	实例密码。设置要求如下： - 字符个数为[8,32]。 - 可输入[A,Z]、[a,z]、[0,9]范围内的字符。 - 可输入的特殊字符包括：感叹号“!”，at“@”，警号“#”、百分号“%”、插入号“^”、星号“*”、括号“()”、下划线“_”。 - 不能设置单一的字母或者数字。 示例值：****
Tags.N	否	Array of TagInfo	实例标签信息。
AutoRenewFlag	否	Integer	自动续费标记。 - 0：不自动续费。 - 1：自动续费。 示例值：1
AutoVoucher	否	Integer	是否自动选择代金券。 - 1：是。 - 0：否。默认为0。 示例值：1
Clone	否	Integer	实例类型。- 1：正式实例。- 3：只读实例。- 4：灾备实例。-5：整实例克隆，注意：克隆实例时，RestoreTime为必填项。 示例值：1
Father	否	String	父实例 ID。当参数Clone为3或者4时，即实例为只读或灾备实例时，该参数必须配置。

参数名称	必选	类型	描述
			示例值: cmgo-nh1o****
SecurityGroup.N	否	Array of String	安全组 ID。 示例值: sg-di48****
RestoreTime	否	String	克隆实例回档时间, 当Clone取值为5或6时为必填。- 若为克隆实例, 则必须配置该参数。输入格式示例: 2021-08-13 16:30:00。- 回档时间范围: 仅能回档7天内时间点的数据。 示例值: 2022-09-15 15:04:05
InstanceName	否	String	实例名称。仅支持长度为60个字符的中文、英文、数字、下划线_、分隔符-。 示例值: myInstance
AvailabilityZoneList.N	否	Array of String	若多可用区部署云数据库实例, 指定多可用区列表。 - 多可用区部署实例, 参数 Zone 指定实例主可用区信息; AvailabilityZoneList 指定所有可用区信息, 包含主可用区。输入格式如: [ap-guangzhou-2, ap-guangzhou-3, ap-guangzhou-4]。 - 通过接口 DescribeSpecInfo 可获取云数据库不同地域规划的可用区信息, 以便指定有效的可用区。 - 多可用区部署节点只能部署在3个不同可用区。不支持将集群的大多数节点部署在同一个可用区。例如: 3节点集群不支持2个节点部署在同一个区。 示例值: ["ap-guangzhou-2", "ap-guangzhou-3", "ap-guangzhou-4"]
MongosCpu	否	Integer	Mongos CPU 核数, 支持1、2、4、8、16。购买分片集群时, 必须填写。 示例值: 1
MongosMemory	否	Integer	Mongos 内存大小。 - 购买分片集群时, 必须填写。 - 单位: GB, 支持1核2GB、2核4GB、4核8GB、8核16GB、16核32GB。 示例值: 2
MongosNodeNum	否	Integer	Mongos 数量。购买分片集群时, 必须填写。 - 单可用区部署实例, 其数量范围为[3,32]。 - 多可用区部署实例, 其数量范围为[6,32]。 示例值: 3
ReadonlyNodeNum	否	Integer	只读节点数量, 取值范围[0,5]。 示例值: 0
ReadonlyNodeAvailabilityZoneList.N	否	Array of String	指只读节点所属可用区数组。跨可用区部署实例, 参数 ReadonlyNodeNum 不为0时, 必须配置该参数。 示例值: ["ap-guangzhou-2"]
HiddenZone	否	String	Hidden节点所属可用区。跨可用区部署实例, 必须配置该参数。 示例值: ap-guangzhou-2
ParamTemplateId	否	String	参数模板 ID。参数模板是一组 MongoDB 的参数并为预设置了参数值的集合, 将一组有相同诉求的参数及值 存为模板, 在创建实例时, 可直接引用参数值到新实例。合理使用参数模板, 可以提高MongoDB数据库的效率。模板列表从 DescribeDBInstanceParamTpl 接口获取, 注意模板支持的版本及实例类型。 示例值: tpl-0vswu****

3. 输出参数

参数名称	类型	描述
DealId	String	订单ID 示例值: 714****

参数名称	类型	描述
InstanceIds	Array of String	创建的实例ID列表 示例值: ["cmgo-po5f****"]
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 创建包年包月的云数据库实例

当前示例仅适合4.0版本，其余版本请严格参考传入参数中各版本说明项

输入示例

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

{
  "VpcId": "vpc-0akbol5v",
  "GoodsNum": 1,
  "Zone": "ap-guangzhou-2",
  "ProjectId": 0,
  "MachineCode": "HIO10G",
  "Period": 1,
  "ClusterType": "REPLSET",
  "Volume": 250,
  "MongoVersion": "MONGO_40_WT",
  "NodeNum": 3,
  "ReplicateSetNum": 1,
  "Memory": 4,
  "SubnetId": "subnet-fyrtjbbqw",
  "Password": "pwd123456"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d88095e5-50e8-4245-a0cf-993a536f9b20",
    "DealId": "7142863",
    "InstanceIds": [
      "cmgo-po5f8991"
    ]
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.InvalidTradeOperation	计费相关错误，不允许对当前实例进行对应的新购/续费/配置变更操作。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。
InvalidParameterValue.ProjectNotFound	项目不存在。
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.ReplicaSetNumError	副本集（分片）数量错误。
InvalidParameterValue.SecondaryNumError	从节点数错误。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.TagNotFound	未找到指定的标签。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.ZoneError	无效的可用区。
UnsupportedOperation.VersionNotSupport	当前版本不支持该操作。

创建云数据库实例（按量计费）

最近更新时间：2025-04-25 01:45:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（CreateDBInstanceHour）用于创建按量计费的MongoDB云数据库实例。

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：CreateDBInstanceHour。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Memory	是	Integer	实例内存大小，单位：GB。具体售卖的内存规格，请通过接口 DescribeSpecInfo 获取。 示例值：4
Volume	是	Integer	实例硬盘大小，单位：GB。每一个 CPU 规格对应的最大磁盘与最小磁盘范围，请通过接口 DescribeSpecInfo 获取。 示例值：100
ReplicateSetNum	是	Integer	– 创建副本集实例，指副本集数量，该参数只能为1。 – 创建分片集群实例，指分片的数量。请通过接口 DescribeSpecInfo 查询分片数量的取值范围，其返回的数据结构SpecItems中的参数MinReplicateSetNum与MaxReplicateSetNum分别对应其最小值与最大值。 示例值：1
NodeNum	是	Integer	– 创建副本集实例，指每个副本集内主从节点数量。每个副本集所支持的最大节点数与最小节点数，请通过接口 DescribeSpecInfo 获取。 – 创建分片集群实例，指每个分片的主从节点数量。每个分片所支持的最大节点数与最小节点数，请通过接口 DescribeSpecInfo 获取。 示例值：2
MongoVersion	是	String	指版本信息。具体支持的版本信息， 请通过接口 DescribeSpecInfo 获取。 – MONGO_36_WT: MongoDB 3.6 WiredTiger存储引擎版本。 – MONGO_40_WT: MongoDB 4.0 WiredTiger存储引擎版本。 – MONGO_42_WT: MongoDB 4.2 WiredTiger存储引擎版本。 – MONGO_44_WT: MongoDB 4.4 WiredTiger存储引擎版本。 – MONGO_50_WT: MongoDB 5.0 WiredTiger存储引擎版本。 – MONGO_60_WT: MongoDB 6.0 WiredTiger存储引擎版本。 示例值：MONGO_44_WT
MachineCode	是	String	产品规格类型。 – HIO10G: 通用高HIO万兆型。

参数名称	必选	类型	描述
			- HCD: 云盘版类型。 示例值: HIO10G
GoodsNum	是	Integer	实例数量, 最小值1, 最大值为10。 示例值: 1
Zone	是	String	可用区信息, 输入格式如: ap-guangzhou-2。 - 具体信息, 请通过接口 DescribeSpecInfo 获取。 - 该参数为主可用区, 如果多可用区部署, Zone必须是AvailabilityZoneList中的一个。 示例值: ap-guangzhou-3
ClusterType	是	String	实例架构类型。 - REPLSET: 副本集。 - SHARD: 分片集群。 示例值: REPLSET
VpcId	否	String	私有网络ID。请登录 私有网络控制台 查询确认正确的ID。示例值: vpc-pxyzim13 示例值: vpc-k515****
SubnetId	否	String	私有网络VPC的子网。请登录 私有网络控制台 查询子网列表确认正确的 ID。 示例值: subnet-7jbabche 示例值: subnet-60dr****
Password	否	String	实例密码。设置要求如下: - 字符个数为[8,32]。 - 可输入[A,Z]、[a,z]、[0,9]范围内的字符。 - 可输入的特殊字符包括: 感叹号“!”, at “@”, 警号“#”、百分号“%”、插入号“^”、星号“*”、括号“()”、下划线“_”。 - 不能设置单一的字母或者数字。 示例值: ****
ProjectId	否	Integer	项目ID。 - 若不设置该参数, 则为默认项目。 - 在 MongoDB 控制台项目管理 页面, 可获取项目ID。 示例值: 1
Tags.N	否	Array of TagInfo	实例标签信息。
Clone	否	Integer	实例类型。- 1: 正式实例。- 3: 只读实例。- 4: 灾备实例。-5: 克隆实例, 注意: 克隆实例RestoreTime为必填项。 示例值: 1
Father	否	String	父实例 ID。当参数Clone为3或者4时, 即实例为只读或灾备实例时, 该参数必须配置。 示例值: cmgo-nh1o****
SecurityGroup.N	否	Array of String	安全组 ID。 示例值: sg-abc****
RestoreTime	否	String	克隆实例回档时间。 - 若为克隆实例, 则必须配置该参数。输入格式示例: 2021-08-13 16:30:00。 - 回档时间范围: 仅能回档7天内时间点的数据。 示例值: 2022-09-10 15:04:05
InstanceName	否	String	实例名称。仅支持长度为60个字符的中文、英文、数字、下划线_、分隔符-。 示例值: testInstance
AvailabilityZoneList.N	否	Array of String	若多可用区部署云数据库实例, 指定多可用区列表。 - 多可用区部署实例, 参数 Zone 指定实例主可用区信息;

参数名称	必选	类型	描述
			AvailabilityZoneList 指定所有可用区信息，包含主可用区。输入格式如： [ap-guangzhou-2,ap-guangzhou-3,ap-guangzhou-4]。 - 通过接口 DescribeSpecInfo 可获取云数据库不同地域规划的可用区信息，以便指定有效的可用区。 - 多可用区部署节点只能部署在3个不同可用区。不支持将集群的大多数节点部署在同一个可用区。例如：3节点集群不支持2个节点部署在同一个区。 示例值：["ap-guangzhou-2","ap-guangzhou-3","ap-guangzhou-4"]
MongosCpu	否	Integer	Mongos CPU 核数，支持1、2、4、8、16。购买分片集群时，必须填写。 示例值：1
MongosMemory	否	Integer	Mongos 内存大小。 - 购买分片集群时，必须填写。 - 单位：GB，支持1核2GB、2核4GB、4核8GB、8核16GB、16核32GB。 示例值：2
MongosNodeNum	否	Integer	Mongos 数量。购买分片集群时，必须填写。 - 单可用区部署实例，其数量范围为[3,32]。 - 多可用区部署实例，其数量范围为[6,32]。 示例值：3
ReadonlyNodeNum	否	Integer	只读节点数量，取值范围[0,5]。 示例值：0
ReadonlyNodeAvailabilityZoneList.N	否	Array of String	指只读节点所属可用区数组。跨可用区部署实例，参数 ReadonlyNodeNum 不为0时，必须配置该参数。 示例值：["ap-guangzhou-2"]
HiddenZone	否	String	Hidden节点所属可用区。跨可用区部署实例，必须配置该参数。 示例值：ap-guangzhou-2
ParamTemplateId	否	String	参数模板 ID。参数模板是一组 MongoDB 的参数并为预设置了参数值的集合，将一组有相同诉求的参数及值 存为模板，在创建实例时，可直接引用参数值到新实例。合理使用参数模板，可以提高MongoDB数据库的效率。模板列表从 DescribeDBInstanceParamTpl 接口获取，注意模板支持的版本。 示例值：tpl-0vswu****

3. 输出参数

参数名称	类型	描述
DealId	String	订单ID。 示例值：2019070916000000350241612094****
InstanceIds	Array of String	创建的实例ID列表。 示例值：["cmgo-m26c****"]
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 创建云数据库实例（按量计费）

当前示例仅适合4.0版本，其余版本请严格参考传入参数中各版本说明项

输入示例

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

{
  "Zone": "ap-guangzhou-3",
  "GoodsNum": 1,
  "Clone": 1,
  "Memory": 4,
  "ClusterType": "REPLSET",
  "Volume": 250,
  "NodeNum": 2,
  "ReplicateSetNum": 1,
  "MachineCode": "HIO10G",
  "MongoVersion": "MONGO_50_WT"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eaf9b19d-5ad4-4ca2-9fc4-a319aeb9181f",
    "DealId": "20190709160000003502416120946732",
    "InstanceIds": [
      "cmgo-m26c****"
    ]
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.ClusterTypeError	无效的实例类型。
InvalidParameterValue.InvalidTradeOperation	计费相关错误，不允许对当前实例进行对应的新购/续费/配置变更操作。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。
InvalidParameterValue.PostPaidInstanceBeyondLimit	单个地域后付费实例数量超过限制。
InvalidParameterValue.ProjectNotFound	项目不存在。
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.ReplicaSetNumError	副本集（分片）数量错误。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。
InvalidParameterValue.TagNotFound	未找到指定的标签。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.ZoneError	无效的可用区。
UnsupportedOperation.VersionNotSupport	当前版本不支持该操作。

删除账号

最近更新时间：2025-04-25 01:45:26

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DeleteAccountUser）用于删除实例的自定义账号。

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：DeleteAccountUser。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定待删除账号的实例 ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-7pjedvz7
UserName	是	String	配置待删除的账号名。 示例值：mong_test
MongoUserPassword	是	String	配置 mongouser 对应的密码。mongouser为系统默认账号，输入其对应的密码。 示例值：Abc123...

3. 输出参数

参数名称	类型	描述
FlowId	Integer	账户删除任务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 删除实例账号

删除实例账号

输入示例

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

{
  "InstanceId": "cmgo-9d0p****",
  "UserName": "test_user",
  "MongoUserPassword": "Abc@123"
}
```

输出示例

```
{
  "Response": {
    "FlowId": 0,
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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](#)。

获取云数据库实例自动备份配置

最近更新时间：2025-04-25 01:45:26

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeBackupRules）用于获取实例自动备份配置信息。

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：DescribeBackupRules。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定实例ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-7pjedvz7

3. 输出参数

参数名称	类型	描述
BackupSaveTime	Integer	备份数据保留期限。单位为：天。 示例值：7
BackupTime	Integer	自动备份开始时间。 示例值：1
BackupMethod	Integer	备份方式。 - 0：逻辑备份。 - 1：物理备份。 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 查询实例的自动备份规则信息

查询实例的自动备份规则信息

输入示例

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

{
  "InstanceId": "cmgo-9d0p****"
}
```

输出示例

```
{
  "Response": {
    "BackupMethod": 0,
    "BackupSaveTime": 7,
    "BackupTime": 1,
    "RequestId": "07ec4290-d682-11ed-aa06-f56fb86dae78"
  }
}
```

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
InternalServerError	内部错误。

错误码	描述
InternalError.CheckAppIdFailed	appId校验失败。

查询实例客户端连接信息

最近更新时间：2025-04-25 01:45:26

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeClientConnections）用于查询实例客户端连接信息，包括连接 IP 和连接数量。

默认接口请求频率限制：5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：DescribeClientConnections。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定待查询的实例ID，例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-eqmoaxxx
Limit	否	Integer	单次请求返回的数量。最小值为1，最大值为1000，默认值为1000。 示例值：1
Offset	否	Integer	偏移量，默认值为0。Offset=Limit*(页码-1)。 示例值：1

3. 输出参数

参数名称	类型	描述
Clients	Array of ClientConnection	客户端连接信息，包括客户端 IP 和对应 IP 的连接数量。
TotalCount	Integer	满足条件的记录总条数，可用于分页查询。 示例值：2
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 查询云数据库实例的客户端连接情况

查询云数据库实例的客户端连接情况

输入示例


```
https://mongodb.tencentcloudapi.com/?Action=DescribeClientConnections
&InstanceId=cmgo-eqmoaxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "Clients": [
      {
        "Count": 1,
        "IP": "192.168.1.1",
        "InternalService": true
      }
    ],
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

错误码	描述
InvalidParameterValue.MongoVersionNotSupportQueryClient	实例版本不支持查询客户端信息。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.ProxyNotSupportQueryClient	proxy版本不支持查询客户端信息，请联系工作人员进行升级。
InvalidParameterValue.RegionNotSupportQueryClient	地域尚不支持查询客户端信息。

查询数据库实例当前正在执行的操作

最近更新时间：2025-04-25 01:45:26

1. 接口描述

接口请求域名： mongodb.tencentcloudapi.com 。

本接口（DescribeCurrentOp）用于查询云数据库实例的当前正在执行的操作。

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：DescribeCurrentOp。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定要查询的实例 ID，例如：cmgo-p8vn。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-p8vn
Ns	否	String	操作所属的命名空间 namespace，格式为 db.collection。 示例值：data.user
MillisecondRunning	否	Integer	设置查询筛选条件为操作任务已经执行的时间。 – 默认值为0，取值范围为[0, 3600000]，单位：毫秒。 – 结果将返回超过设置时间的操作。 示例值：20
Op	否	String	设置查询筛选条件为操作任务类型。取值包括：none、update、insert、query、command、getmore、remove 和 killcursors。 示例值：update
ReplicaSetName	否	String	筛选条件，分片名称。 示例值：cmgo-nh1o****_0
State	否	String	设置查询筛选条件为节点角色。 – primary：主节点。 – secondary：从节点。 示例值：primary
Limit	否	Integer	单次请求返回的数量，默认值为100，取值范围为[0,100] 示例值：10
Offset	否	Integer	偏移量，默认值为0，取值范围为[0,10000] 示例值：1
OrderBy	否	String	返回结果集排序的字段，目前支持："MicrosecsRunning"/"microsecsrunning"，默认为升序排序 示例值：MicrosecsRunning

参数名称	必选	类型	描述
OrderByType	否	String	返回结果集排序方式，可能的取值："ASC"/"asc"或"DESC"/"desc" 示例值：ASC

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合查询条件的操作总数。 示例值：19
CurrentOps	Array of CurrentOp	当前操作列表。
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 查询云数据库实例的当前操作

查询云数据库实例的当前操作

输入示例

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

{
  "InstanceId": "cmgo-p8vn****"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "CurrentOps": [
      {
        "OpId": 1804858933,
        "Ns": "db.****",
        "Query": "",
        "Op": "command",
        "ReplicaSetName": "salve-1",
        "NodeName": "salve-1",
        "Operation": "****",
        "State": "Primary",
        "MicrosecsRunning": 30353723,
        "ExecNode": "从节点"
      }
    ]
  }
}
```

```
] ,
  "RequestId": "66956cf3-2d10-4c68-87c0-7bd558deae7e"
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.QueryOutOfRange	查询offset超出范围。
InvalidParameterValue.RegionNotSupportOperation	当前地域尚不支持该操作。
InvalidParameterValue.ReplicaNotFound	未找到副本集/分片。

查询节点属性

最近更新时间：2025-04-25 01:45:25

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口用于查询节点的属性，包括节点所在可用区、节点名称、地址、角色、状态、主从延迟、优先级、投票权、标签等属性。

默认接口请求频率限制：3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：DescribeDBInstanceNodeProperty。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同。 示例值：cmgo-p8vnipr5
NodeIds.N	否	Array of String	节点ID。 示例值：["cmgo-6heje4jp_0-node-slave0","cmgo-6heje4jp_1-node-slave0"]
Roles.N	否	Array of String	节点角色。可选值包括： - PRIMARY：主节点。 - SECONDARY：从节点。 - READONLY：只读节点。 - ARBITER：仲裁节点。 示例值：PRIMARY
OnlyHidden	否	Boolean	该参数指定节点是否为Hidden节点，默认为false。 示例值：false
Priority	否	Integer	该参数指定选举新主节点的优先级。其取值范围为[0,100]，数值越高，优先级越高。 示例值：10
Votes	否	Integer	该参数指定节点投票权。 1：具有投票权。 0：无投票权。 示例值：1
Tags.N	否	Array of NodeTag	节点标签。

