

云压测 API 文档



腾讯云

【版权声明】

©2013–2025 腾讯云版权所有

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

【商标声明】



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

【服务声明】

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

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

【联系我们】

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

文档目录

API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

参数类型

PTS相关接口

停止定时任务

停止任务

调整任务目标RPS

复制场景

创建告警通知接收组

创建定时任务

创建文件

创建项目

创建场景

删除告警通知接收组

删除定时任务

删除文件

删除任务

删除项目

删除场景

查询告警通知接收组

查询告警历史

查询系统支持的指标

查询检查点汇总信息

列出定时任务

查询文件列表

查询任务列表

查询标签值

查询指标所有的label及values值

查询压测过程日志

查询项目列表

查询地域列表

查询请求汇总信息

批量查询指标

批量查询指标矩阵

查询指标矩阵

查询指标

查询场景及对应的任务

查询场景列表

生成临时COS凭证

重启定时任务

创建并启动任务

更新定时任务

更新关联文件场景

更新任务

更新项目

更新场景

创建环境

- 删除环境
- 查看环境列表
- 查询错误详情汇总信息
- 查询采样日志
- 更新环境
- 数据结构
- 错误码

更新历史

最近更新时间：2025-05-20 01:26:48

第 21 次发布

发布时间：2025-05-20 01:26:43

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [CustomSampleMatrix](#)
 - 新增成员：Step

第 20 次发布

发布时间：2025-03-07 01:23:46

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeSampleLogs](#)
 - 新增入参：Offset
- [DescribeSampleMatrixBatchQuery](#)
 - 新增入参：MaxPoint
- [DescribeSampleMatrixQuery](#)
 - 新增入参：MaxPoint

修改数据结构：

- [ScriptInfo](#)
 - 新增成员：Uploaded

第 19 次发布

发布时间：2025-01-14 01:22:46

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ErrorSummary](#)
 - 修改成员：Status, Result, Count, Rate, Message, Proto
- [Project](#)
 - 修改成员：ProjectId, Name, Description, Tags, Status, CreatedAt, UpdatedAt

第 18 次发布

发布时间：2024-12-06 01:26:37

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RequestSummary](#)
 - 新增成员：RPS

第 17 次发布

发布时间：2024-11-07 01:56:43

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Job](#)
 - 新增成员：Usage

第 16 次发布

发布时间：2024-10-23 01:29:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Job](#)
 - 新增成员：PayMode

第 15 次发布

发布时间：2024-04-19 01:24:07

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeRegions](#)
 - 新增入参：LoadType

第 14 次发布

发布时间：2024-03-27 01:23:10

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateEnvironment](#)
- [DeleteEnvironments](#)
- [DescribeEnvironments](#)
- [UpdateEnvironment](#)

第 13 次发布

发布时间：2023-08-23 01:36:48

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateScenario](#)
 - 新增入参：EnvId

第 12 次发布

发布时间：2023-07-21 01:35:57

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Scenario](#)
 - 修改成员： ScenarioId, Name, Description, Type, Status, Load, EncodedScripts, Configs, Extensions, Datasets, SLAId, CronId, CreatedAt, UpdatedAt, ProjectId, AppId, Uin, SubAccountUin, TestScripts, Protocols, RequestFiles, SLAPolicy, Plugins, DomainNameConfig, NotificationHooks, Owner, ProjectName

第 11 次发布

发布时间：2023-04-05 01:50:08

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeErrorSummary](#)

新增数据结构：

- [ErrorSummary](#)

修改数据结构：

- [Concurrency](#)
 - 新增成员：Resources

第 10 次发布

发布时间：2023-03-02 01:32:50

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Job](#)
 - 新增成员：ProjectName, ScenarioName
- [Scenario](#)
 - 新增成员：ProjectName

第 9 次发布

发布时间：2022-12-28 14:35:16

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateScenario](#)
 - 新增入参：Owner
- [DescribeScenarioWithJobs](#)
 - 新增入参：ScenarioType, Owner
- [UpdateScenario](#)
 - 新增入参：Owner

修改数据结构：

- [Scenario](#)
 - 新增成员：Owner
 - 修改成员： Name, AppId, Uin, SubAccountUin, TestScripts, Protocols, RequestFiles, SLAPolicy, Plugins, DomainNameConfig, NotificationHooks

第 8 次发布

发布时间：2022-12-08 01:40:36

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Job](#)
 - 新增成员：Message

第 7 次发布

发布时间：2022-11-18 06:42:29

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Job](#)
 - 新增成员：NetworkReceiveRate, NetworkSendRate

第 6 次发布

发布时间：2022-11-04 06:33:49

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [File](#)
 - 新增成员：Appld
- [ScriptInfo](#)
 - 新增成员：Fileld

第 5 次发布

发布时间：2022-10-20 06:35:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Concurrency](#)
 - 新增成员：GracefulStopSeconds
- [CustomSample](#)
 - 新增成员：Name
- [File](#)
 - 新增成员：AppID, Uin, SubAccountUin
- [RequestsPerSecond](#)
 - 新增成员：GracefulStopSeconds

第 4 次发布

发布时间：2022-09-19 06:43:24

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [NotificationHook](#)

修改数据结构：

- [AlertChannelRecord](#)
 - 新增成员：AppId, Uin, SubAccountUin
- [AlertRecord](#)
 - 新增成员：AppId, Uin, SubAccountUin, ScenarioName, Target, JobSLAId, JobSLADescription
- [CronJob](#)
 - 新增成员：AppId, Uin, SubAccountUin
- [File](#)
 - 新增成员：CreatedAt, ProjectId
- [Job](#)
 - 新增成员：Type, DomainNameConfig, Debug, AbortReason, CreatedAt, ProjectId, NotificationHooks
- [RequestSummary](#)
 - 新增成员：P99, Status, Result
- [Scenario](#)
 - 新增成员：NotificationHooks

第 3 次发布

发布时间：2022-08-22 06:42:21

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateScenario](#)
 - 新增入参：NotificationHooks

新增数据结构：

- [Notification](#)

第 2 次发布

发布时间：2022-08-15 06:35:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeScenarioWithJobs](#)
 - 新增入参：IgnoreDataset

第 1 次发布

发布时间：2022-07-29 11:04:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AbortCronJobs](#)
- [AbortJob](#)
- [AdjustJobSpeed](#)
- [CopyScenario](#)
- [CreateAlertChannel](#)
- [CreateCronJob](#)
- [CreateFile](#)

- [CreateProject](#)
- [CreateScenario](#)
- [DeleteAlertChannel](#)
- [DeleteCronJobs](#)
- [DeleteFiles](#)
- [DeleteJobs](#)
- [DeleteProjects](#)
- [DeleteScenarios](#)
- [DescribeAlertChannels](#)
- [DescribeAlertRecords](#)
- [DescribeAvailableMetrics](#)
- [DescribeCheckSummary](#)
- [DescribeCronJobs](#)
- [DescribeFiles](#)
- [DescribeJobs](#)
- [DescribeLabelValues](#)
- [DescribeMetricLabelWithValues](#)
- [DescribeNormalLogs](#)
- [DescribeProjects](#)
- [DescribeRegions](#)
- [DescribeRequestSummary](#)
- [DescribeSampleBatchQuery](#)
- [DescribeSampleLogs](#)
- [DescribeSampleMatrixBatchQuery](#)
- [DescribeSampleMatrixQuery](#)
- [DescribeSampleQuery](#)
- [DescribeScenarioWithJobs](#)
- [DescribeScenarios](#)
- [GenerateTmpKey](#)
- [RestartCronJobs](#)
- [StartJob](#)
- [UpdateCronJob](#)
- [UpdateFileScenarioRelation](#)
- [UpdateJob](#)
- [UpdateProject](#)
- [UpdateScenario](#)