3. 输出参数

参数名称	类型	描述
Mongos	Array of NodeProperty	Mongos节点属性。
ReplicateSets	Array of ReplicateSetInfo	副本集节点信息。
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 查询实例节点信息

查询实例节点详细属性

输入示例

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

{
  "InstanceId": "cmgo-by909vwp"
}
```

输出示例

```
{
  "Response": {
    "Mongos": [
      {
        "Address": "10.8.16.37:30000",
        "Hidden": false,
        "NodeName": "mongos-0",
        "Priority": 0,
        "ReplicateSetId": "",
        "Role": "MONGOS",
        "SlaveDelay": 0,
        "Status": "NORMAL",
        "Tags": [
          {
            "TagKey": "role-cmgo",
            "TagValue": "primary-secondary-group"
          }
        ],
        "Votes": 0,
        "Zone": "ap-guangzhou-3",
        "WanServiceAddress": ""
      }
    ],
  }
}
```

```
"ReplicateSets": [
{
  "Nodes": [
    {
      "Address": "",
      "Hidden": false,
      "NodeName": "cmgo-by909vwp_0-node-slave0",
      "Priority": 1,
      "ReplicateSetId": "cmgo-by909vwp_0",
      "Role": "SECONDARY",
      "SlaveDelay": 0,
      "Status": "NORMAL",
      "WanServiceAddress": "",
      "Tags": [
        {
          "TagKey": "role-cmgo",
          "TagValue": "primary-secondary-group"
        }
      ],
      "Votes": 1,
      "Zone": "ap-guangzhou-3"
    }
  ],
  "Votes": 1,
  "Zone": "ap-guangzhou-3"
},
{
  "Nodes": [
    {
      "Address": "",
      "Hidden": false,
      "NodeName": "cmgo-by909vwp_1-node-slave0",
      "Priority": 1,
      "ReplicateSetId": "cmgo-by909vwp_1",
      "Role": "SECONDARY",
      "SlaveDelay": 0,
      "Status": "NORMAL",
      "WanServiceAddress": "",
      "Tags": [
        {
          "TagKey": "role-cmgo",
          "TagValue": "primary-secondary-group"
        }
      ],
      "Votes": 1,
      "Zone": "ap-guangzhou-3"
    }
  ],
  "Votes": 1,
  "Zone": "ap-guangzhou-3"
},
{
  "Address": "",
  "Hidden": false,
  "NodeName": "cmgo-by909vwp_1-node-primary",
  "Priority": 1,
  "ReplicateSetId": "cmgo-by909vwp_1",
  "Role": "PRIMARY",
```



```
"SlaveDelay": 0,
"Status": "NORMAL",
"WanServiceAddress": "",
"Tags": [
{
"TagKey": "role-cmgo",
"TagValue": "primary-secondary-group"
}
],
"Votes": 1,
"Zone": "ap-guangzhou-3"
},
{
"Address": "",
"Hidden": true,
"NodeName": "cmgo-by909vwp_1-node-slave1",
"Priority": 0,
"ReplicateSetId": "cmgo-by909vwp_1",
"Role": "SECONDARY",
"SlaveDelay": 0,
"Status": "NORMAL",
"WanServiceAddress": "",
"Tags": [
{
"TagKey": "role-cmgo",
"TagValue": "primary-secondary-group"
}
],
"Votes": 1,
"Zone": "ap-guangzhou-3"
}
]
}
],
"RequestId": "14ab088a-9d69-44b4-b39d-e4b8fbbb8f25"
}
}
```

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.KernelResponseTimeout	内核响应超时。
InternalServerError	内部错误。
InternalServerError.CheckAppldFailed	appld校验失败。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

查询云数据库实例列表

最近更新时间：2025-04-25 01:45:25

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeDBInstances）用于查询云数据库实例列表，支持通过项目ID、实例ID、实例状态等过滤条件来筛选主实例、灾备实例和只读实例信息列表。

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：DescribeDBInstances。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	否	Array of String	实例 ID 列表。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-hcscymcl
InstanceType	否	Integer	指定查询的实例类型。取值范围如下： 0：所有实例。 1：正式实例。 3：只读实例。 4：灾备实例。 示例值：0
ClusterType	否	Integer	指定所查询实例的集群类型，取值范围如下： 0：副本集实例。 1：分片实例。 -1：副本集与分片实例。 示例值：0
Status.N	否	Array of Integer	指定所查询实例的当前状态，取值范围如下所示： 0：待初始化。 1：流程处理中，例如：变更规格、参数修改等。 2：实例正常运行中。 -2：实例已过期。 示例值：2

参数名称	必选	类型	描述
VpcId	否	String	私有网络的 ID。 - 基础网络则无需配置该参数。 - 请登录 MongoDB 控制台 在实例列表中，单击私有网络名称，在私有网络页面获取其 ID。 示例值：1
SubnetId	否	String	私有网络的子网ID。 - 基础网络则无需配置该参数。 - 请登录 MongoDB 控制台 在实例列表中，单击私有网络名称，在私有网络页面获取其子网 ID。 示例值：1
PayMode	否	Integer	指定所查询实例的付费类型，取值范围如下： • 0：查询按量计费实例。 • 1：查询包年包月实例。 • -1：查询按量计费与包年包月实例。 示例值：1
Limit	否	Integer	单次请求返回的数量。默认值为20，取值范围为[1,100]。 示例值：10
Offset	否	Integer	偏移量，默认值为0。 示例值：1
OrderBy	否	String	配置返回结果排序依据的字段。目前支持依据"ProjectId"、"InstanceName"、"CreateTime"排序。 示例值：ProjectId
OrderByType	否	String	配置返回结果的排序方式。 - ASC：升序。 - DESC：降序。 示例值：ASC
ProjectIds.N	否	Array of Integer	项目 ID。请登录 MongoDB 控制台 ，在右上角的账户信息下拉菜单中，选择项目管理查询项目。 示例值：[1,2]
SearchKey	否	String	指定查询搜索的关键词。支持设置为具体的实例ID、实例名称或者内网 IP 地址。 示例值：cmgo-glh5****
Tags.N	否	Array of TagInfo	标签信息，包含标签键与标签值。

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合查询条件的实例总数。 示例值：19
InstanceDetails	Array of InstanceDetail	实例详细信息列表。
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 查询实例列表

查询实例列表

输入示例

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

{
  "Status": [
    0
  ],
  "OrderBy": "ProjectId",
  "InstanceIds": [
    "cmgo-3vgr****"
  ],
  "VpcId": "vpc-gfb3****",
  "Tags": [
    {
      "TagKey": "testTag",
      "TagValue": "testValue"
    }
  ],
  "ClusterType": 0,
  "PayMode": 0,
  "Limit": 1,
  "SearchKey": "",
  "Offset": 1,
  "SubnetId": "",
  "OrderByType": "",
  "InstanceType": 0,
  "ProjectIds": [
    1
  ]
}
```

输出示例

```
{
  "Response": {
    "InstanceDetails": [
      {
        "AutoRenewFlag": 0,
        "CloneInstances": [],
        "ClusterType": 0,
        "ClusterVer": 1,
        "ConfigServerCpuNum": 0,
        "ConfigServerMemory": 0,
        "ConfigServerNodeNum": 3,
        "ConfigServerVolume": 0,
        "CpuNum": 2,
```

```
"CreateTime": "2023-03-30 07:44:21",
"DeadLine": "2073-03-30 07:44:21",
"InstanceId": "cmgo-3vgr****",
"InstanceName": "测试实例",
"InstanceStatusDesc": "运行中",
"InstanceType": 1,
"InstanceVer": 4,
"MachineType": "HIO10G",
"MaintenanceEnd": "05:00:00",
"MaintenanceStart": "04:00:00",
"Memory": 4096,
"MongoVersion": "MONGO_42_WT",
"MongosCpuNum": 0,
"MongosMemory": 0,
"MongosNodeNum": 0,
"NetType": 1,
"PayMode": 0,
"ProjectId": 0,
"Protocol": 0,
"ReadOnlyInstances": [],
"ReadOnlyNodeNum": 0,
"RealInstanceId": "cmgo-3vgr****",
"Region": "ap-guangzhou",
"RelatedInstance": {
  "InstanceId": "",
  "Region": ""
},
"ReplicaSets": [
  {
    "Memory": 4096,
    "OplogSize": 1024,
    "RealReplicaSetId": "cmgo-3vgrapm5_*",
    "ReplicaSetId": "cmgo-3vgrapm5_*",
    "ReplicaSetName": "cmgo-3vgrapm5_*",
    "SecondaryNum": 2,
    "UsedVolume": 315,
    "Volume": 10240
  }
],
"ReplicationSetNum": 1,
"SecondaryNum": 2,
"StandbyInstances": [],
"Status": 2,
"SubnetId": "subnet-7q7d****",
"Tags": [],
"UsedVolume": 315,
"Vip": "10.5.2.12;10.5.2.14;10.5.2.10",
"Volume": 10240,
"VpcId": "vpc-gfb3****",
"Vport": 27017,
"Zone": "ap-guangzhou-2",
"ZoneList": []
```

```
}
],
"RequestId": "230c9c0e-a56f-4639-81ee-3cb46905c087",
"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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneError	无效的可用区。
LimitExceeded.TooManyRequests	请求太过频繁，触发接口频限。
UnsupportedOperation	操作不支持。

获取当前实例可修改的参数列表

最近更新时间：2025-04-25 01:45:25

1. 接口描述

接口请求域名： mongoddb.tencentcloudapi.com 。

本接口（ DescribeInstanceParams ）用于查询当前实例可修改的参数列表。

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：DescribeInstanceParams。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定待查询参数列表的实例 ID。例如：cmgo-p8vn。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-p8vn

3. 输出参数

参数名称	类型	描述
InstanceEnumParam	Array of InstanceEnumParam	参数值为枚举类型的参数集合。
InstanceIntegerParam	Array of InstanceIntegerParam	参数值为 Integer 类型的参数集合。
InstanceTextParam	Array of InstanceTextParam	参数值为 Text 类型的参数集合。
InstanceMultiParam	Array of InstanceMultiParam	参数值为混合类型的参数集合。
TotalCount	Integer	当前实例支持修改的参数数量。 示例值：2
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 InstanceEnumParam

用户请求该接口获取其mongoDB实例可配置的参数列表；

输入示例

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

```
{
  "InstanceId": "cmgo-9d0p****"
}
```

输出示例

```
{
  "Response": {
    "InstanceEnumParam": [
      {
        "CurrentValue": "off",
        "DefaultValue": "off",
        "EnumValue": [
          "off",
          "slowOp",
          "all"
        ],
        "NeedRestart": "0",
        "ParamName": "operationProfiling.mode",
        "Status": 0,
        "Tips": [
          "",
          "specifies which operations should be profiled."
        ],
        "ValueType": "enum"
      },
      {
        "CurrentValue": "true",
        "DefaultValue": "true",
        "EnumValue": [
          "true",
          "false"
        ],
        "NeedRestart": "0",
        "ParamName": "setParameter.failIndexKeyTooLong",
        "Status": 0,
        "Tips": [
          "",
          "index key length limit applies."
        ],
        "ValueType": "enum"
      }
    ]
  }
}
```

```
"InstanceIntegerParam": [
{
  "CurrentValue": "100",
  "DefaultValue": "100",
  "Max": "65536",
  "Min": "0",
  "NeedRestart": "0",
  "ParamName": "operation.profilng.slowOpThresholdMs",
  "Status": 0,
  "Tips": [
    "",
    "the slow operation time threshold, in milliseconds."
  ],
  "Unit": "",
  "ValueType": "integer"
},
{
  "CurrentValue": "600000",
  "DefaultValue": "600000",
  "Max": "2147483647",
  "Min": "1",
  "NeedRestart": "0",
  "ParamName": "setParameter.cursorTimeoutMillis",
  "Status": 0,
  "Tips": [
    "",
    "sets the expiration threshold in milliseconds for idle cursors before MongoDB removes them."
  ],
  "Unit": "",
  "ValueType": "integer"
},
{
  "CurrentValue": "33554432",
  "DefaultValue": "33554432",
  "Max": "268435456",
  "Min": "33554432",
  "NeedRestart": "0",
  "ParamName": "setParameter.internalQueryExecMaxBlockingSortBytes",
  "Status": 0,
  "Tips": [
    "",
    "internal query exec max blocking sort bytes."
  ],
  "Unit": "",
  "ValueType": "integer"
},
{
  "CurrentValue": "5",
  "DefaultValue": "5",
  "Max": "60",
  "Min": "0",
  "NeedRestart": "0",
```

```
"ParamName": "setParameter.maxTransactionLockRequestTimeoutMillis",
"Status": 0,
"Tips": [
  "",
  "the maximum amount of time in milliseconds that multi-document transactions should wait to acquire locks required by the operations in the transaction."
],
"Unit": "",
"ValueType": "integer"
},
{
  "CurrentValue": "60",
  "DefaultValue": "60",
  "Max": "300",
  "Min": "5",
  "NeedRestart": "0",
  "ParamName": "setParameter.transactionLifetimeLimitSeconds",
  "Status": 0,
  "Tips": [
    "",
    "specifies the lifetime of multi-document transactions."
  ],
  "Unit": "",
  "ValueType": "integer"
}
],
"InstanceMultiParam": [],
"InstanceTextParam": [],
"RequestId": "c07728e2-6747-4b0f-bcc9-e022036d5e28",
"TotalCount": 7
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.CurrentInstanceNotSupportModifyParams	当前实例不支持设置参数。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。
ResourceNotFound	资源不存在。

查询实例绑定的安全组

最近更新时间：2025-04-25 01:45:25

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeSecurityGroup）用于查询实例绑定的安全组。

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：DescribeSecurityGroup。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID。例如：cmgo-p8vn****。 示例值：cmgo-p8vnipr5

3. 输出参数

参数名称	类型	描述
Groups	Array of SecurityGroup	实例绑定的安全组信息。
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 查询实例安全组

查询当前实例已绑定的全部安全组

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeSecurityGroup
&InstanceId=cmgo-p8vn****
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Groups": [
      {
        "ProjectId": 0,
        "CreateTime": "2022-03-23 12:05:22",
        "Inbound": [
          {
            "Action": "Describe*",
            "CidrIp": "127.0.0.1",
            "PortRange": "27017",
            "IpProtocol": "tcp",
            "Id": "",
            "AddressModule": "",
            "ServiceModule": "",
            "Desc": "localip"
          }
        ],
        "Outbound": [
          {
            "Action": "Describe*",
            "CidrIp": "127.0.0.1",
            "PortRange": "27017",
            "IpProtocol": "tcp",
            "AddressModule": "",
            "ServiceModule": "",
            "Desc": "localip"
          }
        ],
        "SecurityGroupId": "sg-cade***",
        "SecurityGroupName": "sg",
        "SecurityGroupRemark": "sg-mark"
      }
    ],
    "RequestId": "bc59fa05-d429-4bcf-863f-1f6f99295485"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。

获取慢日志信息

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeSlowLogs）用于获取云数据库慢日志信息。接口只支持查询最近7天内慢日志。

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：DescribeSlowLogs。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-a3bm93hf
StartTime	是	Timestamp	慢日志起始时间，格式：yyyy-mm-dd hh:mm:ss，如：2019-06-01 10:00:00。查询起止时间间隔不能超过24小时，只允许查询最近7天内慢日志。 示例值：2020-04-20 00:00:00
EndTime	是	Timestamp	慢日志终止时间，格式：yyyy-mm-dd hh:mm:ss，如：2019-06-02 12:00:00。查询起止时间间隔不能超过24小时，只允许查询最近7天内慢日志。 示例值：2020-04-20 22:00:00
SlowMS	是	Integer	慢日志执行时间阈值，返回执行时间超过该阈值的慢日志，单位为毫秒(ms)，最小为100毫秒。 示例值：100
Offset	否	Integer	偏移量，最小值为0，最大值为10000，默认值为0。 示例值：0
Limit	否	Integer	分页大小，最小值为1，最大值为100，默认值为20。 示例值：100
Format	否	String	慢日志返回格式。默认返回原生慢日志格式，4.4及以上版本可设置为json。 示例值：json

3. 输出参数

参数名称	类型	描述
Count	Integer	慢日志总数 示例值：20
SlowLogs	Array of	慢日志详情

参数名称	类型	描述
	String	
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 获取实例慢日志信息

以下示例为获取一实例的慢日志信息。

输入示例

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

{
  "InstanceId": "cmgo-a1sh112****",
  "StartTime": "2020-09-22 00:00:00",
  "EndTime": "2020-09-22 00:00:00",
  "SlowMS": 1,
  "Offset": 1,
  "Limit": 1,
  "Format": "json"
}
```

输出示例

```
{
  "Response": {
    "Count": 20,
    "RequestId": "3a36aa6e-67e2-42a9-adfe-fc694334cc78",
    "SlowLogs": [
      "Mon Apr 20 13:55:21.623 I COMMAND [conn16903] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587362114, 1), signature: { hash: BinData(0, 259955995BDC7E0CA35C4276EB23341C03910A48), keyId: 6817613949303783443 } }, $configServerState: { opTime: { ts: Timestamp(1587362108, 1), t: 1 } }, $db: \"admin\" } numYields:0 res len:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 5335ms",
      "Mon Apr 20 14:01:00.309 I COMMAND [conn1313] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587362457, 1), signature: { hash: BinData(0, DC2B16681C70845D7B6C8C6C97400D4DFB394CA0), keyId: 6817613949303783443 } }, $configServerState: { opTime: { ts: Timestamp(1587362455, 1), t: 1 } }, $db: \"admin\" } numYields:0 res len:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 1264ms",
      "Mon Apr 20 14:20:15.283 I COMMAND [conn586] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587363613, 1), signature: { hash: BinData(0, 72D24884320C8A9A26AF8C4DD1127874085DFAE5), keyId: 6817613949303783443"
    ]
  }
}
```

```
} }, $configServerState: { opTime: { ts: Timestamp(1587363603, 1), t: 1 } }, $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 426ms\",  
\"Mon Apr 20 10:25:45.487 I COMMAND [conn48] command local.system.replset command: replSetReconfig { replSetReconfig: { _id: \"cmgo-a3bm93hf_1\", version: 3, protocolVersion: 1, members: [ { _id: 1, host: \"100.93.44.85:7005\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 2, host: \"10.112.91.43:7009\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 3, host: \"10.53.61.24:7004\" } ], settings: { chainingAllowed: true, heartbeatIntervalMillis: 2000, heartbeatTimeoutSecs: 10, electionTimeoutMillis: 10000, catchUpTimeoutMillis: -1, catchUpTakeoverDelayMillis: 30000, getLastErrorDefaults: { wtimeout: 0, w: 1 }, replicaSetId: ObjectId('5e9d08160c269352426f9575') } }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 3, w: 2, W: 1 } }, Database: { acquireCount: { w: 1, W: 1 } }, oplog: { acquireCount: { w: 1 } } } protocol:op_query 423ms\",  
\"Mon Apr 20 10:25:34.934 I COMMAND [conn28] command local.system.replset command: replSetReconfig { replSetReconfig: { _id: \"cmgo-a3bm93hf_1\", version: 2, protocolVersion: 1, members: [ { _id: 1, host: \"100.93.44.85:7005\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 2, host: \"10.112.91.43:7009\" } ], settings: { chainingAllowed: true, heartbeatIntervalMillis: 2000, heartbeatTimeoutSecs: 10, electionTimeoutMillis: 10000, catchUpTimeoutMillis: -1, catchUpTakeoverDelayMillis: 30000, getLastErrorDefaults: { w: 1, wtimeout: 0 }, replicaSetId: ObjectId('5e9d08160c269352426f9575') } }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 3, w: 2, W: 1 } }, Database: { acquireCount: { w: 1, W: 1 } }, oplog: { acquireCount: { w: 1 } } } protocol:op_query 414ms\",  
\"Mon Apr 20 10:24:52.278 I COMMAND [conn30] command local.system.replset command: replSetReconfig { replSetReconfig: { _id: \"cmgo-a3bm93hf_0\", version: 2, protocolVersion: 1, members: [ { _id: 1, host: \"10.53.80.6:7011\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 2, host: \"100.93.44.26:7012\" } ], settings: { chainingAllowed: true, heartbeatIntervalMillis: 2000, heartbeatTimeoutSecs: 10, electionTimeoutMillis: 10000, catchUpTimeoutMillis: -1, catchUpTakeoverDelayMillis: 30000, getLastErrorDefaults: { w: 1, wtimeout: 0 }, replicaSetId: ObjectId('5e9d07ebf3f99b58c95a33fb') } }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 3, w: 2, W: 1 } }, Database: { acquireCount: { w: 1, W: 1 } }, oplog: { acquireCount: { w: 1 } } } protocol:op_query 406ms\",  
\"Mon Apr 20 10:25:02.675 I COMMAND [conn50] command local.system.replset command: replSetReconfig { replSetReconfig: { _id: \"cmgo-a3bm93hf_0\", version: 3, protocolVersion: 1, members: [ { _id: 1, host: \"10.53.80.6:7011\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 2, host: \"100.93.44.26:7012\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 3, host: \"9.34.46.139:7003\" } ], settings: { chainingAllowed: true, heartbeatIntervalMillis: 2000, heartbeatTimeoutSecs: 10, electionTimeoutMillis: 10000, catchUpTimeoutMillis: -1, catchUpTakeoverDelayMillis: 30000, getLastErrorDefaults: { w: 1, wtimeout: 0 }, replicaSetId: ObjectId('5e9d07ebf3f99b58c95a33fb') } }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 3, w: 2, W: 1 } }, Database: { acquireCount: { w: 1, W: 1 } }, oplog: { acquireCount: { w: 1 } } } protocol:op_query 373ms\",  
\"Mon Apr 20 18:35:15.103 I COMMAND [conn586] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587378913, 1), signature: { hash: BinData(0, A3C176F81B64A7473F88F3F40B84C59EC9E1E1AE), keyId: 6817613949303783443 } }, $configServerState: { opTime: { ts: Timestamp(1587378912, 1), t: 2 } }, $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 248ms\",  
\"Mon Apr 20 10:27:38.171 I COMMAND [conn281] command admin.$cmd command: setFeatureCompatibilityVersion { setFeatureCompatibilityVersion: \"4.0\", $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 5, w: 4, R: 1 } }, Database: { acquireCount: { w: 2, W: 2 } }, Collection: { acquireCount: { w: 2 } }, MuteX: { acquireCount: { W: 1 } }, oplog: { acquireCount: { w: 2 } } } protocol:op_msg 212ms\",  
\"Mon Apr 20 10:26:27.924 I COMMAND [conn111] command local.system.replset command: replSetReconfig { replSetReconfig: { _id: \"cmgo-a3bm93hf_2\", version: 3, protocolVersion: 1, members: [ { _id: 1, host: \"10.112.90.165:7002\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 2, host: \"10.53.80.6:7012\", arbiterOnly: false, buildIndexes: true, hidden: false, priority: 1.0, slaveDelay: 0, votes: 1 }, { _id: 3, host: \"100.93.44.26:7013\" } ], settings: { chainingAllowed: true, heartbeatIntervalM
```

```
illis: 2000, heartbeatTimeoutSecs: 10, electionTimeoutMillis: 10000, catchUpTimeoutMillis: -1, catchUpTakeoverDelayMillis: 30000, getLastErrorDefaults: { w: 1, wtimeout: 0 }, replicaSetId: ObjectId('5e9d0841f0f688fc992fd087') } }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 3, w: 2, W: 1 } }, Database: { acquireCount: { w: 1, W: 1 } }, oplog: { acquireCount: { w: 1 } } } protocol:op_query 212ms\",
\"Mon Apr 20 13:55:15.051 I COMMAND [conn586] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587362114, 1), signature: { hash: BinData(0, 259955995BDC7E0CA35C4276EB23341C03910A48), keyId: 6817613949303783443 } }, $configServerState: { opTime: { ts: Timestamp(1587362085, 6), t: 1 } }, $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 192ms\",
\"Mon Apr 20 10:27:39.400 I COMMAND [conn213] command admin.$cmd command: setFeatureCompatibilityVersion { setFeatureCompatibilityVersion: \"4.0\", $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 5, w: 4, R: 1 } }, Database: { acquireCount: { w: 2, W: 2 } }, Collection: { acquireCount: { w: 2 } }, Mutex: { acquireCount: { W: 1 } }, oplog: { acquireCount: { w: 2 } } } protocol:op_msg 183ms\",
\"Mon Apr 20 10:26:09.357 I COMMAND [conn36] command local.system.replset command: replSetInitiate { replSetInitiate: { _id: \"cmgo-a3bm93hf_2\", version: 1, members: [ { _id: 1, host: \"10.112.90.165:7002\" } ] }, $readPreference: { mode: \"secondaryPreferred\" }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 23, w: 7, W: 2 }, acquireWaitCount: { W: 1 }, timeAcquiringMicros: { W: 59 } }, Database: { acquireCount: { r: 7, w: 4, W: 3 } }, Collection: { acquireCount: { r: 3, w: 2 } }, oplog: { acquireCount: { r: 2, w: 3 } } } protocol:op_query 134ms\",
\"Mon Apr 20 13:00:16.363 I COMMAND [conn16903] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587358814, 1), signature: { hash: BinData(0, 90BD7BB9B481B9825527D89A3579B52E6C0C3915), keyId: 6817613949303783443 } }, $configServerState: { opTime: { ts: Timestamp(1587358809, 2), t: 1 } }, $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 123ms\",
\"Mon Apr 20 13:35:14.983 I COMMAND [conn586] command admin.$cmd command: _flushRoutingTableCacheUpdates { forceRoutingTableRefresh: \"config.system.sessions\", maxTimeMS: 30000, $clusterTime: { clusterTime: Timestamp(1587360912, 1), signature: { hash: BinData(0, 06A84575A66B9316D1ECBFB9FB1C7B2F605AB3A8), keyId: 6817613949303783443 } }, $configServerState: { opTime: { ts: Timestamp(1587360909, 5), t: 1 } }, $db: \"admin\" } numYields:0 reslen:288 locks:{ Global: { acquireCount: { r: 4 } }, Database: { acquireCount: { r: 2 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 122ms\",
\"Mon Apr 20 10:24:43.760 I COMMAND [conn16] command local.system.replset command: replSetInitiate { replSetInitiate: { _id: \"cmgo-a3bm93hf_0\", version: 1, members: [ { _id: 1, host: \"10.53.80.6:7011\" } ] }, $readPreference: { mode: \"secondaryPreferred\" }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 23, w: 7, W: 2 }, acquireWaitCount: { W: 1 }, timeAcquiringMicros: { W: 39 } }, Database: { acquireCount: { r: 7, w: 4, W: 3 } }, Collection: { acquireCount: { r: 3, w: 2 } }, oplog: { acquireCount: { r: 2, w: 3 } } } protocol:op_query 112ms\",
\"Mon Apr 20 10:25:26.300 I COMMAND [conn15] command local.system.replset command: replSetInitiate { replSetInitiate: { _id: \"cmgo-a3bm93hf_1\", version: 1, members: [ { _id: 1, host: \"100.93.44.85:7005\" } ] }, $readPreference: { mode: \"secondaryPreferred\" }, $db: \"admin\" } numYields:0 reslen:60 locks:{ Global: { acquireCount: { r: 23, w: 7, W: 2 }, acquireWaitCount: { W: 1 }, timeAcquiringMicros: { W: 24 } }, Database: { acquireCount: { r: 7, w: 4, W: 3 }, acquireWaitCount: { W: 1 }, timeAcquiringMicros: { W: 7 } }, Collection: { acquireCount: { r: 3, w: 2 } }, oplog: { acquireCount: { r: 2, w: 3 } } } protocol:op_query 110ms\",
\"Mon Apr 20 10:27:39.400 I COMMAND [conn213] command admin.$cmd command: update { update: \"system.version\", writeConcern: { w: \"majority\" }, $db: \"admin\" } numYields:0 reslen:370 locks:{ Global: { acquireCount: { r: 5, w: 4, R: 1 } }, Database: { acquireCount: { w: 2, W: 2 } }, Collection: { acquireCount: { w: 2 } }, Mutex: { acquireCount: { W: 1 } }, oplog: { acquireCount: { w: 2 } } } protocol:op_msg 107ms\",
\"Mon Apr 20 10:27:38.065 I COMMAND [conn281] command admin.$cmd command: update { update: \"system.version\", writeConcern: { w: \"majority\" }, $db: \"admin\" } numYields:0 reslen:370 locks:{ Global: { acquireCount: { r: 2, w: 2 } }, Database: { acquireCount: { w: 1, W: 1 } }, Collection: { acquireCount: { w: 1 } }, Mutex: { acquireCount: { W: 1 } }, oplog: { acquireCount: { w: 1 } } } protocol:op_msg 107ms\",
```

```
"Mon Apr 20 10:27:38.170 I COMMAND [conn281] command admin.$cmd command: update { update: \"system.version\", writeConcern: { w: \"majority\" }, $db: \"admin\" } numYields:0 reslen:370 locks:{ Global: { acquireCount: { r: 5, w: 4, R: 1 } }, Database: { acquireCount: { w: 2, W: 2 } }, Collection: { acquireCount: { w: 2 } }, Mutex: { acquireCount: { W: 1 } }, oplog: { acquireCount: { w: 2 } } } protocol:op_msg 105ms"
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.QueryTimeOutOfRange	只能查询7天内的慢日志。
InvalidParameterValue.QueryTimeRangeBeyondLimit	只能查询7天内的慢日志。
InvalidParameterValue.SlowMSBelowLimit	slowMS参数取值范围[100,65536]。
InvalidParameterValue.StartTimeNotBeforeThanEndTime	起始时间晚于结束时间。

获取实例透明加密的开启状态

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

获取实例透明加密的开启状态

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：DescribeTransparentDataEncryptionStatus。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定实例 ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值："cmgo-jyjcw8yv"

3. 输出参数

参数名称	类型	描述
TransparentDataEncryptionStatus	String	表示是否开启了透明加密。 - close：未开启。 - open：已开启。 示例值：open或close。
KeyInfoList	Array of KMSInfoDetail	已绑定的密钥列表，如未绑定，返回null。 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 获取当前实例是否已开启数据加密

输入示例

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

```
X-TC-Action: DescribeTransparentDataEncryptionStatus
```

<公共请求参数>

```
{
  "InstanceId": "cmgo-igjy****"
}
```

输出示例

```
{
  "Response": {
    "TransparentDataEncryptionStatus": "open",
    "KeyInfoList": [
      {
        "KeyId": "65362a4b-5ec4-11ef-8467-*****",
        "KeyName": "myKey",
        "CreateTime": "2021-10-10 19:55:12",
        "Status": "Enabled",
        "KeyUsage": "TENCENT_KMS",
        "KeyOrigin": "TENCENT_KMS"
      }
    ],
    "RequestId": "59b477da-6473-4ea8-85b3-7f4473744981"
  }
}
```

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
InternalError	内部错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。

开启实例数据透明加密

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（EnableTransparentDataEncryption）用于开启云数据库 MongoDB 的透明加密能力。

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：EnableTransparentDataEncryption。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID，例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。目前支持通用版本包含：4.4、5.0，云盘版暂不支持。 示例值：cmgo-p8vnipr5
KmsRegion	是	String	密钥管理系统（Key Management Service, KMS） 服务所在地域，例如 ap-shanghai。 示例值：ap-shanghai
KeyId	否	String	密钥 ID。若不设置该参数，不指定具体的密钥 ID，由腾讯云自动生成密钥。 示例值：AKID****

3. 输出参数

参数名称	类型	描述
FlowId	Integer	开启透明加密的异步流程id，用于查询流程状态。 示例值：17316505316
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 开启实例数据透明加密

输入示例

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



```
X-TC-Action: EnableTransparentDataEncryption
<公共请求参数>

{
  "InstanceId": "cmgo-igjy****",
  "KmsRegion": "ap-guangzhou",
  "KeyId": "AKID*****"
}
```

输出示例

```
{
  "Response": {
    "FlowId": 17316505316,
    "RequestId": "a8dce7a4-d488-49c9-8d24-33cf623f5992"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。

变更配置询价

最近更新时间：2025-04-25 01:45:24

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口 (InquirePriceModifyDBInstanceSpec) 用于查询实例配置变更后的价格。

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：InquirePriceModifyDBInstanceSpec。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID，格式如：cmgo-p8vn****。与云数据库控制台页面中显示的实例ID相同。 示例值：cmgo-8ojj5631
Memory	是	Integer	变更配置后实例内存大小，单位：GB。 示例值：8
Volume	是	Integer	变更配置后实例磁盘大小，单位：GB。 示例值：200
NodeNum	否	Integer	实例节点数。默认为不变更节点数，暂不支持变更。 示例值：3
ReplicateSetNum	否	Integer	实例分片数。默认为不变更分片数，暂不支持变更。 示例值：2

3. 输出参数

参数名称	类型	描述
Price	DBInstancePrice	价格。
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 示例一

查询指定实例变配后的价格

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=InquirePriceModifyDBInstanceSpec
&InstanceId=cmgo-8oijs5631
&Memory=8
&Volume=200
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Price": {
      "DiscountPrice": 449.36,
      "OriginalPrice": 660.82,
      "UnitPrice": 0
    },
    "RequestId": "ac618a97-5a29-4003-8489-fd8a1e1cc8e9"
  }
}
```

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
InternalServerError.FindInstanceFailed	实例查询失败。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

错误码	描述
InvalidParameterValue.ZoneClosed	可用区已关闭售卖。

修改云数据库实例网络信息

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（ModifyDBInstanceNetworkAddress）用于修改云数据库实例的网络信息，支持基础网络切换为私有网络、私有网络切换私有网络。

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：ModifyDBInstanceNetworkAddress。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定需修改网络的实例 ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-7pjedvz7
OldIpExpiredTime	是	Integer	原 IP 地址保留时长。 - 单位为分钟，0表示立即回收原 IP 地址。 - 原 IP 将在约定时间后释放，在释放前原 IP和新 IP均可访问。 示例值：1
NewUniqVpcId	是	String	切换后的私有网络 ID，若实例当前为基础网络，该字段无需配置。 示例值：vpc-****
NewUniqSubnetId	是	String	切换私有网络的子网 ID。若实例当前为基础网络，该字段无需配置。 示例值：subnet-****
NetworkAddresses.N	否	Array of ModifyNetworkAddress	IP 地址信息，包含新 IP 地址与 原 IP 地址。

3. 输出参数

参数名称	类型	描述
FlowId	Integer	修改网络异步流程任务ID。 示例值：1679402995
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 修改云数据库实例网络信息

修改云数据库实例的VIP信息

输入示例

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

{
  "InstanceId": "cmgo-9d0p****",
  "OldIpExpiredTime": 1,
  "NetworkAddresses": [
    {
      "NewIPAddress": "10.0.0.10",
      "OldIpAddress": "10.0.0.1"
    }
  ],
  "NewUniqVpcId": "vpc-gfb3pqo1",
  "NewUniqSubnetId": "ins-wsdn61fw"
}
```

输出示例

```
{
  "Response": {
    "FlowId": 1679402995,
    "RequestId": "8279f6c2-db93-4b2e-931b-233d780b1a91"
  }
}
```

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.NotAllowModifyAddrAfterOpenWanService	当前实例已开启外网访问。
InternalServerError	内部错误。
InternalServerError.CheckAppIdFailed	appId校验失败。
InvalidParameter	参数错误。
InvalidParameter.InvalidVip	无效Vip信息。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。

修改实例安全组

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（ModifyDBInstanceSecurityGroup）用于修改实例绑定的安全组

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：ModifyDBInstanceSecurityGroup。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID。例如：cmgo-7pje****。 示例值：cmgo-7pjedvz7
SecurityGroupIds.N	是	Array of String	目标安全组 ID。请通过接口 DescribeSecurityGroup 查看具体的安全组 ID。 示例值：["sg-a", "sg-b"]

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 修改实例的安全组信息

修改指定实例的安全组入站规则

输入示例

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

{
```



```
"InstanceId": "cmgo-14i5dgtl1",
"SecurityGroupIds": [
  "sg-l7ipccjl"
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "bc59fa05-d429-4bcf-863f-1f6f99295485"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.SecurityGroupId	安全组ID无效。

错误码	描述
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

调整云数据库实例配置

最近更新时间：2025-05-12 01:45:58

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（ModifyDBInstanceSpec）用于调整MongoDB云数据库实例配置。接口支持的售卖规格，可从查询云数据库的售卖规格（DescribeSpecInfo）获取。

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：ModifyDBInstanceSpec。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID，例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-p8vnipr5
Memory	否	Integer	实例配置变更后的内存大小。- 单位：GB。为空时，默认取实例当前的内存大小。 注意：内存和磁盘必须同时升配或同时降配，即 Memory 与 Volume 需同时配置变更。 示例值：4
Volume	否	Integer	实例配置变更后的硬盘大小，单位：GB。为空时，默认取当前实例的磁盘大小。 - 内存和磁盘必须同时升配或同时降配，即 Memory 与 Volume 需同时配置变更。 - 降配时，变更后的磁盘容量必须大于已用磁盘容量的1.2倍。 示例值：250
NodeNum	否	Integer	实例变更后mongod的节点数（不包含readonly节点数）。 - 变更mongod CPU与内存规格时，该参数可以不配置或者输入当前 mongod(不包含readonly) 节点数量。 - 变更 mongos CPU与内存规格时，该参数可以不配置或者输入当前 mongod(不包含readonly) 节点数量。 - 节点变更时(全部类型)，该参数可不配置或输入变更后的 mongod(不包含readonly) 节点数量。 - 副本集节点数：请确认节点数量取值范围，通过云数据库的售卖规格 DescribeSpecInfo 接口返回的参数 MinNodeNum 与 MaxNodeNum 获取。 - 分片集群每个分片节点数：请确认节点数量取值范围，通过云数据库的售卖规格 DescribeSpecInfo 接口返回的参数 MinReplicateSetNodeNum 与 MaxReplicateSetNodeNum 获取。 示例值：3
ReplicateSetNum	否	Integer	实例变更后的分片数。 - 取值范围请通过云数据库的售卖规格 DescribeSpecInfo 接口返回的参数

参数名称	必选	类型	描述
			MinReplicateSetNum与MaxReplicateSetNum获取。- 该参数只能增加不能减少。 示例值：3
InMaintenance	否	Integer	实例配置变更的切换时间。 - 0：调整完成时，立即执行变配任务。默认为0。 - 1：在维护时间窗内，执行变配任务。 说明 ：调整节点数和分片数不支持在 维护时间窗内 变更。 示例值：0
MongosMemory	否	String	分片实例配置变更后的mongos内存大小。单位：GB。 示例值：2
AddNodeList.N	否	Array of AddNodeList	新增节点列表，节点类型及可用区信息。
RemoveNodeList.N	否	Array of RemoveNodeList	删除节点列表，注意：基于分片实例各片节点的一致性原则，删除分片实例节点时，只需指定0分片对应的节点即可，如：cmgo-9nl1czif_0-node-readonly0 将删除每个分片的第1个只读节点。

3. 输出参数

参数名称	类型	描述
DealId	String	订单 ID。 示例值：20221111870001224049791
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 调整云数据库实例配置

调整云数据库实例配置

输入示例

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

{
  "InstanceId": "cmgo-p8vn****",
  "Volume": 250,
  "Memory": 4
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d88095e5-50e8-4245-a0cf-993a536f9b20",
  }
```

```
"DealId": "7142863"
}
```

示例2 删除mogodb云数据库实例从节点

广州二区和三区共删除2个从节点，最终节点数量变更为3

输入示例

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

{
  "InstanceId": "cmgo-abcdef",
  "Memory": 4,
  "Volume": 100,
  "OplogSize": 30,
  "NodeNum": 3,
  "ReplicateSetNum": 1,
  "InMaintenance": 1,
  "RemoveNodeList": [
    {
      "NodeName": "cmgo-3iecwbfx_0-node0",
      "Role": "SECONDARY",
      "Zone": "ap-guangzhou-2"
    },
    {
      "NodeName": "cmgo-3iecwbfx_0-node1",
      "Role": "SECONDARY",
      "Zone": "ap-guangzhou-3"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "DealId": "7142863",
    "RequestId": "d88095e5-50e8-4245-a0cf-993a536f9b20"
  }
}
```

示例3 新增mogodb云数据库实例从节点

广州二区和三区共新增2个从节点，最终节点数量变更为5

输入示例

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