新增数据结构：

- [AggregationLegend](#)
- [AlertChannel](#)
- [AlertChannelRecord](#)
- [AlertRecord](#)
- [AlertRecordStatus](#)
- [Attributes](#)
- [CheckSummary](#)
- [Concurrency](#)
- [Credentials](#)
- [CronJob](#)
- [CustomSample](#)
- [CustomSampleMatrix](#)
- [DNSConfig](#)
- [DomainNameConfig](#)
- [File](#)
- [FileInfo](#)
- [Filter](#)
- [GeoRegionsLoadItem](#)
- [HostAlias](#)

- [InternalMetricQuery](#)
- [Job](#)
- [Label](#)
- [LabelWithValues](#)
- [Load](#)
- [LoadSource](#)
- [LoadSpec](#)
- [MetricInfo](#)
- [MetricLabelWithValues](#)
- [NormalLog](#)
- [Project](#)
- [ProtocollInfo](#)
- [ReactionTimeRange](#)
- [RegionDetail](#)
- [RequestSummary](#)
- [RequestsPerSecond](#)
- [SLALabel](#)
- [SLAPolicy](#)
- [SLARule](#)
- [SampleLog](#)
- [SamplePair](#)
- [SampleStream](#)
- [Scenario](#)
- [ScenarioRelatedJobsParams](#)
- [ScenarioWithJobs](#)
- [ScriptInfo](#)
- [ScriptOrigin](#)
- [Stage](#)
- [TagSpec](#)
- [TestData](#)
- [VpcLoadDistribution](#)

简介

最近更新时间：2025-04-25 01:50:57

稳定易用的性能测试工具，稳定易用的性能测试工具描述，帮助快速发现业务性能瓶颈。

API 概览

最近更新时间：2024-08-02 02:00:06

PTS相关接口

接口名称	接口功能	频率限制（次/秒）
AbortCronJobs	停止定时任务	20
AbortJob	停止任务	20
AdjustJobSpeed	调整任务目标RPS	20
CopyScenario	复制场景	20
CreateAlertChannel	创建告警通知接收组	20
CreateCronJob	创建定时任务	20
CreateFile	创建文件	20
CreateProject	创建项目	20
CreateScenario	创建场景	20
DeleteAlertChannel	删除告警通知接收组	20
DeleteCronJobs	删除定时任务	20
DeleteFiles	删除文件	20
DeleteJobs	删除任务	20
DeleteProjects	删除项目	20
DeleteScenarios	删除场景	20
DescribeAlertChannels	查询告警通知接收组	20
DescribeAlertRecords	查询告警历史	20
DescribeAvailableMetrics	查询系统支持的指标	20
DescribeCheckSummary	查询检查点汇总信息	20
DescribeCronJobs	列出定时任务	20
DescribeFiles	查询文件列表	20
DescribeJobs	查询任务列表	20
DescribeLabelValues	查询标签值	20
DescribeMetricLabelWithValues	查询指标所有的label及values值	20
DescribeNormalLogs	查询压测过程日志	20
DescribeProjects	查询项目列表	20
DescribeRegions	查询地域列表	20
DescribeRequestSummary	查询请求汇总信息	20
DescribeSampleBatchQuery	批量查询指标	20
DescribeSampleMatrixBatchQuery	批量查询指标矩阵	30
DescribeSampleMatrixQuery	查询指标矩阵	20
DescribeSampleQuery	查询指标	20
DescribeScenarioWithJobs	查询场景及对应的任务	20
DescribeScenarios	查询场景列表	20
GenerateTmpKey	生成临时COS凭证	20
RestartCronJobs	重启定时任务	20

接口名称	接口功能	频率限制（次/秒）
StartJob	创建并启动任务	20
UpdateCronJob	更新定时任务	20
UpdateFileScenarioRelation	更新关联文件场景	20
UpdateJob	更新任务	20
UpdateProject	更新项目	20
UpdateScenario	更新场景	20
CreateEnvironment	创建环境	20
DeleteEnvironments	删除环境	20
DescribeEnvironments	查看环境列表	20
DescribeErrorSummary	查询错误详情汇总信息	20
DescribeSampleLogs	查询采样日志	30
UpdateEnvironment	更新环境	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 01:27:55

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

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

金融区接入地域	金融区域名
华东地区（上海金融）	pts.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	pts.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：2024-11-15 01:56:47

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

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

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

HTTP GET 请求结构示例:

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

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

HTTP POST（application/json）请求结构示例：

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

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

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

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

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

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

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

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

10
--58731222010402--
```

使用签名方法 v1 的公共参数

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

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

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

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

HTTP GET 请求结构示例：

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

Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例：

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

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

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

签名方法 v3

最近更新时间：2024-12-25 01:54:41

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分例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

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

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

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

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

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

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

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

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

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

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

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

注意：

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

根据以上规则，示例中得到的待签名字符串如下：

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

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

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

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

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

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

2) 计算签名，伪代码如下：

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

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

4. 拼接 Authorization

按如下格式拼接 Authorization：

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

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

根据以上规则，示例中得到的值为：

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

最终完整的调用信息如下：

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

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

⚠ 注意:

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

签名演示

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

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

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

- [Signature Demo](#)

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

Java

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

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

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

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

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

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

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

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

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

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

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

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

System.out.println(authorization);

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

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

Python

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

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

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

# ***** 步骤 1: 拼接规范请求串 *****

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

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

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

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

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

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

Golang

```
package main

import (
"crypto/hmac"
```

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

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

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

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

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

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

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

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

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

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

PHP

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

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

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

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

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

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

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

Ruby

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

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

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

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

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

puts canonical_request

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

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

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

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

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

DotNet

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

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

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

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

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

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

// ***** 步骤 2: 拼接待签名字符串 *****

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

// ***** 步骤 3: 计算签名 *****

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

// ***** 步骤 4: 拼接 Authorization *****

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

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

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

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

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

// 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
// DateTime date = DateTime.UtcNow;
// 注意时区, 建议此时间统一采用UTC时间戳, 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++)
{
    snprintf(buf, sizeof(buf), "%02x", hash[i]);
    NewString = NewString + buf;
}
return NewString;
}

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

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

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

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

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

    return (ss.str());
}

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

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

```
}
return output;
}

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

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

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

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

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

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

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

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

C

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

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

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

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

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

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

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

}

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

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

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

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

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

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

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

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

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

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

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

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

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

其他语言

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

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

签名方法

最近更新时间：2024-12-25 01:54:41

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

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

推荐使用 API Explorer

</> 点击调试

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

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

1. 申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

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

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

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

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

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

2.1. 对参数排序

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

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

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

2.2. 拼接请求字符串

此步骤生成请求字符串。

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

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

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

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

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

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

示例的拼接结果为：

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

2.4. 生成签名串

此步骤生成签名串。

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

具体代码如下，以 PHP 语言为例：

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

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

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

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

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

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

注意：有些编程语言的网路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("evm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

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

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

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

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

PHP

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

ksort($params);