```
{
  "InstanceId": "cmgo-abcdef",
  "Memory": 4,
  "Volume": 100,
  "OplogSize": 30,
  "NodeNum": 5,
  "ReplicateSetNum": 1,
  "InMaintenance": 1,
  "AddNodeList": [
    {
      "Role": "SECONDARY",
      "Zone": "ap-guangzhou-2"
    },
    {
      "Role": "SECONDARY",
      "Zone": "ap-guangzhou-3"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "DealId": "7142863",
    "RequestId": "d88095e5-50e8-4245-a0cf-993a536f9b20"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameter.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.InvalidTradeOperation	计费相关错误，不允许对当前实例进行对应的新购/续费/配置变更操作。
InvalidParameterValue.ModifyModeError	内存和磁盘必须同时升配或降配。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.OplogSizeOutOfRange	OplogSize参数设置错误，应介于磁盘容量的10%和90%之间。
InvalidParameterValue.SetDiskLessThanUsed	设置的磁盘大小不得低于已用磁盘的1.2倍。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

修改mongodb实例参数

最近更新时间：2025-04-25 01:45:23

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（ModifyInstanceParams）用于修改mongodb实例的参数配置。

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：ModifyInstanceParams。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定实例 ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值："cmgo-jyjcw8yv"
InstanceParams.N	是	Array of ModifyMongoDBParamType	指定需修改的参数名及值。当前所支持的参数名及对应取值范围，请通过 DescribeInstanceParams 获取。
ModifyType	否	String	操作类型，包括： - IMMEDIATELY：立即调整。 - DELAY：延迟调整。可选字段，不配置该参数则默认为立即调整。 示例值：IMMEDIATELY

3. 输出参数

参数名称	类型	描述
Changed	Boolean	修改参数配置是否生效。 - true：参数修改后的值已生效。 - false：执行失败。 示例值：true
TaskId	Integer	该参数暂时无意义(兼容前端保留)。 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 修改mongoDB实例参数

用户发起修改其mongoDB的参数配置

输入示例

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

{
  "InstanceId": "cmgo-9d0p6umb",
  "InstanceParams": [
    {
      "Value": "operation.profilng.slowOpThresholdMs",
      "Key": "operation.profilng.slowOpThresholdMs"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Changed": true,
    "RequestId": "e51b4590-db7b-4835-9899-1f5c33f92fbf",
    "TaskId": 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
FailedOperation	操作失败。
FailedOperation.OperationNotAllowedInInstanceLocking	实例锁定中不允许操作。
InternalError	内部错误。
InvalidParameter.ModifyMongodbParams	参数名有误
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.ModifyMongodbParams	参数值有误。
InvalidParameterValue.NotFoundInstance	未找到实例。
ResourceNotFound	资源不存在。

实例维护时间设置

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（SetInstanceMaintenance）用于设置实例维护时间窗。

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：SetInstanceMaintenance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定实例ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-7pjedvz7
MaintenanceStart	是	String	维护时间窗开始时间。取值范围为"00:00-23:00"的任意整点或半点，如00:00或00:30。 示例值："04:00"
MaintenanceEnd	是	String	维护时间窗结束时间。 - 取值范围为"00:00-23:00"的任意整点或半点，维护时间持续时长最小为30分钟，最大为3小时。 - 结束时间务必是基于开始时间向后的时间。 示例值："05:00"

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 修改实例维护时间窗口

修改实例维护时间窗口

输入示例

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

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

{
  "InstanceId": "cmgo-9d0p****",
  "MaintenanceEnd": "21:00",
  "MaintenanceStart": "20:00"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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
InternalServerError	内部错误。
InternalServerError.CheckAppIdFailed	appId校验失败。
InvalidParameter	参数错误。

查询实例慢日志详情

最近更新时间：2025-06-30 02:05:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

查询实例慢日志详情

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：DescribeDetailedSlowLogs。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例id 示例值：cmgo-xxxx
StartTime	是	String	待查询慢日志的开始时间 示例值：2025-01-02 13:14:15
EndTime	是	String	待慢日志的结束时间 示例值：2025-01-02 13:14:15
ExecTime	否	Integer	过滤执行时间大于此值的慢日志，单位ms，默认值100 示例值：100
Commands.N	否	Array of String	过滤慢日志的命令类型 示例值：find
Texts.N	否	Array of String	全文搜索关键字，多个关键字间为或关系 示例值：["aaa","bbb"]
NodeNames.N	否	Array of String	根据节点名过滤 示例值：["cmgo-xxxx-node-primary"]
QueryHash.N	否	Array of String	根据queryHash过滤 示例值：["abcdef"]
Offset	否	Integer	分页偏移量 示例值：1
Limit	否	Integer	返回条数 示例值：10
OrderBy	否	String	排序条件，只支持StartTime(按慢日志生成时间)和ExecTime(慢日志执行时间) 示例值：ExecTime

参数名称	必选	类型	描述
OrderByType	否	String	排序。desc倒排，asc正排 示例值：desc

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	满足条件的慢日志数量 示例值：100
DetailedSlowLogs	Array of SlowLogItem	慢日志详情
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 查询慢日志详情

输入示例

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

{
  "InstanceId": "cmgo-qny2****",
  "StartTime": "2025-03-27 12:00:00",
  "EndTime": "2025-03-27 13:33:00"
}
```

输出示例

```
{
  "Response": {
    "DetailedSlowLogs": [
      {
        "Log": "xxxxxxx",
        "NodeName": "cmgo-xxxx_0-node-primary",
        "QueryHash": "BA0DFF88"
      }
    ],
    "RequestId": "a1460a2d-dc8e-467c-b2d5-0beb6b84a7a7",
    "TotalCount": 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
InternalServerError	内部错误。
InvalidParameterValue.LimitParaOutOfRange	limit取值范围[1,100]。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.OffsetParaOutOfRange	offset取值范围[0, 10000]。
InvalidParameterValue.QueryTimeRangeBeyondLimit	只能查询7天内的慢日志。
InvalidParameterValue.SlowMSBelowLimit	slowMS参数取值范围[100,65536]。
InvalidParameterValue.StartTimeNotBeforeThanEndTime	起始时间晚于结束时间。
InvalidParameterValue.TimeFormatErr	非法的时间格式。

备份相关接口

创建备份下载任务

最近更新时间：2025-04-25 01:45:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口用来创建某个备份文件的下载任务

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：CreateBackupDownloadTask。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同。 示例值：cmgo-f555zzzz
BackupName	是	String	要下载的备份文件名，可通过DescribeDBBackups接口获取。 示例值：cmgo-74sjf5mt_2021-03-24 15:40
BackupSets.N	是	Array of ReplicaSetInfo	指定要下载的副本集的节点名称 或 分片集群的分片名称列表。 如副本集cmgo-p8vnipr5，示例(固定取值)：BackupSets.0=cmgo-p8vnipr5_0，可下载全量数据。 如分片集群cmgo-p8vnipr5，示例：BackupSets.0=cmgo-p8vnipr5_0&BackupSets.1=cmgo-p8vnipr5_1，即下载分片0和分片1的数据，分片集群如需全量下载，请按示例方式传入全部分片名称。

3. 输出参数

参数名称	类型	描述
Tasks	Array of BackupDownloadTaskStatus	下载任务状态
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 创建备份下载任务

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=CreateBackupDownloadTask
&InstanceId=cmgo-dygv1rnp
&BackupName=cmgo-dygv1rnp_2021-03-26_10:44
&BackupSets.0.ReplicaSetId=cmgo-dygv1rnp_0
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "6ed8d110-67be-4650-83b0-5df5a897e9e6",
    "Tasks": [
      {
        "ReplicaSetId": "cmgo-dygv1rnp_0",
        "Status": 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
InternalServerError	内部错误。

错误码	描述
InternalError.CheckAppIdFailed	appId校验失败。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.BackupFileNotFound	备份文件未找到。
InvalidParameterValue.NotFoundInstance	未找到实例。

查询备份下载任务信息

最近更新时间：2025-04-25 01:45:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

查询备份下载任务信息

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：DescribeBackupDownloadTask。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-74sjf5mt
BackupName	否	String	备份文件名，用来过滤指定文件的下载任务 示例值：cmgo-74sjf5mt_2021-03-24 15:40
StartTime	否	String	指定查询时间范围内的任务，StartTime指定开始时间，不填默认不限制开始时间 示例值：2021-01-02 15:04:05
EndTime	否	String	指定查询时间范围内的任务，EndTime指定截止时间，不填默认不限制截止时间 示例值：2021-01-02 15:04:05
Limit	否	Integer	此次查询返回的条数，取值范围为1-100，默认为20 示例值：10
Offset	否	Integer	指定此次查询返回的页数，默认为0 示例值：0
OrderBy	否	String	排序字段，取值为createTime，finishTime两种，默认为createTime 示例值：finishTime
OrderByType	否	String	排序方式，取值为asc，desc两种，默认desc 示例值：desc
Status.N	否	Array of Integer	根据任务状态过滤。0-等待执行，1-正在下载，2-下载完成，3-下载失败，4-等待重试。不填默认返回所有类型 示例值：[2]

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	满足查询条件的所有条数 示例值：12
Tasks	Array of BackupDownloadTask	下载任务列表
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 查询备份下载任务

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeBackupDownloadTask
&InstanceId=cmgo-dygv1rnp
&Status.0=2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "c41c0c46-7d5a-49b9-832a-557880c1e9ef",
    "Tasks": [
      {
        "BackupName": "cmgo-dygv1rnp_2021-03-26 10:44",
        "BackupSize": 844,
        "CreateTime": "2021-03-26 10:47:37",
        "Percent": 100,
        "ReplicaSetId": "cmgo-dygv1rnp_0",
        "Status": 2,
        "TimeSpend": 0,
        "Url": "https://mogodb-backup-test-tar-1251937656.cos.ap-guangzhou.myqcloud.com/cmgo-dygv1rnp_2021-03-26%2010%
3A44%2Fcmgo-dygv1rnp_0.tar?q-sign-algorithm=sha1&q-ak=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&q-sign-time=1616726859%3B1
616813259&q-key-time=1616726859%3B1616813259&q-header-list=host&q-url-param-list=&q-signature=xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx"
      }
    ],
    "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。
InternalServerError	内部错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.BackupFileNotFound	备份文件未找到。
InvalidParameterValue.NotFoundInstance	未找到实例。

查询实例备份列表

最近更新时间：2025-04-25 01:45:27

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeDBBackups）用于查询实例备份列表，目前只支持查询7天内的备份记录。

默认接口请求频率限制：150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：DescribeDBBackups。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-f555zzzz
BackupMethod	否	Integer	备份方式，当前支持：0-逻辑备份，1-物理备份，2-所有备份。默认为逻辑备份。 示例值：1
Limit	否	Integer	分页大小，最大值为100，不设置默认查询所有。 示例值：10
Offset	否	Integer	分页偏移量，最小值为0，默认值为0。 示例值：1

3. 输出参数

参数名称	类型	描述
BackupList	Array of BackupInfo	备份列表
TotalCount	Integer	备份总数 示例值：0
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 查询云数据库实例的备份文件信息

输入示例

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

{
  "InstanceId": "cmgo-f555zzzz",
  "BackupMethod": 1
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6EF60BEC-0242-43AF-BB20-270359FB54A7",
    "BackupList": [],
    "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
InvalidParameter	参数错误。

错误码	描述
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。

备份实例接口

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（CreateBackupDBInstance）用于备份实例。

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：CreateBackupDBInstance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-ayxpky6l
BackupMethod	是	Integer	设置备份方式。 - 0：逻辑备份。 - 1：物理备份。 示例值：0
BackupRemark	否	String	备份备注信息。 示例值：remarkInfo

3. 输出参数

参数名称	类型	描述
AsyncRequestId	String	查询备份流程的状态。 示例值：4773
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 备份实例接口

备份实例接口

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=CreateBackupDBInstance
&InstanceId=cmgo-ayxpy6l
&BackupRemark=remarkInfo
&BackupMethod=0
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "AsyncRequestId": "1680780049",
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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
InternalServerError	内部错误。
InternalServerError.CheckAppIdFailed	appId校验失败。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.CheckAppIdFailed	appId校验失败。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。

错误码	描述
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

账号相关接口

创建账号

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名： `mongodb.tencentcloudapi.com` 。

本接口（`CreateAccountUser`）用于自定义实例访问账号。

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：CreateAccountUser。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-7pjedvz7
UserName	是	String	新账号名称。其格式要求如下： - 字符范围[1,32]。 - 可输入[A,Z]、[a,z]、[1,9]范围的字符以及下划线 “_” 与短划线 “-” 。 示例值：mong_test
Password	是	String	新账号密码。密码复杂度要求如下： - 字符长度范围[8,32]。 - 至少包含字母、数字和特殊字符（叹号 “!” 、at"@”、井号 “#” 、百分号 “%” 、插入符 “^” 、星号 “*” 、小括号 “()” 、下划线 “_” ）中的两种。 示例值：abc123!@
MongoUserPassword	是	String	mongouser 账号对应的密码。mongouser 为系统默认账号，即为创建实例时，设置的密码。 示例值：Abc123...
UserDesc	否	String	账号备注信息。 示例值：测试账号
AuthRole.N	否	Array of Auth	账号的读写权限信息。

3. 输出参数

参数名称	类型	描述
FlowId	Integer	创建任务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 自定义账号

自定义实例账号

输入示例

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

{
  "InstanceId": "cmgo-9d0p****",
  "UserName": "test_user",
  "Password": "Abc@123...",
  "MongoUserPassword": "Abc@123...",
  "UserDesc": "测试用户",
  "AuthRole": [
    {
      "Mask": 0,
      "NameSpace": "collection"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "FlowId": 1680767468,
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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
InternalServerError	内部错误。
InternalServerError.CheckAppIdFailed	appId校验失败。
InternalServerError.PasswordError	password与原先记录的password不同。
InvalidParameter	参数错误。
InvalidParameterValue.CheckAppIdFailed	appId校验失败。

全部账号列表

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeAccountUsers）用于获取当前实例的全部账号。

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：DescribeAccountUsers。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定待获取账号的实例ID。例如：cmgo-p8vn。请登录 MongoDB 控制台 在实例列表复制实例 ID。 示例值：cmgo-**

3. 输出参数

参数名称	类型	描述
Users	Array of UserInfo	实例账号列表。
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 查询当前实例的全部账户信息

查询当前实例的全部账户信息

输入示例

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

{
```

```
"InstanceId": "cmgo-9d0p****"
}
```

输出示例

```
{
  "Response": {
    "Users": [
      {
        "UserName": "test_user",
        "AuthRole": [
          {
            "Mask": 0,
            "NameSpace": "collection"
          }
        ],
        "CreateTime": "2022-03-04 19:00:00",
        "UpdateTime": "2022-03-04 19:00:00",
        "UserDesc": "测试账号"
      }
    ],
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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
InternalServerError	内部错误。
InternalServerError.CheckAppldFailed	appld校验失败。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

修改实例用户的密码

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

修改实例用户的密码

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：ResetDBInstancePassword。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例Id 示例值：cmgo-7pjedvz7
UserName	是	String	实例账号名 示例值：rwuser
Password	是	String	新密码，新密码长度不能少于8位 示例值：111111

3. 输出参数

参数名称	类型	描述
AsyncRequestId	String	异步请求Id，用户查询该流程的运行状态 示例值：4769
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 修改实例用户的密码

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=ResetDBInstancePassword
&InstanceId=cmgo-7pjedvz7
&UserName=rwuser
```

```
&Password=111111
```

```
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "AsyncRequestId": "4769",
    "RequestId": "e600a8d0-9014-11ea-84c1-01ccde830c6"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。

错误码	描述
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。
InvalidParameterValue.UserNotFound	用户账户不存在。

设置账户权限

最近更新时间：2025-04-25 01:45:21

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（SetAccountUserPrivilege）用于设置实例的账号权限。

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：SetAccountUserPrivilege。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	指定待设置账号的实例ID。例如：cmgo-p8vn****。请登录 MongoDB 控制台 在实例列表复制实例ID。 示例值：cmgo-kbcfl64r
UserName	是	String	设置账号名称。 示例值：bob
AuthRole.N	是	Array of Auth	设置权限信息。 示例值：11

3. 输出参数

参数名称	类型	描述
FlowId	Integer	任务ID。 示例值：100000
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 为当前实例设置账号权限

为当前实例的指定账号设置操作权限

输入示例

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

<公共请求参数>

```
{
  "InstanceId": "cmgo-9d0p6umb",
  "UserName": "test_user",
  "AuthRole": [
    {
      "Mask": 0,
      "NameSpace": "eg_collection"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "FlowId": 1680767468,
    "RequestId": "1df930fb-eb7e-4e3f-bcab-15a1aa5fede0"
  }
}
```

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
InternalError	内部错误。
InternalError.CheckAppldFailed	appld校验失败。
InvalidParameter	参数错误。

任务相关接口

查询异步任务状态接口

最近更新时间：2025-04-25 01:45:29

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（DescribeAsyncRequestInfo）用于查询异步任务状态接口。

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：DescribeAsyncRequestInfo。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AsyncRequestId	是	String	异步请求Id，涉及到异步流程的接口返回，如CreateBackupDBInstance 示例值：4762

3. 输出参数

参数名称	类型	描述
Status	String	状态。返回参数有：initial-初始化、running-运行中、paused-任务执行失败，已暂停、undoeed-任务执行失败，已回滚、failed-任务执行失败，已终止、success-成功 示例值：success
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 查询异步任务状态接口

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeAsyncRequestInfo
&AsyncRequestId=4762
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "RequestId": "e600a8d0-9014-11ea-84c1-01cccd830c6",
    "Status": "success"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

按key回档相关接口

按 Key 回档

最近更新时间：2025-04-25 01:45:22

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

该接口用于发起按 Key 闪回任务，依据数据的闪回 Key（默认为 id）对数据进行极速回档，快速恢复业务。

说明：按 Key 闪回于2023年09月11日正式进行公测，在此期间，该接口仅对公测用户开放。

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：FlashBackDBInstance。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	开启按 Key 回档的实例 ID。 示例值：cmgo-iga0****
TargetFlashbackTime	是	String	源数据想恢复到的时间。 示例值：2023-08-04 14:00:00
TargetDatabases.N	是	Array of FlashbackDatabase	源数据所在的库表信息。 示例值：[{"DBName":"mydb","Collections":[{"CollectionName":"my_collection","KeyValues":[{"Key":"_id","Value":"*****"}]}],
TargetInstanceId	否	String	数据最终写入的实例 ID。 示例值：cmgo-iga0****

3. 输出参数

参数名称	类型	描述
FlowId	Integer	回档数据异步任务 ID。 示例值：16033****
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 执行按key回档

输入示例

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

{
  "InstanceId": "cmgo-igjysqkr",
  "TargetFlashbackTime": "2023-08-04 14:00:00",
  "TargetDatabases": [
    {
      "DBName": "my_test",
      "Collections": [
        {
          "CollectionName": "my_test1",
          "KeyValues": [
            {
              "Key": "name",
              "Value": "name-demo"
            }
          ],
          "TargetResultCollectionName": "my_test_bak0804143230",
          "FilterKey": "name"
        }
      ]
    }
  ],
  "TargetInstanceId": "cmgo-0517****"
}
```

输出示例

```
{
  "Response": {
    "FlowId": 1234567,
    "RequestId": "59b477da-6473-4ea8-85b3-7f4473744****"
  }
}
```

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.FlashbackByKeyNotOpen	按key回档未开启
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.ParametersNil	接口参数为空。
InvalidParameterValue.NotFoundInstance	未找到实例。
UnauthorizedOperation.NoAccess	没有访问权限
UnsupportedOperation.KernelVersionNotSupport	内核版本不支持。
UnsupportedOperation.VersionNotSupport	当前版本不支持该操作。

其他接口

创建数据库参数模板

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(CreateDBInstanceParamTpl)用于创建云数据库MongoDB实例的参数模板
说明：CreateDBInstanceParamTpl API正在公测中，在此期间，该接口仅对公测用户开放

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：CreateDBInstanceParamTpl。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	否	String	公共参数 ，本接口不需要传递此参数。
TplName	是	String	参数模板名称。 示例值：高性能模板
MongoVersion	否	String	版本号，该参数模板支持的售卖版本请参照查询云数据库的售卖规格（DescribeSpecInfo）返回结果。参数与版本对应关系是：MONGO_36_WT：MongoDB 3.6 WiredTiger存储引擎版本，MONGO_40_WT：MongoDB 4.0 WiredTiger存储引擎版本，MONGO_42_WT：MongoDB 4.2 WiredTiger存储引擎版本。当MirrorTplId为空时，该字段必填。 示例值：MONGO_42_WT
ClusterType	否	String	实例类型，REPLSET-副本集，SHARD-分片集群，STANDALONE-单节点 当MirrorTplId为空时，该字段必填。 示例值：SHARD
TplDesc	否	String	模板描述信息。 示例值：用于压测场景
Params.N	否	Array of ParamType	模板参数，若为空，则以系统默认模板作为新版本参数。
MirrorTplId	否	String	镜像模板ID，若该字段不为空，则以该模板为镜像，克隆出一个新的模板。注意：MirrorTplId不为空时，MongoVersion及ClusterType将以MirrorTpl模板的版本及实例类型为准。 示例值：tpl-sdah144hg

3. 输出参数

参数名称	类型	描述
TplId	String	模板ID 示例值: tpl-sdah144hg
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 创建参数模板

输入示例

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

{
  "ClusterType": "SHARD",
  "MongoVersion": "MONGO_44_WT",
  "TplDesc": "",
  "TplName": "test-4.4"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "f3924f2d-6b76-4fe3-b705-1e0144049b90",
    "TplId": "tpl-bi5f78srm"
  }
}
```