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

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

Ruby

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

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

```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

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

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

DotNet

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

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

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

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

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

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

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

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

NodeJS

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

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

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

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

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

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

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

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

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

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

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

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

返回结果

最近更新时间：2024-03-12 01:44:53

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 06:43:44

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。

PTS相关接口

停止定时任务

最近更新时间：2025-04-25 01:50:57

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

停止定时任务

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：AbortCronJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目ID 示例值：project-xx
CronJobIds.N	是	Array of String	定时任务ID数组 示例值：["cron-xx"]

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 停止定时任务

停止定时任务

输入示例

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

{
  "ProjectId": "project-xx",
  "CronJobIds": [
    "cron-xx"
```

```
]
}
```

输出示例

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

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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](#)。

停止任务

最近更新时间：2025-04-25 01:50:57

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

停止任务

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：AbortJob。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	待停止的压测任务的 ID（所有的压测任务 ID 可以从 DescribeJobs 接口获取） 示例值：job-xx
ProjectId	是	String	项目ID 示例值：project-xx
ScenarioId	是	String	场景ID 示例值：scenario-xx
AbortReason	否	Integer	中断原因 示例值：0 unknown reason; 1 abort by user; 2 abort by sla rules

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 停止任务

输入示例

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

{
  "ProjectId": "project-btksohr0",
  "ScenarioId": "scenario-j1u85ixm",
```

```
"JobId": "job-61ovraba",
"AbortReason": 1
}
```

输出示例

```
{
  "Response": {
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.JobStatusNotRunning	任务状态不是运行中。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

错误码	描述
ResourceNotFound	资源不存在。

调整任务目标RPS

最近更新时间：2025-04-25 01:50:57

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

调整任务的目标RPS

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：AdjustJobSpeed。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-xxx
TargetRequestsPerSecond	是	Integer	目标 RPS。其取值应大于起始 RPS，并且小于最大 RPS 示例值：60

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 调整任务目标RPS

输入示例

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

{
  "JobId": "job-61ovraba",
  "TargetRequestsPerSecond": 10
}
```

输出示例

```
{
  "Response": {
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.ResourceNotFound	资源不存在。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

复制场景

最近更新时间：2025-04-25 01:50:57

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

复制场景

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：CopyScenario。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目 ID 示例值：project-xx
ScenarioId	是	String	场景 ID 示例值：scenario-xx

3. 输出参数

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

4. 示例

示例1 复制场景

输入示例

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

{
  "ProjectId": "project-a5qp5dry",
  "ScenarioId": "scenario-5fjmk132"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx",
    "ScenarioId": "scenario-83osaafm"
  }
}
```

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.ResourceNotFound	资源不存在。

创建告警通知接收组

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建告警通知接收组

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：CreateAlertChannel。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
NoticeId	是	String	monitor 服务的告警通知模板的 NoticeId，可从 monitor 服务的云 API 的 DescribeAlarmNotices 接口响应里的 Id 字段获取。（CreateAlertChannel 接口的入参里用于标识一个告警通知模板的 AMPConsumerId 与 NoticeId 二选一即可） 示例值：notice-xx
ProjectId	是	String	项目 ID 示例值：project-xx
AMPConsumerId	否	String	monitor 服务的告警通知模板的 AMPConsumerId，可从 monitor 服务的云 API 的 DescribeAlarmNotices 接口响应里的 AMPConsumerId 字段获取。（CreateAlertChannel 接口的入参里用于标识一个告警通知模板的 AMPConsumerId 与 NoticeId 二选一即可） 示例值：consumer-xx

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 创建告警通知接收组

输入示例

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

{
  "NoticeId": "notice-f0yvdmq1",
```

```
"ProjectId": "project-btksohr0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

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.DbRecordCreateFailed	数据库创建记录失败。

创建定时任务

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建定时任务

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：CreateCronJob。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Name	是	String	定时任务名字 示例值：cronjob
ProjectId	是	String	项目ID 示例值：project-xx
ScenarioId	是	String	场景ID 示例值：scenario-xx
ScenarioName	是	String	场景名称 示例值：scenario name
FrequencyType	是	Integer	执行频率类型，1:只执行一次; 2:日粒度; 3:周粒度; 4:高级 示例值：1
CronExpression	是	String	cron表达式 示例值：*/1 * * * *
JobOwner	是	String	任务发起人 示例值：tom
EndTime	否	Timestamp ISO8601	结束时间 示例值：2014-08-03T12:00:00+08:00
NoticeId	否	String	Notice ID 示例值：notice-xx
Note	否	String	备注 示例值：定时任务备注

3. 输出参数

参数名称	类型	描述
CronJobId	String	定时任务ID 示例值：cron-xx
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 创建定时任务

创建定时任务

输入示例

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

{
  "ProjectId": "project-xx",
  "ScenarioId": "scenario-xx",
  "ScenarioName": "scenario name",
  "Name": "cron name",
  "CronExpression": "*/1 * * * *",
  "FrequencyType": 1,
  "EndTime": "2020-09-22T00:00:00+00:00",
  "NoticeId": "notice-xx",
  "JobOwner": "tom"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "req-xx",
    "CronJobId": "cron-xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。

创建文件

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建文件

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：CreateFile。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
FileId	是	String	文件 ID。其值应为前序步骤上传该文件到 cos 桶后，文件在 cos 桶中的相应目录 示例值：file-235c3000
ProjectId	是	String	项目 ID 示例值：project-btksohr0
Kind	是	Integer	文件种类，参数文件-1，协议文件-2，请求文件-3 示例值：1
Name	是	String	文件名 示例值：uin.csv
Size	是	Integer	文件大小 示例值：3896
Type	是	String	文件类型，文件夹-folder 示例值：csv
LineCount	否	Integer	行数 示例值：1001
HeadLines.N	否	Array of String	前几行数据 示例值：["uin", "1", "2"]
TailLines.N	否	Array of String	后几行数据 示例值：["998", "999", "1000"]
HeaderInFile	否	Boolean	表头是否在文件内 示例值：true
HeaderColumns.N	否	Array of String	表头 示例值：["uin"]
FileInfos.N	否	Array of FileInfo	文件夹中的文件

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 创建场景与文件的关联关系

输入示例

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

{
  "FileId": "file-235c3000",
  "ProjectId": "project-btksohr0",
  "Kind": 1,
  "Name": "abc.csv",
  "Size": 100,
  "Type": "csv",
  "LineCount": 10,
  "HeadLines": [
    "head-line-content"
  ],
  "TailLines": [
    "tail-line-content"
  ],
  "HeaderInFile": true,
  "HeaderColumns": [
    "header-columns-content"
  ],
  "FileInfos": [
    {
      "Name": "file-name",
      "Size": 0,
      "Type": "file-type",
      "UpdatedAt": "2020-09-22T00:00:00+00:00",
      "FileId": "file-1a2b3c4d"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz"
  }
}
```

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.DbRecordCreateFailed	数据库创建记录失败。

创建项目

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建项目

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：CreateProject。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Name	是	String	项目名 示例值：project name
Description	否	String	项目描述 示例值：this is a test project
Tags.N	否	Array of TagSpec	标签数组

3. 输出参数

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

4. 示例

示例1 创建项目

输入示例

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

{
  "Name": "MyProject",
  "Description": "Test project"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c21cbdvsqxaoc7yp5dxveh3mtpf27c4o",
    "ProjectId": "project-xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
FailedOperation.AccessTagFail	获取标签失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

创建场景

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建场景

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：CreateScenario。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Name	是	String	场景名 示例值：scenario-name
Type	是	String	压测引擎类型 示例值：pts-http, pts-js,pts-jmeter,pts-trpc
ProjectId	是	String	项目ID 示例值：project-xx
Description	否	String	场景描述 示例值：this is a test scenario
Load	否	Load	施压配置
Configs.N	否	Array of String	deprecated 示例值：["a.prop","b.prop"]
Datasets.N	否	Array of TestData	测试数据集
Extensions.N	否	Array of String	deprecated 示例值：["a.jar","b.jar"]
SLAId	否	String	deprecated 示例值：sla-xx
CronId	否	String	cron job ID 示例值：cron-xx
Scripts.N	否	Array of String	deprecated 示例值：["a.js"]
TestScripts.N	否	Array of ScriptInfo	测试脚本文件信息
Protocols.N	否	Array of ProtocolInfo	协议文件路径
RequestFiles.N	否	Array of FileInfo	请求文件路径
SLAPolicy	否	SLAPolicy	SLA 策略
Plugins.N	否	Array of FileInfo	拓展包文件路径

参数名称	必选	类型	描述
DomainNameConfig	否	DomainNameConfig	域名解析配置
Owner	否	String	创建人名 示例值：owner-name

3. 输出参数

参数名称	类型	描述
ScenarioId	String	场景ID 示例值：scenario-xx
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 创建场景

创建场景

输入示例

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

{
  "Load": {
    "GeoRegionsLoadDistribution": [
      {
        "Region": "ap-guangzhou",
        "RegionId": 1,
        "Percentage": 30
      },
      {
        "Region": "ap-shanghai",
        "RegionId": 5,
        "Percentage": 70
      }
    ],
    "LoadSpec": {
      "Concurrency": {
        "Stages": [
          {
            "DurationSeconds": 60,
            "TargetVirtualUsers": 1000
          }
        ]
      }
    },
    "Name": "name",
    "ProjectId": "project-1a2b3c4d",
```

```
"Type": "pts-js"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "ScenarioId": "scenario-1a2b3c4d"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

删除告警通知接收组

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除告警通知接收组

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：DeleteAlertChannel。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目 ID 示例值：project-xx
NoticeId	是	String	待删除的通知渠道的 Notice ID（所有通知渠道的 Notice ID 可以从 DescribeAlertChannels 接口获取） 示例值：notice-xx

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 删除告警通知接收组

输入示例

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

{
  "ProjectId": "project-btksohr0",
  "NoticeId": "notice-ms38of42"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

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](#)。

删除定时任务

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除定时任务

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：DeleteCronJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目ID 示例值：project-xx
CronJobIds.N	是	Array of String	定时任务ID数组 示例值：["cron-xx"]

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 删除定时任务

删除定时任务

输入示例

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

{
  "ProjectId": "project-xx",
  "CronJobIds": [
    "cron-xx"
  ]
}
```

输出示例

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

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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](#)。

删除文件

最近更新时间：2025-04-25 01:50:55

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除文件

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：DeleteFiles。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目 ID 示例值：project-hxmdlc3c
FileIds.N	是	Array of String	待删除的文件的 ID（所有文件 ID 可从接口 DescribeFiles 获取） 示例值：["file-4og13yd4"]

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 删除一个文件

输入示例

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

{
  "ProjectId": "project-btksohr0",
  "FileIds": [
    "file-j1u85ixm"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

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](#)。

删除任务

最近更新时间：2025-04-25 01:50:55

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除任务

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：DeleteJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobIds.N	是	Array of String	待删除的任务的 ID（所有任务的 ID 可以从 DescribeJobs 获取） 示例值：["job-xx"]
ProjectId	是	String	项目ID 示例值：project-xx
ScenarioIds.N	是	Array of String	场景ID数组 示例值：["scenario-xx"]

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 删除一个任务

输入示例

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

{
  "JobIds": [
    "job-61ovraba"
  ],
  "ProjectId": "project-btksohr0",
  "ScenarioIds": [
```

```
"scenario-j1u85ixm"
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

删除项目

最近更新时间：2025-04-25 01:50:55

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除项目

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：DeleteProjects。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectIds.N	是	Array of String	项目ID数组 示例值：["project-xx"]
DeleteScenarios	否	Boolean	是否删除项目相关的场景。默认为否。 示例值：true
DeleteJobs	否	Boolean	是否删除项目相关的任务。默认为否。 示例值：true

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 删除项目

输入示例

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

{
  "ProjectIds": [
    "project-xx"
  ]
}
```

输出示例

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

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

删除场景

最近更新时间：2025-04-25 01:50:55

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除场景

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：DeleteScenarios。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ScenarioIds.N	是	Array of String	场景ID数组 示例值：["scenario-xx"]
ProjectId	是	String	项目ID 示例值：project-xx
DeleteJobs	否	Boolean	是否删除场景相关的任务。默认为否。 示例值：true

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 删除场景

输入示例

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

{
  "ProjectId": "project-xx",
  "DeleteJobs": true,
  "ScenarioIds": [
    "scenario-xx"
  ]
}
```

```

    ]
  }
}

```

输出示例

```

{
  "Response": {
    "RequestId": "req-xx"
  }
}

```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询告警通知接收组

最近更新时间：2025-04-25 01:50:55

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询告警通知接收组

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：DescribeAlertChannels。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectIds.N	是	Array of String	项目 ID 列表 示例值：["project-xx"]
Offset	否	Integer	偏移量，默认为0 示例值：1
Limit	否	Integer	返回数量，默认为20，最大为100 示例值：10
NoticeIds.N	否	Array of String	Notice ID 列表 示例值：["notice-xx"]
OrderBy	否	String	排序项 示例值：CreatedAt
Ascend	否	Boolean	是否正序 示例值：true

3. 输出参数

参数名称	类型	描述
AlertChannelSet	Array of AlertChannelRecord	告警通知接收组列表 注意：此字段可能返回 null，表示取不到有效值。
Total	Integer	告警通知接收组数目 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 查询告警通知接收组列表

输入示例

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

```
{
  "OrderBy": "asc",
  "Ascend": true,
  "Limit": 1,
  "Offset": 1,
  "ProjectIds": [
    "project-xx"
  ]
}
```

输出示例

```
{
  "Response": {
    "AlertChannelSet": [
      {
        "Status": 1,
        "NoticeId": "notice-xx",
        "ProjectId": "projecct-xx",
        "UpdatedAt": "2022-09-09 13:11:12",
        "AMPConsumerId": "consumer-xx",
        "CreatedAt": "2022-09-09 13:11:12",
        "AppId": 11111,
        "Uin": "11100000",
        "SubAccountUin": "22200000"
      }
    ],
    "Total": 1,
    "RequestId": "abc-123-xyz"
  }
}
```

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.DbQueryFailed	数据库查询失败。

查询告警历史

最近更新时间：2025-04-25 01:50:55

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

返回告警历史项的列表

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：DescribeAlertRecords。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectIds.N	是	Array of String	项目 ID 列表 示例值：["project-xx"]
ScenarioIds.N	否	Array of String	场景 ID 列表 示例值：["scenario-xx"]
JobIds.N	否	Array of String	任务 ID 列表 示例值：["job-xx"]
Ascend	否	Boolean	是否正序 示例值：true
OrderBy	否	String	排序项 示例值：CreatedAt
Offset	否	Integer	偏移量，默认为0 示例值：1
Limit	否	Integer	返回数量，默认为20，最大为100 示例值：10
ScenarioNames.N	否	Array of String	按场景名筛选 示例值：["my-scenario"]

3. 输出参数

参数名称	类型	描述
AlertRecordSet	Array of AlertRecord	告警历史 注意：此字段可能返回 null，表示取不到有效值。
Total	Integer	告警历史记录总数 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 查询告警历史

输入示例

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

{
  "OrderBy": "asc",
  "JobIds": [
    "job-1a2b3c4d"
  ],
  "Ascend": true,
  "Limit": 1,
  "Offset": 1,
  "ProjectIds": [
    "project-1a2b3c4d"
  ],
  "ScenarioIds": [
    "scenario-1a2b3c4d"
  ]
}
```