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](#)。

查询实例参数模板列表

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeDBInstanceParamTpl)用于查询当前账号下所有MongoDB数据库参数模板

说明：DescribeDBInstanceParamTpl API正在公测中，在此期间，该接口仅对公测用户开放

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：DescribeDBInstanceParamTpl。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	否	String	公共参数 ，本接口不需要传递此参数。
TplIds.N	否	Array of String	参数模板 ID 查询条件。 示例值：["tpl-sdjf143qu"]
TplNames.N	否	Array of String	模板名称，查询条件。 示例值：["test"]
MongoVersion.N	否	Array of String	根据版本号查询参数模板，具体支持的售卖版本，请参见 DescribeSpecInfo 查询云数据库的售卖规格的返回结果。参数与版本对应关系如下所示： – MONGO_36_WT：MongoDB 3.6 WiredTiger存储引擎版本。 – MONGO_40_WT：MongoDB 4.0 WiredTiger存储引擎版本。 – MONGO_42_WT：MongoDB 4.2 WiredTiger存储引擎版本。 示例值：["MONGO_42_WT"]
TplType	否	String	根据模板类型查询参数模板，支持DEFAULT（默认模板）和CUSTOMIZE（自定义模板）两种。 示例值：CUSTOMIZE

3. 输出参数

参数名称	类型	描述
ParamTpls	Array of ParamTpl	参数模板列表信息。
TotalCount	Integer	参数模板总数。 示例值：0
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 根据模板ID查询参数模板

输入示例

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

{
  "TplIds": [
    "tpl-eykkk2khs"
  ]
}
```

输出示例

```
{
  "Response": {
    "ParamTpls": [
      {
        "ClusterType": "REPLSET",
        "MongoVersion": "MONGO_42_WT",
        "TplDesc": "System predefined, forbid modify and delete.",
        "TplId": "tpl-eykkk2khs",
        "TplName": "default parameter template",
        "TplType": "DEFAULT"
      }
    ],
    "RequestId": "7878d7d7-8c80-4050-93a1-5e1785effd1a",
    "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
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

参数模板详情

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeDBInstanceParamTplDetail)用于查询MongoDB云数据库实例的参数模板详情。

说明：DescribeDBInstanceParamTplDetail API正在公测中，在此期间，该接口仅对公测用户开放

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：DescribeDBInstanceParamTplDetail。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	否	String	公共参数 ，本接口不需要传递此参数。
TplId	是	String	参数模板 ID。 示例值：tpl-sdvj2jh4k
ParamName	否	String	参数名称，传入该值，则只会获取该字段的参数详情。为空时，返回全部参数。 示例值：operationProfiling.mode

3. 输出参数

参数名称	类型	描述
InstanceEnumParams	Array of InstanceEnumParam	枚举类参数详情列表。
InstanceIntegerParams	Array of InstanceIntegerParam	整形参数详情列表。
InstanceTextParams	Array of InstanceTextParam	文本参数详情列表。
InstanceMultiParams	Array of InstanceMultiParam	多值参数详情列表。
TotalCount	Integer	参数总个数。 示例值：0
MongoVersion	String	模板适配实例版本。 示例值：MONGO_50_WT
ClusterType	String	模板适配集群类型，副本集或分片。。 示例值：SHARD

参数名称	类型	描述
TplName	String	参数模板名称。 示例值：my_tpl_demo
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 查询参数模板详情

输入示例

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

{
  "TplId": "tpl-2284g3nmw"
}
```

输出示例

```
{
  "Response": {
    "ClusterType": "REPLSET",
    "InstanceEnumParams": [
      {
        "CurrentValue": "off",
        "DefaultValue": "off",
        "EnumValue": [
          "off",
          "slowOp",
          "all"
        ],
        "NeedRestart": "0",
        "ParamName": "operationProfiling.mode",
        "Status": 1,
        "Tips": [
          "",
          "specifies which operations should be profiled."
        ],
        "ValueType": "enum"
      },
      {
        "CurrentValue": "snappy",
        "DefaultValue": "snappy",
        "EnumValue": [
```

```
"snappy",
"zlib",
"zstd",
"none"
],
"NeedRestart": "1",
"ParamName": "storage.wiredTiger.collectionConfig.blockCompressor",
"Status": 1,
"Tips": [
  "",
  "you can uses zstd for journal compression and data compression."
],
"ValueType": "enum"
},
],
"InstanceIntegerParams": [
  {
    "CurrentValue": "100",
    "DefaultValue": "100",
    "Max": "65536",
    "Min": "0",
    "NeedRestart": "0",
    "ParamName": "operation.profilng.slowOpThresholdMs",
    "Status": 1,
    "Tips": [
      "",
      "the slow operation time threshold, in milliseconds."
    ],
    "Unit": "",
    "ValueType": "integer"
  },
  {
    "CurrentValue": "600000",
    "DefaultValue": "600000",
    "Max": "2147483647",
    "Min": "1",
    "NeedRestart": "0",
    "ParamName": "setParameter.cursorTimeoutMillis",
    "Status": 1,
    "Tips": [
      "",
      "sets the expiration threshold in milliseconds for idle cursors before MongoDB removes them."
    ],
    "Unit": "",
    "ValueType": "integer"
  },
  {
    "CurrentValue": "33554432",
    "DefaultValue": "33554432",
    "Max": "268435456",
    "Min": "33554432",
    "NeedRestart": "0",
```

```
"ParamName": "setParameter.internalQueryMaxBlockingSortMemoryUsageBytes",
"Status": 1,
"Tips": [
  "",
  "internal query exec max blocking sort bytes."
],
"Unit": "",
"ValueType": "integer"
},
{
  "CurrentValue": "5",
  "DefaultValue": "5",
  "Max": "60",
  "Min": "0",
  "NeedRestart": "0",
  "ParamName": "setParameter.maxTransactionLockRequestTimeoutMillis",
  "Status": 1,
  "Tips": [
    "",
    "the maximum amount of time in milliseconds that multi-document transactions should wait to acquire locks required by the operations in the transaction."
  ],
  "Unit": "",
  "ValueType": "integer"
},
{
  "CurrentValue": "60",
  "DefaultValue": "60",
  "Max": "300",
  "Min": "5",
  "NeedRestart": "0",
  "ParamName": "setParameter.transactionLifetimeLimitSeconds",
  "Status": 1,
  "Tips": [
    "",
    "specifies the lifetime of multi-document transactions."
  ],
  "Unit": "",
  "ValueType": "integer"
}
],
"InstanceMultiParams": [],
"InstanceTextParams": [],
"MongoVersion": "MONGO_50_WT",
"RequestId": "5dedc31c-ee80-4a12-9ed3-6003504b5a57",
"TotalCount": 7,
"TplName": "demo"
}
}
```

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
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

删除参数模板

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DropDBInstanceParamTpl)用于删除云数据库MongoDB实例的参数模板

说明：DropDBInstanceParamTpl API正在公测中，在此期间，该接口仅对公测用户开放

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：DropDBInstanceParamTpl。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	否	String	公共参数 ，本接口不需要传递此参数。
TplId	是	String	参数模板 ID。 示例值：tpl-djtuhy1d

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 删除参数模板

输入示例

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

{
  "TplId": "tpl-b211n0886"
}
```


输出示例

```
{
  "Response": {
    "RequestId": "33bb57b9-4090-40d8-a608-22299973063f"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。

刷新路由配置

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

在所有mongos上执行FlushRouterConfig命令

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：FlushInstanceRouterConfig。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：cmgo-a3bm9xxx

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 在所有Mongos上执行FlushRouterConfig

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=FlushInstanceRouterConfig
&InstanceId=cmgo-a3bm93hf
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0ad9f471-fb8f-402a-aa78-b1640c6670c4"
```

```
}
}
```

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
InternalServerError	内部错误。
InternalServerError.CheckAppIdFailed	appId校验失败。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue	参数取值错误。

修改数据库参数模板

最近更新时间：2025-04-25 01:45:28

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(ModifyDBInstanceParamTpl)用于修改MongoDB云数据库实例的参数模板。

说明：ModifyDBInstanceParamTpl API正在公测中，在此期间，该接口仅对公测用户开放

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：ModifyDBInstanceParamTpl。
Version	是	String	公共参数 ，本接口取值：2019-07-25。
Region	否	String	公共参数 ，本接口不需要传递此参数。
TplId	是	String	待修改的参数模板 ID，示例：tpl-jglr91vew。 示例值：tpl-sdh134udi
TplName	否	String	待修改参数模板名称，为空时，保持原有名称。 示例值：压测模板
TplDesc	否	String	待修改参数模板描述，为空时，保持原有描述。 示例值：用于压测
Params.N	否	Array of ParamType	待修改参数名及参数值，为空时，各参数保持原有值，支持单条或批量修改。

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 修改参数

输入示例

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

```
X-TC-Action: ModifyDBInstanceParamTpl
```

<公共请求参数>

```
{
  "Params": [
    {
      "Key": "operation.profilng.slowOpThresholdMs",
      "Value": "1001"
    }
  ],
  "TplId": "tpl-2284g3nmw"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "55c81522-95ae-4dc1-a8a4-15b3efd06f36"
  }
}
```

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
InvalidParameter	参数错误。

错误码	描述
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。

数据结构

最近更新时间：2025-06-30 02:05:14

AddNodeList

修改实例节点详情

被如下接口引用：ModifyDBInstanceSpec。

名称	类型	必选	描述
Role	String	是	需要新增的节点角色。 - SECONDARY：Mongod 节点。 - READONLY：只读节点。 - MONGOS：Mongos 节点。 示例值：SECONDARY
Zone	String	是	节点所对应的可用区。 - 单可用区，所有节点在同一可用区。 - 多可用区：当前标准规格是三可用区分布，主从节点不在同一可用区，需注意配置新增节点对应的可用区，且新增后必须满足任意2个可用区节点数大于第3个可用区原则。 示例值：ap-guangzhou-2

Auth

用户权限

被如下接口引用：CreateAccountUser, DescribeAccountUsers, SetAccountUserPrivilege。

名称	类型	必选	描述
Mask	Integer	是	当前账号具有的权限信息。 • 0：无权限。 • 1：只读。 • 2：只写。 • 3：读写。 示例值：1
Namespace	String	是	指具有当前账号权限的数据库名。 • *：表示所有数据库。 • db.name：表示特定name的数据库。 示例值："db_test"

BackupDownloadTask

备份下载任务

被如下接口引用：DescribeBackupDownloadTask。

名称	类型	描述
CreateTime	String	任务创建时间 示例值：2021-01-02 15:04:05
BackupName	String	备份文件名 示例值：cmgo-74sj****_2021-03-24 15:40

名称	类型	描述
ReplicaSetId	String	分片名称 示例值: cmgo-74sj****_0
BackupSize	Integer	备份数据大小, 单位为字节 示例值: 102400
Status	Integer	任务状态。0-等待执行, 1-正在下载, 2-下载完成, 3-下载失败, 4-等待重试 示例值: 0
Percent	Integer	任务进度百分比 示例值: 20
TimeSpend	Integer	耗时, 单位为秒 示例值: 60
Url	String	备份数据下载链接 示例值: http://cmgo-****.b.c/d.tar?e
BackupMethod	Integer	备份文件备份类型, 0-逻辑备份, 1-物理备份 示例值: 1
BackupDesc	String	发起备份时指定的备注信息 示例值: 我的备份
Region	String	地区信息。 示例值: ap-guangzhou
Bucket	String	Bucket信息。 示例值: my_bucket

BackupDownloadTaskStatus

创建备份下载任务结果

被如下接口引用: CreateBackupDownloadTask。

名称	类型	描述
ReplicaSetId	String	分片名 示例值: cmgo-74sjf5mt_0
Status	Integer	任务当前状态。0-等待执行, 1-正在下载, 2-下载完成, 3-下载失败, 4-等待重试 示例值: 0

BackupInfo

备份信息

被如下接口引用: DescribeDBBackups。

名称	类型	描述
InstanceId	String	实例ID。 示例值: cmgo-ca2n****
BackupType	Integer	备份方式, 0-自动备份, 1-手动备份 示例值: 0
BackupName	String	备份名称 示例值: cmgo-ca2n****_2021-12-13 02:02

名称	类型	描述
BackupDesc	String	备份备注 示例值：系统后台每日定时自动备份
BackupSize	Integer	备份文件大小，单位KB 示例值：326594030
StartTime	String	备份开始时间 示例值：2021-12-13 02:02:51
EndTime	String	备份结束时间 示例值：2021-12-13 03:02:51
Status	Integer	备份状态，1-备份中，2-备份成功 示例值：2
BackupMethod	Integer	备份方法，0-逻辑备份，1-物理备份 示例值：0
BackId	Integer	备份记录id 示例值：6997433
DeleteTime	String	备份删除时间 示例值：2099-01-01 00:00:00
BackupRegion	String	异地备份地域 示例值：ap-guangzhou

ClientConnection

客户端连接信息，包括客户端IP和连接数

被如下接口引用：DescribeClientConnections。

名称	类型	描述
IP	String	连接的客户端IP 示例值：192.168.1.1
Count	Integer	对应客户端IP的连接数 示例值：9
InternalService	Boolean	是否为内部ip 示例值：true

CurrentOp

云数据库实例当前操作

被如下接口引用：DescribeCurrentOp。

名称	类型	描述
OpId	Integer	操作序号。 示例值：-130987****
Ns	String	操作所在的命名空间，形式如db.collection。 示例值：db.test
Query	String	操作执行语句。 示例值：执行语句

名称	类型	描述
Op	String	操作类型，可能的取值：aggregate、count、delete、distinct、find、findAndModify、getMore、insert、mapReduce、update和command。 示例值：update
ReplicaSetName	String	操作所在的分片名称。 示例值：cmgo-12asa***_0
NodeName	String	操作所在的节点名称。 示例值：9.18.157.*:70:234****
Operation	String	操作详细信息。 示例值：操作详情
State	String	筛选条件，节点状态，可能的取值为：Primary、Secondary。 示例值：Secondary
MicrosecsRunning	Integer	操作已执行时间（ms）。 示例值：MicrosecsRunning
ExecNode	String	当前操作所在节点信息。 示例值：从节点

DBInstanceInfo

实例信息

被如下接口引用：DescribeDBInstances。

名称	类型	描述
InstanceId	String	实例ID 示例值：cmgo-12sad****
Region	String	地域信息 示例值：ap-guangzhou

DBInstancePrice

数据库实例价格

被如下接口引用：InquirePriceCreateDBInstances, InquirePriceModifyDBInstanceSpec, InquirePriceRenewDBInstances。

名称	类型	必选	描述
UnitPrice	Float	是	单价 示例值：0.17
OriginalPrice	Float	是	原价 示例值：210
DiscountPrice	Float	是	折扣价 示例值：198

DbURL

数据库实例 URI 形式的连接串访问地址示例。

被如下接口引用：DescribeDBInstanceURL。

名称	类型	描述
URLType	String	指 URI 类别，包括：， - CLUSTER_ALL：指通过该 URI 连接库实例的主节点，可读写。 - CLUSTER_READ_READONLY：指通过该 URI 连接实例只读节点。 - CLUSTER_READ_SECONDARY：指通过该 URI 连接实例从节点。 - CLUSTER_READ_SECONDARY_AND_READONLY：指通过该 URI 连接实例只读从节点。 - CLUSTER_PRIMARY_AND_SECONDARY：指通过该 URI 连接实例 主节点与从节点。 - MONGOS_ALL：指通过该 URI 连接每个 Mongos 节点，可读写。 - MONGOS_READ_READONLY：指通过该 URI 连接 Mongos 的只读节点。 - MONGOS_READ_SECONDARY：指通过该 URI 连接 Mongos 的从节点。 - MONGOS_READ_PRIMARY_AND_SECONDARY：指通过该URI 连接 Mongos 的主节点与从节点。 - MONGOS_READ_SECONDARY_AND_READONLY：指通过该URI 连接 Mongos 的从节点与只读节点。 示例值：MONGOS_ALL
Address	String	实例 URI 形式的连接串访问地址示例。 示例值：mongodb://mongouser:**@10.0.5.:27017/test?replicaSet=cmgo-ldig**_0&authSource=admin

FBKeyValue

按key回档，用于筛选数据的键值对

被如下接口引用：FlashBackDBInstance。

名称	类型	必选	描述
Key	String	否	用于按key回档过滤的key 示例值：fbKey
Value	String	否	用于按key回档过滤的value 示例值：fbValue

FlashbackCollection

按key回档，源数据所在的表

被如下接口引用：FlashBackDBInstance。

名称	类型	必选	描述
CollectionName	String	是	按key回档指定的集合名 示例值：my_collection
TargetResultCollectionName	String	是	按key回档到的目标集合名 示例值：my_collection_re
FilterKey	String	是	上传到cos的文件的value所对应的key值 示例值：fKey
KeyValues	Array of FBKeyValue	否	用于按key回档过滤的键值对 示例值：fValue

FlashbackDatabase

按key回档，源数据所在的库表

被如下接口引用：FlashBackDBInstance。

名称	类型	必选	描述
DBName	String	是	按key回档源数据所在库

名称	类型	必选	描述
			示例值：my-db
Collections	Array of FlashbackCollection	是	按key回档的集群数组 示例值：my-collection

InstanceChargePrepaid

描述了实例的计费模式

被如下接口引用：InquirePriceRenewDBInstances, RenewDBInstances。

名称	类型	必选	描述
Period	Integer	否	购买实例的时长，单位：月。取值范围：1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36。默认为1。 示例值：1
RenewFlag	String	否	自动续费标识。取值范围： - NOTIFY_AND_AUTO_RENEW：通知过期且自动续费。在账户余额充足的情况下，实例到期后将按月自动续费。 - NOTIFY_AND_MANUAL_RENEW：通知过期不自动续费。默认为NOTIFY_AND_MANUAL_RENEW。 - DISABLE_NOTIFY_AND_MANUAL_RENEW：不通知过期不自动续费。 示例值：NOTIFY_AND_AUTO_RENEW

InstanceDetail

实例详情

被如下接口引用：DescribeDBInstances。

名称	类型	描述
InstanceId	String	实例ID。 示例值：cmgo-7pje****
InstanceName	String	实例名称。 示例值：test-instance
PayMode	Integer	付费类型，可能的返回值：1-包年包月；0-按量计费 示例值：1
ProjectId	Integer	项目ID。 示例值：1
ClusterType	Integer	集群类型，可能的返回值：0-副本集实例，1-分片实例。 示例值：1
Region	String	地域信息。 示例值：ap-guangzhou
Zone	String	可用区信息。 示例值：ap-guangzhou-2
NetType	Integer	网络类型，可能的返回值：0-基础网络，1-私有网络 示例值：1
VpcId	String	私有网络的ID。 示例值：vpc-0vjm****
SubnetId	String	私有网络的子网ID。 示例值：subnet-empt****

名称	类型	描述
Status	Integer	实例状态，可能的返回值：0-创建中，1-流程处理中，2-运行中，-2-实例已过期。 示例值：2
Vip	String	实例IP。 示例值：10.100.0.**
Vport	Integer	端口号。 示例值：27017
CreateTime	Timestamp	实例创建时间。 示例值：2010-07-25 18:00:00
DeadLine	Timestamp	实例到期时间。 示例值：2010-07-25 20:00:00
MongoVersion	String	实例版本信息。 示例值：MONGO_42_WT
Memory	Integer	实例内存规格，单位为MB。 示例值：4
Volume	Integer	实例磁盘规格，单位为MB。 示例值：120
CpuNum	Integer	实例CPU核心数。 示例值：2
MachineType	String	实例机器类型。 示例值：HIO10G
SecondaryNum	Integer	实例从节点数。 示例值：2
ReplicationSetNum	Integer	实例分片数。 示例值：3
AutoRenewFlag	Integer	实例自动续费标志，可能的返回值：0-手动续费，1-自动续费，2-确认不续费。 示例值：0
UsedVolume	Integer	已用容量，单位MB。 示例值：100
MaintenanceStart	String	维护窗口起始时间。 示例值：04:00:00
MaintenanceEnd	String	维护窗口结束时间。 示例值：05:00:00
ReplicaSets	Array of ShardInfo	分片信息。
ReadonlyInstances	Array of DBInstanceInfo	只读实例信息。
StandbyInstances	Array of DBInstanceInfo	灾备实例信息。
CloneInstances	Array of DBInstanceInfo	临时实例信息。
RelatedInstance	DBInstanceInfo	关联实例信息，对于正式实例，该字段表示它的临时实例信息；对于临时实例，则表示它的正式实例信息;如果为只读/灾备实例,则表示他的主实例信息。
Tags	Array of TagInfo	实例标签信息集合。