输出示例

```
{
  "Response": {
    "AlertRecordSet": [
      {
        "CreatedAt": "2021-12-22T15:04:36+08:00",
        "UpdatedAt": "2021-12-22T15:04:36+08:00",
        "AppId": 13200000,
        "Uin": "13200000",
        "SubAccountUin": "13200000",
        "AlertRecordId": "alert-1a2b3c4d",
        "ProjectId": "project-1a2b3c4d",
        "ScenarioId": "scenario-1a2b3c4d",
        "ScenarioName": "my-scenario",
        "JobId": "job-1a2b3c4d",
        "JobSLAId": "sla-1a2b3c4d",
        "JobSLADescription": "this is a test",
        "Status": {
          "AbortJob": 1,
          "SendNotice": 1
        },
        "Target": "target-value"
      }
    ],
    "RequestId": "abc-123-xyz",
    "Total": 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

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](#)。

查询系统支持的指标

最近更新时间：2025-04-25 01:50:54

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询系统支持的指标

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：DescribeAvailableMetrics。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。

3. 输出参数

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

4. 示例

示例1 查询系统支持的指标

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "MetricSet": [
      {
        "Aggregations": [
          {
```

```
"Aggregation": "Rate",
"Legend": "请求RPS",
"Unit": "reqs/s"
},
{
"Aggregation": "MaxRateOverTime",
"Legend": "请求RPS峰值",
"Unit": "reqs/s"
},
{
"Aggregation": "Count",
"Legend": "请求总数",
"Unit": "reqs"
},
{
"Aggregation": "ErrorPercentage",
"Legend": "请求失败率",
"Unit": "%"
},
{
"Aggregation": "ErrorPercentageByResult",
"Legend": "请求失败率(结果码)",
"Unit": "%"
}
],
"Alias": "RequestsCount",
"Description": "请求数量",
"InnerMetric": false,
"Metric": "pts_engine_req_total",
"MetricType": "counter",
"Unit": "reqs"
},
{
"Aggregations": [
{
"Aggregation": "Percentiles",
"Legend": "响应时间Percentiles",
"Unit": "s"
},
{
"Aggregation": "Timings",
"Legend": "各阶段平均耗时分布",
"Unit": "s"
},
{
"Aggregation": "Avg",
"Legend": "平均响应时间",
"Unit": "s"
},
{
"Aggregation": "P50",
"Legend": "响应时间P50",
"Unit": "s"
},
{
"Aggregation": "P90",
"Legend": "响应时间P90",

```

```
"Unit": "s"
},
{
  "Aggregation": "P95",
  "Legend": "响应时间P95",
  "Unit": "s"
},
{
  "Aggregation": "P99",
  "Legend": "响应时间P99",
  "Unit": "s"
}
],
"Alias": "RequestDuration",
"Description": "请求响应时间",
"InnerMetric": false,
"Metric": "pts_engine_req_duration_seconds",
"MetricType": "histogram",
"Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Rate",
      "Legend": "检查点RPS",
      "Unit": "checks/s"
    },
    {
      "Aggregation": "Count",
      "Legend": "检查点总数",
      "Unit": "checks"
    },
    {
      "Aggregation": "ErrorPercentage",
      "Legend": "检查点错误率",
      "Unit": "%"
    }
  ],
  "Alias": "ChecksCount",
  "Description": "检查点数量",
  "InnerMetric": false,
  "Metric": "pts_engine_checks_total",
  "MetricType": "counter",
  "Unit": "checks"
},
{
  "Aggregations": [
    {
      "Aggregation": "Gauge",
      "Legend": "并发数",
      "Unit": "VUs"
    },
    {
      "Aggregation": "MaxOverTime",
      "Legend": "并发数峰值",
      "Unit": "VUs"
    }
  ]
}
```

```
],
"Alias": "NumberVirtualUsers",
"Description": "并发用户数",
"InnerMetric": false,
"Metric": "pts_engine_num_vus",
"MetricType": "gauge",
"Unit": "VUs"
},
{
"Aggregations": [
{
"Aggregation": "Rate",
"Legend": "发送速率",
"Unit": "bytes/s"
},
{
"Aggregation": "Count",
"Legend": "发送总数",
"Unit": "bytes"
},
{
"Aggregation": "UsagePercentage",
"Legend": "出口带宽使用率",
"Unit": "%"
}
],
"Alias": "SendBytesCount",
"Description": "发送字节数",
"InnerMetric": false,
"Metric": "pts_engine_send_bytes_total",
"MetricType": "counter",
"Unit": "bytes"
},
{
"Aggregations": [
{
"Aggregation": "Rate",
"Legend": "接收速率",
"Unit": "bytes/s"
},
{
"Aggregation": "Count",
"Legend": "接收总数",
"Unit": "bytes"
},
{
"Aggregation": "UsagePercentage",
"Legend": "入口带宽使用率",
"Unit": "%"
}
],
"Alias": "ReceiveBytesCount",
"Description": "接收字节数",
"InnerMetric": false,
"Metric": "pts_engine_receive_bytes_total",
"MetricType": "counter",
"Unit": "bytes"
```

```
},
{
  "Aggregations": [
    {
      "Aggregation": "Rate",
      "Legend": "场景执行TPS",
      "Unit": "iters/s"
    },
    {
      "Aggregation": "Count",
      "Legend": "场景执行总数",
      "Unit": "iters"
    }
  ],
  "Alias": "IterationsCount",
  "Description": "场景迭代数",
  "InnerMetric": false,
  "Metric": "pts_engine_iterations_total",
  "MetricType": "counter",
  "Unit": "iters"
},
{
  "Aggregations": [
    {
      "Aggregation": "Rate",
      "Legend": "已发送请求RPS",
      "Unit": "reqs/s"
    },
    {
      "Aggregation": "Count",
      "Legend": "已发送请求总数",
      "Unit": "reqs"
    }
  ],
  "Alias": "RequestSentCount",
  "Description": "已发送请求数量",
  "InnerMetric": false,
  "Metric": "pts_engine_req_sent_total",
  "MetricType": "counter",
  "Unit": "reqs"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
      "Legend": "场景执行Percentiles",
      "Unit": "s"
    },
    {
      "Aggregation": "Avg",
      "Legend": "场景执行平均耗时",
      "Unit": "s"
    },
    {
      "Aggregation": "P50",
      "Legend": "场景执行P50",
      "Unit": "s"
    }
  ]
}
```

```

    },
    {
      "Aggregation": "P90",
      "Legend": "场景执行P90",
      "Unit": "s"
    },
    {
      "Aggregation": "P95",
      "Legend": "场景执行P95",
      "Unit": "s"
    },
    {
      "Aggregation": "P99",
      "Legend": "场景执行P99",
      "Unit": "s"
    }
  ],
  "Alias": "IterationDuration",
  "Description": "场景执行耗时",
  "InnerMetric": false,
  "Metric": "pts_engine_iterations_duration_seconds",
  "MetricType": "histogram",
  "Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
      "Legend": "发送耗时Percentiles",
      "Unit": "s"
    },
    {
      "Aggregation": "Avg",
      "Legend": "发送请求平均耗时",
      "Unit": "s"
    },
    {
      "Aggregation": "P50",
      "Legend": "发送耗时P50",
      "Unit": "s"
    },
    {
      "Aggregation": "P90",
      "Legend": "发送耗时P90",
      "Unit": "s"
    },
    {
      "Aggregation": "P95",
      "Legend": "发送耗时P95",
      "Unit": ""
    },
    {
      "Aggregation": "P99",
      "Legend": "发送耗时P99",
      "Unit": "s"
    }
  ],

```

```
"Alias": "RequestSendingDuration",
"Description": "发送请求阶段耗时",
"InnerMetric": false,
"Metric": "pts_engine_req_send_duration_seconds",
"MetricType": "histogram",
"Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
      "Legend": "等待响应Percentiles",
      "Unit": "s"
    },
    {
      "Aggregation": "Avg",
      "Legend": "等待响应平均耗时",
      "Unit": "s"
    },
    {
      "Aggregation": "P50",
      "Legend": "等待响应P50",
      "Unit": "s"
    },
    {
      "Aggregation": "P90",
      "Legend": "等待响应P90",
      "Unit": "s"
    },
    {
      "Aggregation": "P95",
      "Legend": "等待响应P95",
      "Unit": "s"
    },
    {
      "Aggregation": "P99",
      "Legend": "等待响应P99",
      "Unit": "s"
    }
  ],
  "Alias": "RequestWaitingDuration",
  "Description": "等待响应阶段耗时",
  "InnerMetric": false,
  "Metric": "pts_engine_req_wait_duration_seconds",
  "MetricType": "histogram",
  "Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
      "Legend": "下载响应体Percentiles",
      "Unit": "s"
    },
    {
      "Aggregation": "Avg",
      "Legend": "下载响应体平均耗时",
```

```
"Unit": "s"
},
{
  "Aggregation": "P50",
  "Legend": "下载响应体P50",
  "Unit": "s"
},
{
  "Aggregation": "P90",
  "Legend": "下载响应体P90",
  "Unit": "s"
},
{
  "Aggregation": "P95",
  "Legend": "下载响应体P95",
  "Unit": "s"
},
{
  "Aggregation": "P99",
  "Legend": "下载响应体P99",
  "Unit": "s"
}
],
"Alias": "RequestReceivingDuration",
"Description": "下载响应体耗时",
"InnerMetric": false,
"Metric": "pts_engine_req_receive_duration_seconds",
"MetricType": "histogram",
"Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
      "Legend": "等待连接Percentiles",
      "Unit": "s"
    },
    {
      "Aggregation": "Avg",
      "Legend": "等待连接平均耗时",
      "Unit": "s"
    },
    {
      "Aggregation": "P50",
      "Legend": "等待连接P50",
      "Unit": "s"
    },
    {
      "Aggregation": "P90",
      "Legend": "等待连接P90",
      "Unit": "s"
    },
    {
      "Aggregation": "P95",
      "Legend": "等待连接P95",
      "Unit": "s"
    }
  ]
}
```

```
{
  "Aggregation": "P99",
  "Legend": "等待连接P99",
  "Unit": "s"
},
{
  "Alias": "RequestBlockingDuration",
  "Description": "等待连接耗时",
  "InnerMetric": false,
  "Metric": "pts_engine_req_block_duration_seconds",
  "MetricType": "histogram",
  "Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
      "Legend": "建立连接Percentiles",
      "Unit": "s"
    },
    {
      "Aggregation": "Avg",
      "Legend": "建立连接平均耗时",
      "Unit": "s"
    },
    {
      "Aggregation": "P50",
      "Legend": "建立连接P50",
      "Unit": "s"
    },
    {
      "Aggregation": "P90",
      "Legend": "建立连接P90",
      "Unit": "s"
    },
    {
      "Aggregation": "P95",
      "Legend": "建立连接P95",
      "Unit": "s"
    },
    {
      "Aggregation": "P99",
      "Legend": "建立连接P99",
      "Unit": "s"
    }
  ],
  "Alias": "RequestConnectingDuration",
  "Description": "建立连接耗时",
  "InnerMetric": false,
  "Metric": "pts_engine_req_connect_duration_seconds",
  "MetricType": "histogram",
  "Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Percentiles",
```

```
"Legend": "TLS握手Percentiles",
"Unit": "s"
},
{
  "Aggregation": "Avg",
  "Legend": "TLS握手平均耗时",
  "Unit": "s"
},
{
  "Aggregation": "P50",
  "Legend": "TLS握手P50",
  "Unit": "s"
},
{
  "Aggregation": "P90",
  "Legend": "TLS握手P90",
  "Unit": "s"
},
{
  "Aggregation": "P95",
  "Legend": "TLS握手P95",
  "Unit": "s"
},
{
  "Aggregation": "P99",
  "Legend": "TLS握手P99",
  "Unit": "s"
}
],
"Alias": "RequestHandshakingDuration",
"Description": "TLS握手耗时",
"InnerMetric": false,
"Metric": "pts_engine_req_tls_handshake_duration_seconds",
"MetricType": "histogram",
"Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Max",
      "Legend": "最大响应时间",
      "Unit": "s"
    }
  ],
  "Alias": "RequestMaxDuration",
  "Description": "请求最大响应时间",
  "InnerMetric": true,
  "Metric": "pts_engine_req_max_duration_seconds",
  "MetricType": "gauge",
  "Unit": "s"
},
{
  "Aggregations": [
    {
      "Aggregation": "Min",
      "Legend": "最小响应时间",
      "Unit": "s"
    }
  ]
}
```

```
}
],
"Alias": "RequestMinDuration",
"Description": "请求最小响应时间",
"InnerMetric": true,
"Metric": "pts_engine_req_min_duration_seconds",
"MetricType": "gauge",
"Unit": "s"
},
{
"Aggregations": [
{
"Aggregation": "UsagePercentage",
"Legend": "CPU使用率",
"Unit": "%"
}
],
"Alias": "ContainerCPUUsageCount",
"Description": "施压机 CPU 使用量",
"InnerMetric": true,
"Metric": "container_cpu_usage_seconds_total",
"MetricType": "counter",
"Unit": "s"
},
{
"Aggregations": [
{
"Aggregation": "UsagePercentage",
"Legend": "内存使用率",
"Unit": "%"
}
],
"Alias": "ContainerMemUsage",
"Description": "施压机内存使用量",
"InnerMetric": true,
"Metric": "container_memory_usage_bytes",
"MetricType": "gauge",
"Unit": "bytes"
},
{
"Aggregations": [
{
"Aggregation": "Rate",
"Legend": "发送速率",
"Unit": "bytes/s"
},
{
"Aggregation": "Count",
"Legend": "发送总数",
"Unit": "bytes"
}
],
"Alias": "ContainerNetworkSendCount",
"Description": "施压机发送字节数",
"InnerMetric": true,
"Metric": "container_network_transmit_bytes_total",
"MetricType": "counter",
```

```
"Unit": "bytes"
},
{
  "Aggregations": [
    {
      "Aggregation": "Rate",
      "Legend": "接收速率",
      "Unit": "bytes/s"
    },
    {
      "Aggregation": "Count",
      "Legend": "接收总数",
      "Unit": "bytes"
    }
  ],
  "Alias": "ContainerNetworkReceiveCount",
  "Description": "施压机接收字节数",
  "InnerMetric": true,
  "Metric": "container_network_receive_bytes_total",
  "MetricType": "counter",
  "Unit": "bytes"
},
{
  "RequestId": "50233a27-a7c0-4e3e-b598-bf213373aa99"
}
}
```