名称	类型	描述
InstanceVer	Integer	实例版本标记。 示例值：4
ClusterVer	Integer	实例版本标记。 示例值：1
Protocol	Integer	协议信息，可能的返回值：1-mongodb, 2-dynamodb。 示例值：0
InstanceType	Integer	实例类型，可能的返回值，1-正式实例，2-临时实例，3-只读实例，4-灾备实例 示例值：1
InstanceStatusDesc	String	实例状态描述 示例值：运行中
RealInstanceid	String	实例对应的物理实例id，回档并替换过的实例有不同的Instanceid和RealInstanceid，从barad获取监控数据等场景下需要用物理id获取 示例值：cmgo-8go1o9xl
ZoneList	Array of String	实例当前可用区信息。 示例值：["ap-guangzhou-2","ap-guangzhou-3","ap-guangzhou-4"]
MongosNodeNum	Integer	mongos节点个数。 示例值：3
MongosMemory	Integer	mongos节点内存。 示例值：2
MongosCpuNum	Integer	mongos节点CPU核数。 示例值：1
ConfigServerNodeNum	Integer	Config Server节点个数。 示例值：3
ConfigServerMemory	Integer	Config Server节点内存。 示例值：2
ConfigServerVolume	Integer	Config Server节点磁盘大小。 示例值：20
ConfigServerCpuNum	Integer	Config Server节点CPU核数。 示例值：1
ReadonlyNodeNum	Integer	readonly节点个数。 示例值：3

InstanceEnumParam

实例可修改参数枚举类型集合。

被如下接口引用：DescribeDBInstanceParamTplDetail, DescribeInstanceParams。

名称	类型	描述
CurrentValue	String	参数当前值。 示例值："all"
DefaultValue	String	参数默认值。 示例值："slowOp"
EnumValue	Array of String	枚举值，所有支持的值。 示例值：["off","slowOp","all"]

名称	类型	描述
NeedRestart	String	参数修改之后是否需要重启生效。 – 1：需要重启后生效。 – 0：无需重启，设置成功即可生效。 示例值："0"
ParamName	String	参数名称。 示例值："operationProfiling.mode"
Tips	Array of String	参数说明。 示例值：["事务加锁超时时间","max transaction lockrequest timeout millis"]
ValueType	String	参数值类型说明。 示例值："enum"
Status	Integer	是否为运行中参数值。 – 1：运行中参数值。 – 0：非运行中参数值。 示例值：1

InstanceIntegerParam

实例可修改参数 Integer 类型集合。

被如下接口引用：DescribeDBInstanceParamTplDetail, DescribeInstanceParams。

名称	类型	描述
CurrentValue	String	参数当前值。 示例值："100"
DefaultValue	String	参数默认值。 示例值："100"
Max	String	参数最大值。 示例值："65536"
Min	String	最小值。 示例值："0"
NeedRestart	String	参数修改之后是否需要重启生效。 – 1:需要重启后生效。 – 0：无需重启，设置成功即可生效。 示例值："1"
ParamName	String	参数名称。 示例值："operation.profilng.slowOpThresholdMs"
Tips	Array of String	参数说明。 示例值：["数据库profiler确定慢查询的阈值，单位毫秒，默认值100","The Performance Advisor re"]
ValueType	String	参数类型。 示例值："integer"
Status	Integer	是否为运行中参数值。 – 1：运行中参数值。 – 0：非运行中参数值。 示例值：1
Unit	String	冗余字段，可忽略。 示例值：1

InstanceMultiParam

实例可修改参数Multi类型集合。

被如下接口引用：DescribeDBInstanceParamTplDetail, DescribeInstanceParams。

名称	类型	描述
CurrentValue	String	参数当前值。 示例值："01:00"
DefaultValue	String	参数默认值。 示例值：NULL
EnumValue	Array of String	参考值范围。 示例值：["00:00","23:00"]
NeedRestart	String	参数修改后是否需要重启才会生效。 – 1：需要重启后生效。 – 0：无需重启，设置成功即可生效。 示例值："1"
ParamName	String	参数名称。 示例值："openBalance.window"
Status	Integer	是否为运行中参数值。 – 1：运行中参数值。 – 0：非运行中参数值。 示例值："1"
Tips	Array of String	参数说明。 示例值：["负载均衡时间窗","balance.window"]
ValueType	String	当前值的类型描述，默认为multi。 示例值："multi"

InstanceTextParam

实例可修改参数为 Text 类型的参数集合。

被如下接口引用：DescribeDBInstanceParamTplDetail, DescribeInstanceParams。

名称	类型	描述
CurrentValue	String	参数当前值。 示例值：textParam
DefaultValue	String	参数默认值。 示例值：textParam
NeedRestart	String	修改参数值之后是否需要重启。 示例值：1
ParamName	String	参数名称。 示例值：operationProfiling.****
TextValue	String	Text 类型参数对应的值。 示例值：currentTtext
Tips	Array of String	参数说明。 示例值：参数描述
ValueType	String	参数值类型说明。 示例值：txt

名称	类型	描述
Status	String	是否为运行中的参数值。 - 1: 运行中参数值。 - 0: 非运行中参数值。 示例值: 1

KMSInfoDetail

KMS密钥信息

被如下接口引用: DescribeTransparentDataEncryptionStatus。

名称	类型	必选	描述
KeyId	String	否	主密钥 ID。 示例值: e648b6c-584a-11ef-a5d2-5254002****
KeyName	String	否	主密钥名称。 示例值: stg02-ai-bj-kms-**
CreateTime	String	否	实例与密钥绑定时间。 示例值: 2021-01-02 15:04:05
Status	String	否	密钥状态。 示例值: Enabled
KeyUsage	String	否	密钥用途。 示例值: ENCRYPT_DECRYPT
KeyOrigin	String	否	密钥来源。 示例值: TENCENT_KMS
KmsRegion	String	否	kms所在地域。 示例值: ap-guangzhou

ModifyMongoDBParamType

修改mongoDB实例，请求参数

被如下接口引用: ModifyInstanceParams。

名称	类型	必选	描述
Key	String	是	需要修改的参数名称，请严格参考通过 DescribeInstanceParams 获取的当前实例支持的参数名。 示例值: "operationProfiling.****"
Value	String	是	需要修改的参数名称对应的值，请严格参考通过 DescribeInstanceParams 获取的参数对应的值的范围。 示例值: off

ModifyNetworkAddress

修改数据库地址

被如下接口引用: ModifyDBInstanceNetworkAddress。

名称	类型	必选	描述
NewIPAddress	String	是	新IP地址。 示例值: 172.16.0.**

名称	类型	必选	描述
OldIpAddress	String	是	原IP地址。 示例值：172.16.1.**

NodeProperty

节点属性

被如下接口引用：DescribeDBInstanceNodeProperty。

名称	类型	描述
Zone	String	节点所在的可用区。 示例值：ap-guangzhou-1
NodeName	String	节点名称。 示例值：cmgo-ocdv****_0-node-primary
Address	String	节点访问地址。 示例值：10.4.0.7:30000
WanServiceAddress	String	节点公网访问外网地址(IP或域名，示例为IP方式)。 示例值：113.0.:.27017
Role	String	角色。 示例值：PRIMARY
Hidden	Boolean	是否为Hidden节点 示例值：false
Status	String	节点状态，包括： ORMAL/STARTUP/RECOVERING/STARTUP2/UNKNOWN/DOWN/ROLLBACK/REMOVED 等。 示例值：NORMAL
SlaveDelay	Integer	主从延迟，单位秒。 示例值：1
Priority	Integer	节点优先级。 示例值：10
Votes	Integer	节点投票权。 示例值：1
Tags	Array of NodeTag	节点标签。 示例值：[{"TagKey":"role-cmgo","TagValue":"primary-secondary-group"}]
ReplicateSetId	String	副本集Id。 示例值：cmgo-ocdv****_0

NodeTag

节点Tag

被如下接口引用：DescribeDBInstanceNodeProperty。

名称	类型	必选	描述
TagKey	String	否	节点Tag key 示例值：testKey
TagValue	String	否	节点Tag Value

名称	类型	必选	描述
			示例值：testValue

Operation

需要终止的操作

被如下接口引用：KillOps。

名称	类型	必选	描述
ReplicaSetName	String	是	操作所在的分片名 示例值：cmgo-bbzl****_0
NodeName	String	是	操作所在的节点名 示例值：30.99...:7005:244**
OpId	Integer	是	操作序号 示例值：918759117

ParamTpl

数据库参数模板

被如下接口引用：DescribeDBInstanceParamTpl。

名称	类型	描述
TplName	String	参数模板名称 示例值：压测参数模板
TplId	String	参数模板ID 示例值：tpl-adh42hg1y
MongoVersion	String	适用数据库版本 示例值：MONGO_42_WT
ClusterType	String	适用数据库类型 示例值：SHARD
TplDesc	String	参数模板描述 示例值：用于压测
TplType	String	模板类型，包括DEFAULT（默认模板）及CUSTOMIZE（定制模板）两种类型 示例值：DEFAULT

ParamType

数据库参数

被如下接口引用：CreateDBInstanceParamTpl, ModifyDBInstanceParamTpl。

名称	类型	必选	描述
Key	String	是	参数 示例值：balance.window
Value	String	是	参数值 示例值：1 2

RemoveNodeList

修改实例节点详情

被如下接口引用：ModifyDBInstanceSpec。

名称	类型	必选	描述
Role	String	是	需要删除的节点角色。 – SECONDARY：Mongod 从节点。 – READONLY：只读节点。 – MONGOS：Mongos 节点。 示例值：SECONDARY
NodeName	String	是	要删除的节点 ID。分片集群须指定一组分片要删除的节点名称即可，其余分片对改组对齐。 – 获取方式：登录 MongoDB控制台 ，在 节点管理 页签，可获取节点 ID。 – 特别说明：分片集群同一节点上的分片，仅需指定0分片节点 ID 即可。例如：cmgo-6hfk****_0-node-primary。 示例值：cmgo-3iecwbfx_0-node-readonly0
Zone	String	是	节点所对应的可用区。 – 单可用区，所有节点在同一可用区。 – 多可用区：当前标准规格是三可用区分布，主从节点不在同一可用区，需注意配置所删除节点对应的可用区，且删除后必须满足任意2个可用区节点数大于第3个可用区原则。 示例值：ap-guangzhou-2

ReplicaSetInfo

分片信息

被如下接口引用：CreateBackupDownloadTask。

名称	类型	必选	描述
ReplicaSetId	String	是	副本集ID 示例值：cmgo_ih8uido9_0

ReplicateSetInfo

副本集信息

被如下接口引用：DescribeDBInstanceNodeProperty。

名称	类型	描述
Nodes	Array of NodeProperty	节点属性

SecurityGroup

安全组信息

被如下接口引用：DescribeSecurityGroup。

名称	类型	描述
ProjectId	Integer	所属项目id 示例值：0
CreateTime	String	创建时间 示例值：2021-01-02 15:04:05
Inbound	Array of SecurityGroupBound	进站规则

名称	类型	描述
Outbound	Array of SecurityGroupBound	出站规则
SecurityGroupId	String	安全组id 示例值：sg-lxdvbitn
SecurityGroupName	String	安全组名称 示例值：全放通
SecurityGroupRemark	String	安全组备注 示例值：暴露全部端口到公网和内网，有一定安全风险

SecurityGroupBound

安全组规则

被如下接口引用：DescribeSecurityGroup。

名称	类型	描述
Action	String	执行规则。ACCEPT或DROP 示例值：ACCEPT
CidrIp	String	ip段。 示例值：0.0.0.0/0
PortRange	String	端口范围 示例值：80,443
IpProtocol	String	传输层协议。tcp, udp或ALL 示例值：tcp
Id	String	安全组id代表的地址集合 示例值：sg-
AddressModule	String	地址组id代表的地址集合 示例值：地址集合
ServiceModule	String	服务组id代表的协议和端口集合 示例值：协议和端口集合
Desc	String	描述 示例值：my-SecurityGroupBound

ShardInfo

实例分片详情

被如下接口引用：DescribeDBInstances。

名称	类型	描述
UsedVolume	Float	分片已使用容量 示例值：100
ReplicaSetId	String	分片ID 示例值：cmgo-pzb****_0
ReplicaSetName	String	分片名 示例值：cmgo-pzb****_0
Memory	Integer	分片内存规格，单位为MB

名称	类型	描述
		示例值：8192
Volume	Integer	分片磁盘规格，单位为MB 示例值：51200
OplogSize	Integer	分片Oplog大小，单位为MB 示例值：1024
SecondaryNum	Integer	分片从节点数 示例值：2
RealReplicaSetId	String	分片物理id 示例值：cmgo-pzb****_0

SlowLogItem

慢日志详情

被如下接口引用：DescribeDetailedSlowLogs。

名称	类型	描述
Log	String	慢日志
NodeName	String	节点名称
QueryHash	String	queryHash

SlowLogPattern

用于描述MongoDB数据库慢日志统计信息

被如下接口引用：DescribeSlowLogPatterns。

名称	类型	描述
Pattern	String	慢日志模式 示例值：2024-12-08T18:09:23.364+0800
QueryHash	String	queryHash 示例值：abcdefg
MaxTime	Integer	最大执行时间 示例值：578
AverageTime	Integer	平均执行时间 示例值：546
Total	Integer	该模式慢日志条数 示例值：1

SpecItem

mongodb售卖规格

被如下接口引用：DescribeSpecInfo。

名称	类型	描述
SpecCode	String	规格信息标识。格式如：mongo.H1010G.128g。由节点类型、规格类型、内存规格三部分组成。 - 节点类型，如下所示。

名称	类型	描述
		– mongo: Mongod 节点。 – mongos: Mongos 节点。 – cfgstr: Configserver 节点。 – 规格类型，如下所示。 – HIO10G: 通用高HIO万兆型。 – HCD: 云盘版类型。 – 内存规格，如下所示： – 支持4、8、16、32、64、128、240、512。 – 单位g: 表示GB。128g则表示128GB。 示例值：mongo.HIO10G.128g
Status	Integer	售卖规格有效标志，取值范围如下： – 0: 停止售卖， – 1: 开放售卖。 示例值：1
Cpu	Integer	计算资源规格，CPU核数。 示例值：2
Memory	Integer	内存规格，单位为：MB。 示例值：4096
DefaultStorage	Integer	默认磁盘规格，单位为：MB。 示例值：3072000
MaxStorage	Integer	最大磁盘规格，单位为：MB。 示例值：5120000
MinStorage	Integer	最小磁盘规格，单位为：MB。 示例值：1536000
Qps	Integer	指每秒最大请求次数，单位为：次/秒。 示例值：36000
Conns	Integer	规格所支持的最大连接数限制。 示例值：20000
MongoVersionCode	String	实例存储引擎版本信息。 – MONGO_36_WT: MongoDB 3.6 WiredTiger存储引擎版本。 – MONGO_40_WT: MongoDB 4.0 WiredTiger存储引擎版本。 – MONGO_42_WT: MongoDB 4.2 WiredTiger存储引擎版本。 – MONGO_44_WT: MongoDB 4.4 WiredTiger存储引擎版本。 – MONGO_50_WT: MongoDB 5.0 WiredTiger存储引擎版本。 – MONGO_60_WT: MongoDB 6.0 WiredTiger存储引擎版本。 示例值：MONGO_36_WT
MongoVersionValue	Integer	实例版本对应的数字版本。 示例值：4
Version	String	实例版本信息。支持：3.6、4.2、4.4、5.0、6.0。 示例值：3.6
EngineName	String	存储引擎。 示例值：WiredTiger
ClusterType	Integer	集群类型，取值如下： – 1: 分片集群。 – 0: 副本集集群。 示例值：1
MinNodeNum	Integer	每个副本集最小节点数。 示例值：3

名称	类型	描述
MaxNodeNum	Integer	每个副本集最大节点数。 示例值：7
MinReplicateSetNum	Integer	最小分片数。 示例值：2
MaxReplicateSetNum	Integer	最大分片数。 示例值：20
MinReplicateSetNodeNum	Integer	每个分片最小节点数。 示例值：3
MaxReplicateSetNodeNum	Integer	每个分片最大节点数。 示例值：7
MachineType	String	集群的规格类型，取值范围如下： – HIO10G：通用高HIO万兆型。 – HCD：云盘版类型。 示例值：HIO10G

SpecificationInfo

实例规格信息

被如下接口引用：DescribeSpecInfo。

名称	类型	描述
Region	String	地域信息 示例值：ap-guangzhou
Zone	String	可用区信息 示例值：ap-guangzhou-1
SpecItems	Array of SpecItem	售卖规格信息
SupportMultiAZ	Integer	是否支持跨可用区部署 1-支持，0-不支持 示例值：1

TagInfo

实例标签信息

被如下接口引用：CreateDBInstance, CreateDBInstanceHour, DescribeDBInstances。

名称	类型	必选	描述
TagKey	String	是	标签键 示例值：myTag
TagValue	String	是	标签值 示例值：myTagValue

UserInfo

账户基本信息

被如下接口引用：DescribeAccountUsers。

名称	类型	描述
UserName	String	账号名。 示例值：mongo_test
AuthRole	Array of Auth	账号权限详情。
CreateTime	String	账号创建时间。 示例值：2021-09-28 14:25:38
UpdateTime	String	账号更新时间。 示例值：2021-09-28 14:25:38
UserDesc	String	备注信息。 示例值：mongodb测试账号

错误码

最近更新时间：2025-06-30 02:05:15

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 <code>Authorization</code> 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。

错误码	说明
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。
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.FlashbackByKeyNotOpen	按key回档未开启
FailedOperation.KernelResponseTimeout	内核响应超时。
FailedOperation.NotAllowModifyAddrAfterOpenWanService	当前实例已开启外网访问。
FailedOperation.OperationNotAllowedInInstanceLocking	实例锁定中不允许操作。
InternalError.CheckAppIdFailed	appId校验失败。
InternalError.FindInstanceFailed	实例查询失败。
InternalError.PasswordError	password与原先记录的password不同。
InvalidParameter.CurrentInstanceNotSupportModifyParams	当前实例不支持设置参数。

错误码	说明
InvalidParameter.InvalidVip	无效Vip信息。
InvalidParameter.ModifyMongodbParams	参数名有误
InvalidParameter.ParametersNil	接口参数为空。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameter.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.BackupFileNotFound	备份文件未找到。
InvalidParameterValue.CheckAppIdFailed	appId校验失败。
InvalidParameterValue.ClusterTypeError	无效的实例类型。
InvalidParameterValue.IllegalInstanceName	非法的实例名。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。
InvalidParameterValue.IllegalStatusToOffline	实例状态不允许进行下线，进行下线操作的实例状态必须为隔离状态。
InvalidParameterValue.InstanceHasBeenDeleted	实例已删除。
InvalidParameterValue.InstanceHasBeenIsolated	实例已隔离。
InvalidParameterValue.InvalidTradeOperation	计费相关错误，不允许对当前实例进行对应的新购/续费/配置变更操作。
InvalidParameterValue.LimitParaOutOfRange	limit取值范围[1,100]。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.ModifyModeError	内存和磁盘必须同时升配或降配。
InvalidParameterValue.ModifyMongodbParams	参数值有误。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.MongoVersionNotSupportQueryClient	实例版本不支持查询客户端信息。
InvalidParameterValue.NodeNotFoundInReplica	当前副本集/分片未找到该节点。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.OffsetParaOutOfRange	offset取值范围[0, 10000]。
InvalidParameterValue.OplogSizeOutOfRange	OplogSize参数设置错误，应介于磁盘容量的10%和90%之间。
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。
InvalidParameterValue.PostPaidInstanceBeyondLimit	单个地域后付费实例数量超过限制。
InvalidParameterValue.PrePaidInstanceUnableToIsolate	预付费实例不支持销毁。
InvalidParameterValue.ProjectNotFound	项目不存在。
InvalidParameterValue.ProxyNotSupportQueryClient	proxy版本不支持查询客户端信息，请联系工作人员进行升级。
InvalidParameterValue.QueryOutOfRange	查询offset超出范围。
InvalidParameterValue.QueryTimeOutOfRange	只能查询7天内的慢日志。
InvalidParameterValue.QueryTimeRangeBeyondLimit	只能查询7天内的慢日志。

错误码	说明
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.RegionNotSupportOperation	当前地域尚不支持该操作。
InvalidParameterValue.RegionNotSupportQueryClient	地域尚不支持查询客户端信息。
InvalidParameterValue.ReplicaNotFound	未找到副本集/分片。
InvalidParameterValue.ReplicaSetNumError	副本集（分片）数量错误。
InvalidParameterValue.SecondaryNumError	从节点数错误。
InvalidParameterValue.SecurityGroupId	安全组ID无效。
InvalidParameterValue.SetDiskLessThanUsed	设置的磁盘大小不得低于已用磁盘的1.2倍。
InvalidParameterValue.SlowMSBelowLimit	slowMS参数取值范围[100,65536]。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.StartTimeNotBeforeThanEndTime	起始时间晚于结束时间。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。
InvalidParameterValue.TagNotFound	未找到指定的标签。
InvalidParameterValue.TimeFormatErr	非法的时间格式。
InvalidParameterValue.UserNotFound	用户账户不存在。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.ZoneError	无效的可用区。
LimitExceeded.TooManyRequests	请求太过频繁，触发接口频限。
UnauthorizedOperation.NoAccess	没有访问权限
UnsupportedOperation.KernelVersionNotSupport	内核版本不支持。
UnsupportedOperation.VersionNotSupport	当前版本不支持该操作。

云数据库 MongoDB API 2018-04-08 更新历史

最近更新时间：2025-01-08 01:23:09

第 8 次发布

发布时间：2025-01-08 01:23:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstanceHour](#)
 - 新增入参：UniqVpcId, UniqSubnetId

第 7 次发布

发布时间：2024-12-19 01:47:07

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstance](#)
 - 新增入参：InstanceType

第 6 次发布

发布时间：2024-03-11 15:09:36

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeClientConnections](#)
 - 新增出参：TotalCount

第 5 次发布

发布时间：2020-01-16 18:25:41

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SpecificationInfo](#)
 - 新增成员：SpecItems
 - 删除成员：SpecItem

第 4 次发布

发布时间：2019-08-16 23:30:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeClientConnections](#)

新增数据结构：

- [ClientConnection](#)

第 3 次发布

发布时间：2019-07-16 19:28:09

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateDBInstance](#)
 - 新增出参：InstanceIds
- [CreateDBInstanceHour](#)
 - 新增出参：InstanceIds

第 2 次发布

发布时间：2019-06-27 15:13:10

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AssignProject](#)
- [DescribeDBInstances](#)
- [DescribeSlowLog](#)
- [DescribeSpecInfo](#)
- [RenameInstance](#)
- [SetAutoRenew](#)
- [SetPassword](#)

新增数据结构：

- [MongoDBInstance](#)
- [MongoDBInstanceDetail](#)
- [MongoddbShardInfo](#)
- [SpecItem](#)
- [SpecificationInfo](#)
- [TagInfo](#)

第 1 次发布

发布时间：2018-12-20 20:21:26

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateDBInstance](#)
- [CreateDBInstanceHour](#)
- [TerminateDBInstance](#)
- [UpgradeDBInstance](#)
- [UpgradeDBInstanceHour](#)

简介

最近更新时间：2025-04-25 01:45:15

云数据库 MongoDB（TencentDB for MongoDB），是腾讯云基于全球广受欢迎的 MongoDB 打造的高性能 NoSQL 数据库，100% 完全兼容 MongoDB 协议，提供稳定丰富的监控管理，弹性可扩展、自动容灾，适用于文档型数据库场景。

注意：

- 本章节云数据库 MongoDB API 接口均为最新 API 3.0 接口，后续云数据库 MongoDB 相关新增功能都会在此章节更新。我们强烈推荐您使用最新 API3.0 接口。
- 现有旧版 API 接口功能依然保持，未来可能停止维护，如您仍需使用旧版接口可参考：[云数据库 MongoDB API（旧版）简介](#)。

API 概览

最近更新时间：2024-08-02 01:52:23

账号相关接口

接口名称	接口功能	频率限制（次/秒）
SetPassword	修改云数据库实例的账户密码	20

实例相关接口

接口名称	接口功能	频率限制（次/秒）
AssignProject	指定云数据库实例的所属项目	20
CreateDBInstance	创建云数据库实例（包年包月）	20
CreateDBInstanceHour	创建云数据库实例（按量计费）	20
DescribeClientConnections	查询实例客户端连接信息	20
DescribeDBInstances	查询实例列表	20
DescribeSpecInfo	查询云数据库的售卖规格	20
RenameInstance	修改实例名称	20
SetAutoRenew	设置云数据库实例的续费选项	20
TerminateDBInstance	销毁云数据库实例（按量计费--废弃）	20
UpgradeDBInstance	升级云数据库实例（包年包月）	20
UpgradeDBInstanceHour	升级云数据库实例（按量计费）	20
DescribeSlowLog	查询实例慢日志	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

最近更新时间：2025-05-21 01:24:59

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 `mongodb.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `mongodb.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 `mongodb.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）	mongodb.tencentcloudapi.com
华南地区（广州）	mongodb.ap-guangzhou.tencentcloudapi.com
华东地区（上海）	mongodb.ap-shanghai.tencentcloudapi.com
华东地区（南京）	mongodb.ap-nanjing.tencentcloudapi.com
华北地区（北京）	mongodb.ap-beijing.tencentcloudapi.com
西南地区（成都）	mongodb.ap-chengdu.tencentcloudapi.com
西南地区（重庆）	mongodb.ap-chongqing.tencentcloudapi.com
港澳台地区（中国香港）	mongodb.ap-hongkong.tencentcloudapi.com
亚太东南（新加坡）	mongodb.ap-singapore.tencentcloudapi.com
亚太东南（雅加达）	mongodb.ap-jakarta.tencentcloudapi.com
亚太东南（曼谷）	mongodb.ap-bangkok.tencentcloudapi.com
亚太东北（首尔）	mongodb.ap-seoul.tencentcloudapi.com
亚太东北（东京）	mongodb.ap-tokyo.tencentcloudapi.com
美国东部（弗吉尼亚）	mongodb.na-ashburn.tencentcloudapi.com
美国西部（硅谷）	mongodb.na-siliconvalley.tencentcloudapi.com
南美地区（圣保罗）	mongodb.sa-saopaulo.tencentcloudapi.com
欧洲地区（法兰克福）	mongodb.eu-frankfurt.tencentcloudapi.com

注意：由于金融区和非金融区是隔离不互通的，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）	mongodb.ap-shanghai-fsi.tencentcloudapi.com

金融区接入地域	金融区域名
华南地区（深圳金融）	mongodb.ap-shenzhen-fsi.tencentcloudapi.com

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：2025-03-28 01:47:57

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入参公共参数 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 mongoddb；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, Sign
```

```
edHeaders=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 。
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
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

接口鉴权 v3

最近更新时间：2024-12-25 01:47:57

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区实例列表作为例子，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；

2. 该接口是只读的，不会改变现有资源的状态；
3. 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。

字段名称	解释
	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 是二进制的的数据，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, 否则容易出错
    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++)  
    {  
        sprintf(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;
}
```



```
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 OPENSSL_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 OPENSSL_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, "\nxc-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%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）。

接口鉴权

最近更新时间：2024-12-25 01:47:57

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：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云 API 密钥](#) 的控制台页面
3. 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

参数名称	中文	参数值
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。

签名原文串的拼接规则为：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-guangzhou&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&Region=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都需要做 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");
$param["Nonce"] = 11886;//rand();
```

```
$param["Timestamp"] = 1465185768;//time();
$param["Region"] = "ap-guangzhou";
$param["SecretId"] = $secretId;
$param["Version"] = "2017-03-12";
$param["Action"] = "DescribeInstances";
$param["InstanceIds.0"] = "ins-09dx96dg";
$param["Limit"] = 20;
$param["Offset"] = 0;

ksort($param);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $param 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
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// 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. 拼接签名原文字符串
```

```
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 01:40:10

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：2024-03-11 15:09:46

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。

实例相关接口

升级云数据库实例（按量计费）

最近更新时间：2025-04-25 01:45:14

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(UpgradeDBInstanceHour)用于升级按量计费的MongoDB云数据库实例，可以扩容内存、存储以及oplog

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：UpgradeDBInstanceHour。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例ID，格式如：cmgo-iga0 示例值：cmgo-iga0
Memory	是	Integer	升级后的内存大小，单位：GB 示例值：4
Volume	是	Integer	升级后的硬盘大小，单位：GB 示例值：250
OplogSize	否	Integer	升级后oplog的大小，单位：GB，默认为磁盘空间的10%，允许设置的最小值为磁盘的10%，最大值为磁盘的90% 示例值：25

3. 输出参数

参数名称	类型	描述
DealId	String	订单ID 示例值：2017120116000000267022659982****
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 升级包年包月的云数据库实例

用户需要通过API升级包年包月的云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=UpgradeDBInstanceHour
&Memory=4
&Volume=250
&InstanceId=cmgo-f555zzz
&OplogSize=25
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "6EF60BEC-0242-43AF-BB20-270359FB54A7",
    "DealId": "20171201160000002670226599824833"
  }
}
```

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
InvalidParameter	参数错误。

升级云数据库实例（包年包月）

最近更新时间：2025-04-25 01:45:14

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(UpgradeDBInstance)用于升级包年包月的MongoDB云数据库实例，可以扩容内存、存储以及Oplog

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：UpgradeDBInstance。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例ID，格式如：cmgo-iga0。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-iga0
Memory	是	Integer	升级后的内存大小，单位：GB 示例值：4
Volume	是	Integer	升级后的硬盘大小，单位：GB 示例值：250
OplogSize	否	Integer	升级后oplog的大小，单位：GB，默认为磁盘空间的10%，允许设置的最小值为磁盘的10%，最大值为磁盘的90% 示例值：26

3. 输出参数

参数名称	类型	描述
DealId	String	订单ID 示例值：19297475
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 升级包年包月的云数据库实例

客户需要通过API升级包年包月的云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=UpgradeDBInstance
&Memory=4
&Volume=250
&InstanceId=cmgo-f555zzzz
&OplogSize=26
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "be8f4a30-ea2e-4623-8b6b-0ccce04cd9f7",
    "DealId": "19297475"
  }
}
```

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
InvalidParameter	参数错误。

销毁云数据库实例（按量计费--废弃）

最近更新时间：2025-04-25 01:45:14

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(TerminateDBInstance)用于销毁按量计费的MongoDB云数据库实例。

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：TerminateDBInstance。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。 示例值：cmgo-f555zzzz

3. 输出参数

参数名称	类型	描述
AsyncRequestId	String	订单ID，表示注销实例成功。 示例值：28920
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 销毁按量计费的云数据库实例

输入示例

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

{
```

```
"InstanceId": "cmgo-f555zzzz"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6EF60BEC-0242-43AF-BB20-270359FB54A7",
    "AsyncRequestId": "28920"
  }
}
```

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
InternalServerError.AsyncRequestError	查询异步任务错误。
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.InstanceHasBeenIsolated	实例已隔离。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。

创建云数据库实例（ 按量计费 ）

最近更新时间：2025-04-25 01:45:15

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com 。

本接口(CreateDBInstanceHour)用于创建按量计费的MongoDB云数据库实例，可通过传入实例规格、实例类型、MongoDB版本、购买时长和数量等信息创建云数据库实例。

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：CreateDBInstanceHour。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Memory	是	Integer	实例内存大小，单位：GB 示例值：4
Volume	是	Integer	实例硬盘大小，单位：GB 示例值：250
ReplicateSetNum	是	Integer	副本集个数，1为单副本集实例，大于1为分片集群实例，最大不超过10 示例值：1
SecondaryNum	是	Integer	每个副本集内从节点个数，目前只支持从节点数为2 示例值：2
EngineVersion	是	String	MongoDB引擎版本，值包括MONGO_3_WT 、MONGO_3_ROCKS和MONGO_36_WT 示例值：MONGO_3_WT
Machine	是	String	实例类型，HIO10G：高IO万兆。 示例值：HIO10G
GoodsNum	是	Integer	实例数量，默认值为1, 最小值1，最大值为10 示例值：1
Zone	是	String	可用区信息，格式如：ap-guangzhou-2 示例值：ap-guangzhou-3
InstanceRole	是	String	实例角色，默认传MASTER即可 示例值：MASTER
InstanceType	是	String	实例类型，REPLSET-副本集，SHARD-分片集群 示例值：REPLSET

参数名称	必选	类型	描述
Encrypt	否	Integer	数据是否加密，当且仅当引擎版本为MONGO_3_ROCKS，可以选择加密 示例值：1
VpcId	否	String	私有网络ID，如果不传则默认选择基础网络 示例值：vpc-112****
SubnetId	否	String	私有网络下的子网ID，如果设置了 VpcId，则 SubnetId必填 示例值：subnet-12asd****
ProjectId	否	Integer	项目ID，不填为默认项目 示例值：1118179
SecurityGroup.N	否	Array of String	安全组参数 示例值：sg-123a****
UniqVpcId	否	String	私有网络ID，如果不传则默认选择基础网络 示例值：vpc-112****
UniqSubnetId	否	String	私有网络下的子网ID，如果设置了 VpcId，则 SubnetId必填 示例值：subnet-12asd****

3. 输出参数

参数名称	类型	描述
DealId	String	订单ID 示例值：2019070916000000350241612094****
InstanceIds	Array of String	创建的实例ID列表 示例值：["cmgo-m26****"]
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 创建云数据库实例（按量计费）

客户需要通过API创建按量计费的云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=CreateDBInstanceHour
&Memory=4
&Volume=250
&ReplicateSetNum=1
&SecondaryNum=2
&EngineVersion=MONGO_3_WT
&Machine=TGIO
&GoodsNum=1
&Zone=ap-guangzhou-3
&InstanceRole=MASTER
&InstanceType=REPLSET
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "RequestId": "eaf9b19d-5ad4-4ca2-9fc4-a319aeb9181f",
    "DealId": "20190709160000003502416120946732",
    "InstanceIds": [
      "cmgo-m26c6jsf"
    ]
  }
}
```

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
InvalidParameterValue.InstanceTypeError	实例类型取值错误，需要指定实例类型为副本集或分片。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.ZoneError	无效的可用区。

创建云数据库实例（包年包月）

最近更新时间：2025-04-25 01:45:15

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(CreateDBInstance)用于创建包年包月的MongoDB云数据库实例。

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：CreateDBInstance。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SecondaryNum	是	Integer	每个副本集内从节点个数 示例值：2
Memory	是	Integer	实例内存大小，单位：GB 示例值：4
Volume	是	Integer	实例硬盘大小，单位：GB 示例值：100
MongoVersion	是	String	指版本信息。具体支持的版本信息，请通过接口 DescribeSpecInfo 获取。- MONGO_36_WT：MongoDB 3.6 WiredTiger存储引擎版本。- MONGO_40_WT：MongoDB 4.0 WiredTiger存储引擎版本。- MONGO_42_WT：MongoDB 4.2 WiredTiger存储引擎版本。- MONGO_44_WT：MongoDB 4.4 WiredTiger存储引擎版本。- MONGO_50_WT：MongoDB 5.0 WiredTiger存储引擎版本。- MONGO_60_WT：MongoDB 6.0 WiredTiger存储引擎版本。 示例值：MONGO_44_WT
MachineCode	是	String	机器类型，HIO10G：高IO万兆。 示例值：HIO10G
GoodsNum	是	Integer	实例数量，默认值为1，最小值1，最大值为10 示例值：1
Zone	是	String	实例所属区域名称，格式如：ap-guangzhou-2 示例值：ap-guangzhou-2
TimeSpan	是	Integer	时长，购买月数 示例值：1
Password	是	String	实例密码 示例值：****

参数名称	必选	类型	描述
ProjectId	否	Integer	项目ID，不填为默认项目 示例值：0
SecurityGroup.N	否	Array of String	安全组参数 示例值：sg-1ad2****
UniqVpcId	否	String	私有网络ID，如果不传则默认选择基础网络 示例值：vpc-0akb****
UniqSubnetId	否	String	私有网络下的子网ID，如果设置了 VpcId，则 SubnetId必填 示例值：subnet-fyrt**** (
InstanceType	否	String	实例类型，REPLSET-副本集，SHARD-分片集群，默认为REPLSET 示例值：REPLSET

3. 输出参数

参数名称	类型	描述
DealId	String	订单ID 示例值：7142863
InstanceIds	Array of String	创建的实例ID列表 示例值：["cmgo-po5f****"]
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 创建包年包月的云数据库实例

客户需要通过API创建包年包月云数据库实例

输入示例

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

{
  "Zone": "ap-guangzhou-2",
  "UniqVpcId": "vpc-0akbol5v",
  "TimeSpan": "1",
  "ProjectId": "0",
  "MachineCode": "TGIO",
  "GoodsNum": "1",
  "Volume": "250",
  "SecondaryNum": "2",
  "UniqSubnetId": "subnet-fyrtjbqw",
  "Memory": "4",
  "MongoVersion": "MONGO_42_WT",
```

```
"Password": "pwd123456"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d88095e5-50e8-4245-a0cf-993a536f9b20",
    "DealId": "7142863",
    "InstanceIds": [
      "cmgo-po5f8991"
    ]
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameter.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。

错误码	描述
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.SecondaryNumError	从节点数错误。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneError	无效的可用区。

查询实例客户端连接信息

最近更新时间：2025-04-25 01:45:15

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeClientConnections)用于查询实例客户端连接信息，包括连接IP和连接数量。目前只支持3.2版本的MongoDB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：DescribeClientConnections。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-eqmoaxxx

3. 输出参数

参数名称	类型	描述
Clients	Array of ClientConnection	客户端连接信息，包括客户端IP和对应IP的连接数量 示例值：[{"Count":1,"IP":"xx"}, {"Count":1,"IP":"xx"}]
TotalCount	Integer	连接数总结 示例值：10
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 查询云数据库实例的客户端连接情况

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeClientConnections
&InstanceId=cmgo-eqmoaxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Clients": [
      {
        "IP": "10.0.0.1",
        "Count": 1
      }
    ],
    "TotalCount": 0,
    "RequestId": "59b477da-6473-4ea8-85b3-7f4473744981"
  }
}
```

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
InternalServerError	内部错误。
InvalidParameterValue.MongoVersionNotSupportQueryClient	实例版本不支持查询客户端信息。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.ProxyNotSupportQueryClient	proxy版本不支持查询客户端信息，请联系工作人员进行升级。
InvalidParameterValue.RegionNotSupportQueryClient	地域尚不支持查询客户端信息。

设置云数据库实例的续费选项

最近更新时间：2025-04-25 01:45:14

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(SetAutoRenew)用于设置包年包月云数据库实例的续费选项。

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：SetAutoRenew。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	是	Array of String	实例ID列表，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：["cmgo-8l3r6yyt\r\n"]
AutoRenewFlag	是	Integer	配置自动续费标识。 - 0：手动续费。 - 1：自动续费。 - 2：确认不续费。 示例值：1

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 设置包年包月实例的续费选项

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=SetAutoRenew
&InstanceIds.0=cmgo-8l3r6yyt
&AutoRenewFlag=1
&<公共请求参数>
```


输出示例

```
{
  "Response": {
    "RequestId": "1d0d55f9-15ee-4f00-b460-7b645a62cc3a"
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.PostPayRenewError	按量计费实例不允许续费。

修改实例名称

最近更新时间：2025-04-25 01:45:14

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(RenameInstance)用于修改云数据库实例的名称。

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：RenameInstance。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-eqmoa5sf
NewName	是	String	实例自定义名称 示例值：test_name

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 重命名云数据库实例

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=RenameInstance
&InstanceId=cmgo-eqmoa5sf
&NewName=test_name
&<<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8e2a339e-5ba9-4da3-8dc2-65a0a6314811"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalInstanceName	非法的实例名。
InvalidParameterValue.InstanceHasBeenDeleted	实例已删除。
InvalidParameterValue.InstanceHasBeenIsolated	实例已隔离。
InvalidParameterValue.NotFoundInstance	未找到实例。

查询云数据库的售卖规格

最近更新时间：2025-04-25 01:45:14

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeSpecInfo)用于查询实例的售卖规格。

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：DescribeSpecInfo。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Zone	否	String	可用区 示例值：ap-guangzhou-2

3. 输出参数

参数名称	类型	描述
SpecInfoList	Array of SpecificationInfo	实例售卖规格信息列表 示例值：详见数据结构。
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 查询云数据库售卖

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeSpecInfo&
Region=ap-guangzhou
&<公共请求参数>
```