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](#)。

查询检查点汇总信息

最近更新时间：2025-04-25 01:50:54

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询检查点汇总信息

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：DescribeCheckSummary。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-1a2b3c4d
ScenarioId	是	String	场景ID 示例值：scenario-1a2b3c4d
ProjectId	是	String	项目ID 示例值：project-1a2b3c4d

3. 输出参数

参数名称	类型	描述
CheckSummarySet	Array of CheckSummary	检查点汇总信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询检查点汇总信息

输入示例

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

{
  "ProjectId": "project-1a2b3c4d",
  "ScenarioId": "scenario-1a2b3c4d",
```

```
"JobId": "job-1a2b3c4d"
}
```

输出示例

```
{
  "Response": {
    "CheckSummarySet": [
      {
        "Name": "400",
        "Step": "sample",
        "SuccessCount": 0,
        "FailCount": 92343,
        "ErrorRate": 1
      },
      {
        "Name": "status 200",
        "Step": "sample",
        "SuccessCount": 831084,
        "FailCount": 0,
        "ErrorRate": 0
      }
    ],
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。

错误码	描述
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

列出定时任务

最近更新时间：2025-04-25 01:50:54

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

列出定时任务，非必填数组为空就默认全选

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：DescribeCronJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectIds.N	是	Array of String	项目ID数组 示例值：["project-xx"]
Offset	否	Integer	偏移量，默认为0 示例值：0
Limit	否	Integer	返回数量 示例值：10
CronJobIds.N	否	Array of String	定时任务ID数组 示例值：["cron-xx"]
CronJobName	否	String	定时任务名字，模糊查询 示例值：cron-name
CronJobStatus.N	否	Array of Integer	定时任务状态数组 示例值：[1,2]
OrderBy	否	String	排序的列 示例值：status
Ascend	否	Boolean	是否正序 示例值：true

3. 输出参数

参数名称	类型	描述
Total	Integer	定时任务总数 示例值：10
CronJobSet	Array of CronJob	定时任务信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 DescribeCronJobs

列出定时任务,数组为空默认全选

输入示例

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

{
  "ProjectIds": [
    "project-xx"
  ],
  "CronJobIds": [
    "cron-xx"
  ],
  "OrderBy": "Status",
  "Limit": 10,
  "Ascend": true
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "CronJobSet": [
      {
        "AppId": 1258300000,
        "Uin": "438100000",
        "SubAccountUin": "100026600000",
        "CreatedAt": "2021-09-27T10:46:26+08:00",
        "UpdatedAt": "2021-09-27T10:46:26+08:00",
        "CronJobId": "cron-pybdiltm",
        "Name": "name",
        "ProjectId": "project-1a2b3c4d",
        "ScenarioId": "scenario-1a2b3c4d",
        "ScenarioName": "name",
        "CronExpression": "6 16 * * 1,2,3,4,5,6,7",
        "FrequencyType": 3,
        "EndTime": null,
        "AbortReason": 0,
        "Status": 3,
        "NoticeId": "",
        "Note": "",
        "JobOwner": "owner"
      }
    ],
    "Total": 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
FailedOperation.DbQueryFailed	数据库查询失败。

查询文件列表

最近更新时间：2025-04-25 01:50:54

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询文件列表

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：DescribeFiles。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectIds.N	是	Array of String	项目 ID 数组 示例值：["project-xx"]
FileIds.N	否	Array of String	文件 ID 数组 示例值：["file-xx"]
FileName	否	String	文件名 示例值：uin.csv
Offset	否	Integer	偏移量，默认为 0 示例值：1
Limit	否	Integer	返回数量，默认为 20，最大为 100 示例值：10
Kind	否	Integer	文件种类，参数文件-1，协议文件-2，请求文件-3 示例值：1

3. 输出参数

参数名称	类型	描述
FileSet	Array of File	文件列表 注意：此字段可能返回 null，表示取不到有效值。
Total	Integer	文件总数 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 DescribeFiles

输入示例

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

```
{
  "ProjectIds": [
    "project-xx"
  ],
  "Offset": 0,
  "FileIds": [
    "file-xx"
  ],
  "Limit": 0,
  "FileName": "file name",
  "Kind": 1
}
```

输出示例

```
{
  "Response": {
    "FileSet": [
      {
        "AppId": 1258344690,
        "Uin": "438167612",
        "SubAccountUin": "100019497941",
        "CreatedAt": "2023-04-11T15:20:18+08:00",
        "UpdatedAt": "2023-04-11T15:20:18+08:00",
        "FileId": "file-218d0901",
        "ProjectId": "project-r92xzqe3",
        "Kind": 1,
        "Name": "demo.csv",
        "Size": 18,
        "Type": "CSV",
        "Status": 2,
        "LineCount": 3,
        "HeadLines": [
          "msg\r",
          "world\r",
          "pts"
        ],
        "TailLines": [],
        "HeaderInFile": true,
        "HeaderColumns": [
          "msg"
        ],
        "FileInfos": null,
        "ScenarioSet": null,
        "AppID": 0
      }
    ],
    "RequestId": "7baba29e-5545-44e8-b39e-9be261d6369a",
    "Total": 36
  }
}
```

```
}  
}
```

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](#)。

查询任务列表

最近更新时间：2025-04-25 01:50:53

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询任务列表

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：DescribeJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ScenarioIds.N	是	Array of String	场景ID数组 示例值：["scenario-xx"]
ProjectIds.N	是	Array of String	项目ID数组 示例值：["project-xx"]
Offset	否	Integer	分页起始位置 示例值：0
Limit	否	Integer	每页最大数目 示例值：10
JobIds.N	否	Array of String	任务ID数组 示例值：["job-xx"]
OrderBy	否	String	按字段排序 示例值：id
Ascend	否	Boolean	升序/降序 示例值：true
StartTime	否	Timestamp ISO8601	任务开始时间 示例值：2021-09-08T16:08:50+08:00
EndTime	否	Timestamp ISO8601	任务结束时间 示例值：2021-09-08T16:08:50+08:00
Debug	否	Boolean	调试任务标记 示例值：false
Status.N	否	Array of Integer	任务的状态 示例值：[10,11]

3. 输出参数

参数名称	类型	描述
JobSet	Array of Job	任务列表 注意：此字段可能返回 null，表示取不到有效值。

参数名称	类型	描述
Total	Integer	任务数量 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 查询任务列表