输出示例

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

```
"RequestId": "7e143607-fc69-4761-b0ba-529cb1786ef3",
"SpecInfoList": [
  {
    "Region": "ap-guangzhou",
    "SpecItems": [
      {
        "ClusterType": 1,
        "Conns": 0,
        "Cpu": 1,
        "DefaultStorage": 20480,
        "EngineName": "WiredTiger",
        "MachineType": "CONFIG_SERVER",
        "MaxNodeNum": 32,
        "MaxReplicateSetNodeNum": 0,
        "MaxReplicateSetNum": 0,
        "MaxStorage": 20480,
        "Memory": 2048,
        "MinNodeNum": 3,
        "MinReplicateSetNodeNum": 0,
        "MinReplicateSetNum": 0,
        "MinStorage": 20480,
        "MongoVersionCode": "MONGO_42_WT",
        "MongoVersionValue": 9,
        "Qps": 0,
        "SpecCode": "cfgsrv.STD.1C2G",
        "Status": 1,
        "Version": "4.2"
      }
    ],
    "Zone": "ap-guangzhou-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
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.ZoneError	无效的可用区。

查询实例列表

最近更新时间：2025-04-25 01:45:15

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeDBInstances)用于查询云数据库实例列表，支持通过项目ID、实例ID、实例状态等过滤条件来筛选实例。支持查询主实例、灾备实例和只读实例信息列表。

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：DescribeDBInstances。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	否	Array of String	实例ID列表，格式如：cmgo-p8vn。与云数据库控制台页面中显示的实例ID相同。 示例值： cmgo-7pje
InstanceType	否	Integer	实例类型，取值范围： 0：所有实例 1：正式实例 2：临时实例 3：只读实例 -1：正式实例+只读+灾备实例 示例值：0
ClusterType	否	Integer	集群类型，取值范围： 0：副本集实例 1：正式实例 -1：所有实例 示例值：0
Status.N	否	Array of Integer	实例状态，取值范围： 0：待初始化 1：流程执行中 2：有效实例 -2：已过期实例 示例值：2

参数名称	必选	类型	描述
VpcId	否	String	私有网络的ID，基础网络则不传该参数。 示例值：1
SubnetId	否	String	私有网络的子网ID，基础网络则不传该参数。入参设置该参数的同时，必须设置相应的VpcId。 示例值：1
PayMode	否	Integer	付费类型，取值范围： 0：按量计费 1：包年包月 -1：按量计费+包年包月 示例值：1
Limit	否	Integer	单次请求返回的数量，最小值为1，最大值为100，默认值为20。 示例值：10
Offset	否	Integer	偏移量，默认值为0。 示例值：1
OrderBy	否	String	返回结果集排序的字段，目前支持： - ProjectId：按照项目ID排序 - InstanceName：按照实例名称排序 - CreateTime：根据创建时间排序 示例值：InstanceName
OrderByType	否	String	返回结果集排序方式，目前支持："ASC"或者"DESC"。 - ASC：顺序取值 - DESC：倒序取值 示例值：ASC

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合查询条件的实例总数 示例值：19
InstanceDetails	Array of MongoDBInstanceDetail	实例详细信息 示例值：{"Response":{"InstanceDetails":[],"RequestId":"88c9e2c6-a116-4d68-b77e-8136c5e0****","TotalCount":0}}
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 查询实例列表

输入示例

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


X-TC-Action: DescribeDBInstances

<公共请求参数>

```
{
  "Limit": "1"
}
```

输出示例

```
{
  "Response": {
    "InstanceDetails": [
      {
        "AutoRenewFlag": 0,
        "CloneInstances": [],
        "ClusterType": 0,
        "ClusterVer": 1,
        "CpuNum": 2,
        "CreateTime": "2019-04-02 19:29:16",
        "DeadLine": "2019-04-04 19:29:16",
        "InstanceId": "cmgo-rw2v134z",
        "InstanceName": "cmgo-rw2v134z",
        "InstanceStatusDesc": "回档中",
        "InstanceType": 2,
        "InstanceVer": 4,
        "MachineType": "HIO10G",
        "MaintenanceEnd": "05:00:00",
        "MaintenanceStart": "04:00:00",
        "Memory": 4096,
        "MongoVersion": "MONGO_3_WT",
        "NetType": 1,
        "PayMode": 0,
        "ProjectId": 0,
        "Protocol": 0,
        "ReadOnlyInstances": [],
        "RealInstanceId": "cmgo-da0pc0dr",
        "Region": "ap-guangzhou",
        "RelatedInstance": {
          "InstanceId": "",
          "Region": ""
        },
        "ReplicaSets": [
          {
            "Memory": 4096,
            "OplogSize": 10240,
            "RealReplicaSetId": "cmgo-rw2v134z_0",
            "ReplicaSetId": "cmgo-rw2v134z_0",
            "ReplicaSetName": "cmgo-rw2v134z_0",
            "SecondaryNum": 2,
            "UsedVolume": 0,
            "Volume": 102400
          }
        ]
      }
    ]
  }
}
```

```

    }
  ],
  "ReplicationSetNum": 1,
  "SecondaryNum": 2,
  "StandbyInstances": [],
  "Status": 1,
  "SubnetId": "subnet-b0lagkxi",
  "Tags": [],
  "UsedVolume": 0,
  "Vip": "",
  "Volume": 102400,
  "VpcId": "vpc-orgohlut",
  "Vport": 27017,
  "Zone": "ap-guangzhou-3"
}
],
"RequestId": "302530d2-ee57-495e-89d0-51e03b11815e",
"TotalCount": 19
}
}

```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。

错误码	描述
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneError	无效的可用区。

指定云数据库实例的所属项目

最近更新时间：2025-04-25 01:45:15

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口（AssignProject）用于指定云数据库实例的所属项目。

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：AssignProject。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceIds.N	是	Array of String	实例 ID 列表。格式如：cmgo-p8vn****，与云数据库控制台页面中显示的实例 ID 相同。 示例值：["cmgo-eqmoa5sflr\n"]
ProjectId	是	Integer	项目ID。项目 ID 具有唯一性，请 登录 MongoDB 控制台 ，在右上角的账户信息的下拉菜单中，选择 项目管理 ，即可获取项目ID。 示例值：12

3. 输出参数

参数名称	类型	描述
FlowIds	Array of Integer	返回的异步任务ID列表。 示例值：[1081]
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 指定云数据库实例的所属项目

调整实例的所属项目

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=AssignProject
&InstanceIds.0=cmgo-eqmo****
```

```
&ProjectId=12
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "FlowIds": [
      1081
    ],
    "RequestId": "8e2a339e-5ba9-4da3-8dc2-65a0a6314811"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.ProjectNotFound	项目不存在。

查询实例慢日志

最近更新时间：2025-04-25 01:45:15

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(DescribeSlowLogs)用于获取云数据库实例的慢查询日志。

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：DescribeSlowLog。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vnipr5。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-eqmoa5sf
StartTime	是	Timestamp	慢日志起始时间，格式：yyyy-mm-dd hh:mm:ss，如：2019-06-01 10:00:00。查询起止时间间隔不能超过24小时，只允许查询最近7天内慢日志。 示例值：2019-06-04 20:21:00
EndTime	是	Timestamp	慢日志终止时间，格式：yyyy-mm-dd hh:mm:ss，如：2019-06-02 12:00:00。查询起止时间间隔不能超过24小时，只允许查询最近7天内慢日志。 示例值：2019-06-04 20:22:00
SlowMS	是	Integer	慢日志执行时间阈值，返回执行时间超过该阈值的慢日志，单位为毫秒(ms)，最小为100毫秒。 示例值：200
Offset	否	Integer	偏移量，最小值为0，最大值为10000，默认值为0。 示例值：20
Limit	否	Integer	分页大小，最小值为1，最大值为100，默认值为20。 示例值：20

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合查询条件的慢查询日志总数。 示例值：1
SlowLogList	Array of String	符合查询条件的慢查询日志详情。 示例值：["Tue Jun 4 17:52:39.247 I COMMAND [conn2898] xxxx"]
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

参数名称	类型	描述
		RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询云数据库MongoDB实例慢日志

客户需要通过API查询云数据库MongoDB实例慢日志

输入示例

```
https://mongodb.tencentcloudapi.com/?Action=DescribeSlowLog
&InstanceId=cmgo-eqmoa5sf
&StartTime=2019-06-04 20:21:00
&EndTime=2019-06-04 20:22:00
&SlowMS=200
&Offset=20
&Limit=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "RequestId": "302530d2-ee57-495e-89d0-51e03b11815e",
    "SlowLogList": [
      "Tue Jun 4 17:52:39.247 I COMMAND [conn2898] xxxx"
    ]
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.LimitParaOutOfRange	limit取值范围[1,100]。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.OffsetParaOutOfRange	offset取值范围[0, 10000]。
InvalidParameterValue.QueryTimeOutOfRange	只能查询7天内的慢日志。
InvalidParameterValue.QueryTimeRangeBeyondLimit	只能查询7天内的慢日志。
InvalidParameterValue.SlowMSBelowLimit	slowMS参数取值范围[100,65536]。
InvalidParameterValue.StartTimeNotBeforeThanEndTime	起始时间晚于结束时间。
InvalidParameterValue.TimeFormatErr	非法的时间格式。

账号相关接口

修改云数据库实例的账户密码

最近更新时间：2025-04-25 01:45:13

1. 接口描述

接口请求域名：mongodb.tencentcloudapi.com。

本接口(SetPassword)用于设置（初始化）MongoDB云数据库实例账户密码。

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：SetPassword。
Version	是	String	公共参数 ，本接口取值：2018-04-08。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID，格式如：cmgo-p8vn。与云数据库控制台页面中显示的实例ID相同 示例值：cmgo-eqmo
UserName	是	String	实例账户名。初始化实例密码，本参数传mongouser。 示例值：test_user
Password	是	String	指定账户的新密码，密码格式为8-32个字符长度，至少包含字母、数字和字符（!@#%^()_）中的两种 示例值：***

3. 输出参数

参数名称	类型	描述
FlowId	Integer	返回的异步任务ID 示例值：1082****
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 设置云数据库实例账户密码

输入示例

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

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

{
  "InstanceId": "cmgo-egmoa5sf",
  "UserName": "test_user",
  "Password": "test_password123"
}
```

输出示例

```
{
  "Response": {
    "FlowId": 1082,
    "RequestId": "a02372cf-5e09-4d17-ab79-2ee3f60e86ff"
  }
}
```

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
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.NotFoundInstance	未找到实例。

错误码	描述
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。
InvalidParameterValue.UserNotFound	用户账户不存在。

数据结构

最近更新时间：2025-01-09 01:22:20

ClientConnection

客户端连接信息，包括客户端IP和连接数

被如下接口引用：DescribeClientConnections。

名称	类型	描述
IP	String	连接的客户端IP 示例值：198.0.0.1
Count	Integer	对应客户端IP的连接数 示例值：10

MongoDBInstance

实例信息

被如下接口引用：DescribeDBInstances。

名称	类型	描述
InstanceId	String	实例ID 示例值：cmgo-12as3****
Region	String	地域信息 示例值：ap-guangzhou

MongoDBInstanceDetail

实例详情

被如下接口引用：DescribeDBInstances。

名称	类型	描述
InstanceId	String	实例ID 示例值：cmgo-12adfw****
InstanceName	String	实例名称 示例值：myInstance
PayMode	Integer	付费类型，可能的返回值：1-包年包月；0-按量计费 示例值：1
ProjectId	Integer	项目ID 示例值：1
ClusterType	Integer	集群类型，可能的返回值：0-副本集实例，1-分片实例， 示例值：0
Region	String	地域信息 示例值：ap-guangzhou
Zone	String	可用区信息 示例值：ap-guangzhou-2
NetType	Integer	网络类型，可能的返回值：0-基础网络，1-私有网络

名称	类型	描述
		示例值：1
VpcId	String	私有网络的ID 示例值：vpc-12w1f***
SubnetId	String	私有网络的子网ID 示例值：subnet-12sdsa****
Status	Integer	实例状态，可能的返回值：0-待初始化，1-流程处理中，2-运行中，-2-实例已过期 示例值：1
Vip	String	实例IP 示例值：vpc-12w1f***
Vport	Integer	端口号 示例值：27017
CreateTime	Timestamp	实例创建时间 示例值：2021-01-02 15:04:05
DeadLine	Timestamp	实例到期时间 示例值：2021-05-02 15:04:05
MongoVersion	String	实例版本信息 示例值：MONGO_40_WT
Memory	Integer	实例内存规格，单位为MB 示例值：4096
Volume	Integer	实例磁盘规格，单位为MB 示例值：512000
CpuNum	Integer	实例CPU核心数 示例值：2
MachineType	String	实例机器类型 示例值：HIO10G
SecondaryNum	Integer	实例从节点数 示例值：2
ReplicationSetNum	Integer	实例分片数 示例值：0
AutoRenewFlag	Integer	实例自动续费标志，可能的返回值：0-手动续费，1-自动续费，2-确认不续费 示例值：1
UsedVolume	Integer	已用容量，单位MB 示例值：10240
MaintenanceStart	String	维护窗口起始时间 示例值：00:00
MaintenanceEnd	String	维护窗口结束时间 示例值：03:00
ReplicaSets	Array of MongoDbShardInfo	分片信息
ReadonlyInstances	Array of MongoDBInstance	只读实例信息
StandbyInstances	Array of MongoDBInstance	灾备实例信息

名称	类型	描述
CloneInstances	Array of MongoDBInstance	临时实例信息
RelatedInstance	MongoDBInstance	关联实例信息，对于正式实例，该字段表示它的临时实例信息；对于临时实例，则表示它的正式实例信息;如果为只读/灾备实例,则表示他的主实例信息 示例值：1
Tags	Array of TagInfo	实例标签信息集合
InstanceVer	Integer	实例标记 示例值：4
ClusterVer	Integer	实例标记 示例值：1
Protocol	Integer	协议信息，可能的返回值：1-mongodb, 2-dynamodb 示例值：2
InstanceType	Integer	实例类型，可能的返回值，1-正式实例，2-临时实例，3-只读实例，4-灾备实例 示例值：1
InstanceStatusDesc	String	实例状态描述 示例值：运行中
RealInstanceid	String	实例对应的物理实例ID，回档并替换过的实例有不同的Instanceid和RealInstanceid，从barad获取监控数据等场景下需要用物理id获取 示例值：cmgo-12adfw****

MongodbShardInfo

实例分片详情

被如下接口引用：DescribeDBInstances。

名称	类型	描述
UsedVolume	Float	分片已使用容量 示例值：0
ReplicaSetId	String	分片ID 示例值：cmgo-ghr****_0
ReplicaSetName	String	分片名 示例值：cmgo-ghr****_0
Memory	Integer	分片内存规格，单位为MB 示例值：4096
Volume	Integer	分片磁盘规格，单位为MB 示例值：512000
OplogSize	Integer	分片Oplog大小，单位为MB 示例值：10240
SecondaryNum	Integer	分片从节点数 示例值：3
RealReplicaSetId	String	分片物理ID 示例值：分片ID详情列表

SpecItem

mongodb售卖规格

被如下接口引用：DescribeSpecInfo。

名称	类型	必选	描述
SpecCode	String	是	规格信息标识 示例值：mongo.HIO10G.128g
Status	Integer	是	规格有效标志，取值：0-停止售卖，1-开放售卖 示例值：1
MachineType	String	是	机器类型，取值：0-HIO，4-HIO10G 示例值：4
Cpu	Integer	是	cpu核心数 示例值：2
Memory	Integer	是	内存规格，单位为MB 示例值：4096
DefaultStorage	Integer	是	默认磁盘规格，单位MB 示例值：3072000
MaxStorage	Integer	是	最大磁盘规格，单位MB 示例值：5120000
MinStorage	Integer	是	最小磁盘规格，单位MB 示例值：1536000
Qps	Integer	是	可承载qps信息 示例值：36000
Conns	Integer	是	连接数限制 示例值：20000
MongoVersionCode	String	是	实例mongodb版本信息 示例值：MONGO_40_WT
MongoVersionValue	Integer	是	实例mongodb版本号 示例值：9
Version	String	是	实例mongodb版本号（短） 示例值：4.0
EngineName	String	是	存储引擎 示例值：WiredTiger
ClusterType	Integer	是	集群类型，取值：1-分片集群，0-副本集集群 示例值：1
MinNodeNum	Integer	是	最小副本集从节点数 示例值：3
MaxNodeNum	Integer	是	最大副本集从节点数 示例值：7
MinReplicateSetNum	Integer	是	最小分片数 示例值：2
MaxReplicateSetNum	Integer	是	最大分片数 示例值：20
MinReplicateSetNodeNum	Integer	是	最小分片从节点数 示例值：3

名称	类型	必选	描述
MaxReplicateSetNodeNum	Integer	是	最大分片从节点数 示例值：7

SpecificationInfo

实例规格信息

被如下接口引用：DescribeSpecInfo。

名称	类型	描述
Region	String	地域信息 示例值：ap-guangzhou
Zone	String	可用区信息 示例值：ap-guangzhou-2
SpecItems	Array of SpecItem	售卖规格信息 示例值：详见结构

TagInfo

实例标签信息

被如下接口引用：DescribeDBInstances。

名称	类型	描述
TagKey	String	标签Key值 示例值：myTag
TagValue	String	标签值 示例值：myTagValue

错误码

最近更新时间：2025-01-08 01:23:08

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 <code>Authorization</code> 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。

错误码	说明
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。
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
InternalError.AsyncRequestError	查询异步任务错误。
InvalidParameter.PermissionDenied	当前子账号无权执行该操作。
InvalidParameter.ZoneClosed	可用区已关闭售卖。
InvalidParameterValue.IllegalInstanceName	非法的实例名。
InvalidParameterValue.IllegalInstanceStatus	非法的实例状态。
InvalidParameterValue.InstanceHasBeenDeleted	实例已删除。
InvalidParameterValue.InstanceHasBeenIsolated	实例已隔离。
InvalidParameterValue.InstanceTypeError	实例类型取值错误，需要指定实例类型为副本集或分片。
InvalidParameterValue.LimitParaOutOfRange	limit取值范围[1,100]。

错误码	说明
InvalidParameterValue.LockFailed	实例锁定失败。
InvalidParameterValue.MachineTypeError	错误的机型。
InvalidParameterValue.MongoVersionError	实例版本错误。
InvalidParameterValue.MongoVersionNotSupportQueryClient	实例版本不支持查询客户端信息。
InvalidParameterValue.NotFoundInstance	未找到实例。
InvalidParameterValue.OffsetParaOutOfRange	offset取值范围[0, 10000]。
InvalidParameterValue.PasswordRuleFailed	密码不符合规范。
InvalidParameterValue.PostPayRenewError	按量计费实例不允许续费。
InvalidParameterValue.ProjectNotFound	项目不存在。
InvalidParameterValue.ProxyNotSupportQueryClient	proxy版本不支持查询客户端信息，请联系工作人员进行升级。
InvalidParameterValue.QueryTimeOutOfRange	只能查询7天内的慢日志。
InvalidParameterValue.QueryTimeRangeBeyondLimit	只能查询7天内的慢日志。
InvalidParameterValue.RegionError	无效的地域。
InvalidParameterValue.RegionNotSupportQueryClient	地域尚不支持查询客户端信息。
InvalidParameterValue.SecondaryNumError	从节点数错误。
InvalidParameterValue.SlowMSBelowLimit	slowMS参数取值范围[100,65536]。
InvalidParameterValue.SpecNotOnSale	购买规格错误。
InvalidParameterValue.StartTimeNotBeforeThanEndTime	起始时间晚于结束时间。
InvalidParameterValue.StatusAbnormal	实例处于不允许操作的状态。
InvalidParameterValue.TimeFormatErr	非法的时间格式。
InvalidParameterValue.UserNotFound	用户账户不存在。
InvalidParameterValue.VpcIdOrSubnetIdNotFound	未找到虚拟网络（子网）。
InvalidParameterValue.ZoneError	无效的可用区。