输入示例

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

{
  "ProjectIds": [
    "project-xx"
  ],
  "OrderBy": "asc",
  "JobIds": [
    "job-xx"
  ],
  "Ascend": true,
  "Limit": 0,
  "StartTime": "2020-09-22T00:00:00+00:00",
  "Offset": 0,
  "EndTime": "2020-09-22T00:00:00+00:00",
  "ScenarioIds": [
    "scenario-xx"
  ]
}
```

输出示例

```
{
  "Response": {
    "JobSet": [
      {
        "ErrorRate": 0,
        "RequestTotal": 78,
        "RequestsPerSecond": 3.6055887,
        "ResponseTimeAverage": 0.041749522,
        "ResponseTimeP99": 0.0988,
        "ResponseTimeP95": 0.094,
        "ResponseTimeP90": 0.088,
        "ResponseTimeMax": 0.08245612,
        "ResponseTimeMin": 0.004263497,
        "NetworkReceiveRate": 1842.4558,
        "NetworkSendRate": 773.39874,
        "JobId": "job-1a2b3c4d",
        "ProjectId": "project-1a2b3c4d",
      }
    ]
  }
}
```

```
"ProjectName": "project-name",
"ScenarioId": "scenario-1a2b3c4d",
"ScenarioName": "pts-js (2024-08-07 11:40:53)",
"CronId": "",
"Type": "pts-js",
"Load": {
  "LoadSpec": {
    "Concurrency": {
      "Stages": [
        {
          "DurationSeconds": 120,
          "TargetVirtualUsers": 2
        },
        {
          "DurationSeconds": 120,
          "TargetVirtualUsers": 4
        },
        {
          "DurationSeconds": 120,
          "TargetVirtualUsers": 5
        },
        {
          "DurationSeconds": 240,
          "TargetVirtualUsers": 5
        }
      ],
      "Resources": 1,
      "IterationCount": 0,
      "MaxRequestsPerSecond": 40,
      "GracefulStopSeconds": 3
    },
    "RequestsPerSecond": null,
    "ScriptOrigin": null
  },
  "VpcLoadDistribution": null,
  "GeoRegionsLoadDistribution": [
    {
      "Region": "ap-guangzhou",
      "RegionId": 1,
      "Percentage": 100
    }
  ],
  "TestScripts": null,
  "Datasets": null,
  "Protocols": [],
  "RequestFiles": [],
  "Plugins": [],
  "StartTime": "2024-08-08T14:31:50+08:00",
  "EndTime": "2024-08-08T14:32:12+08:00",
  "Note": "",
  "JobOwner": "job-owner",
  "AbortReason": 1,
  "Status": 16,
  "Message": "用户中断",
  "DomainNameConfig": {
    "HostAliases": null,
```

```
"DNSConfig": null
},
"Debug": false,
"NotificationHooks": null,
"Duration": 600,
"MaxVirtualUserCount": 5,
"MaxRequestsPerSecond": 40,
"LoadSourceInfos": null,
"CreatedAt": "2024-08-08T14:31:44+08:00",
"Configs": null,
"Extensions": null,
"Scripts": null,
"LoadSources": null
}
],
"RequestId": "ae27925c-eb40-47d3-8643-c8c204ab77be",
"Total": 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。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。

错误码	描述
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

查询标签值

最近更新时间：2025-04-25 01:50:53

-

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询标签内容

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：DescribeLabelValues。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-1a2b3c4d
ScenarioId	是	String	场景ID 示例值：scenario-1a2b3c4d
Metric	是	String	指标名。取值范围参见 DescribeMetricLabelWithValues 接口返回的所有指标名 示例值：pts_engine_req_total
LabelName	是	String	标签名。取值范围参见 DescribeMetricLabelWithValues 接口返回的指标及其支持的标签名 示例值：service
ProjectId	是	String	项目ID 示例值：project-1a2b3c4d

3. 输出参数

参数名称	类型	描述
LabelValueSet	Array of String	标签值数组 注意：此字段可能返回 null，表示取不到有效值。 示例值：["http://url1", "http://url2"]
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 查询标签值

输入示例

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

{
  "JobId": "job-61ovraba",
```

```
"ScenarioId": "scenario-j1u85ixm",
"ProjectId": "project-btksohr0",
"Metric": "pts_engine_req_total",
"LabelName": "service"
}
```

输出示例

```
{
  "Response": {
    "LabelValueSet": [
      "http://pets.com/0",
      "http://pets.com/1",
      "http://pets.com/10",
      "http://pets.com/11",
      "http://pets.com/12",
      "http://pets.com/13",
      "http://pets.com/14",
      "http://pets.com/15",
      "http://pets.com/16",
      "http://pets.com/17",
      "http://pets.com/18",
      "http://pets.com/19",
      "http://pets.com/2",
      "http://pets.com/3",
      "http://pets.com/4",
      "http://pets.com/5",
      "http://pets.com/6",
      "http://pets.com/7",
      "http://pets.com/8",
      "http://pets.com/9"
    ],
    "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询指标所有的label及values值

最近更新时间：2025-04-25 01:50:53

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询指标所有的label及values值

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：DescribeMetricLabelWithValues。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	job id 示例值：job-12345
ProjectId	是	String	project id 示例值：project-12345
ScenarioId	是	String	scenario id 示例值：scenario-12345

3. 输出参数

参数名称	类型	描述
MetricLabelWithValuesSet	Array of MetricLabelWithValues	指标所有的label和values数组 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询指标所有的label及values值

查询指标所有的label及values值

输入示例

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

{
  "JobId": "job-jbnipvne",
```

```
"ScenarioId": "scenario-frfvrab4",
"ProjectId": "project-c9drst6a"
}
```

输出示例

```
{
  "Response": {
    "MetricLabelWithValuesSet": [
      {
        "MetricName": "pts_engine_req_total",
        "LabelValuesSet": [
          {
            "LabelName": "service",
            "LabelValues": [
              "http://httpbin.org/get",
              "http://www.baidu.com"
            ]
          },
          {
            "LabelName": "status",
            "LabelValues": [
              "200",
              "502"
            ]
          },
          {
            "LabelName": "result",
            "LabelValues": [
              "502 Bad Gateway",
              "ok"
            ]
          },
          {
            "LabelName": "method",
            "LabelValues": [
              "GET"
            ]
          },
          {
            "LabelName": "proto",
            "LabelValues": [
              "HTTP/1.1"
            ]
          },
          {
            "LabelName": "region",
            "LabelValues": [
              "ap-shanghai"
            ]
          }
        ],
        "MetricName": "pts_engine_req_duration_seconds",
        "LabelValuesSet": [
```

```
{
  "LabelName": "service",
  "LabelValues": [
    "http://httpbin.org/get",
    "http://www.baidu.com"
  ],
},
{
  "LabelName": "status",
  "LabelValues": [
    "200",
    "502"
  ],
},
{
  "LabelName": "result",
  "LabelValues": [
    "502 Bad Gateway",
    "ok"
  ],
},
{
  "LabelName": "method",
  "LabelValues": [
    "GET"
  ],
},
{
  "LabelName": "proto",
  "LabelValues": [
    "HTTP/1.1"
  ],
},
{
  "LabelName": "region",
  "LabelValues": [
    "ap-shanghai"
  ],
},
},
{
  "MetricName": "pts_engine_checks_total",
  "LabelValuesSet": [
    {
      "LabelName": "check",
      "LabelValues": [
        "body.args.name1 equals value1",
        "status is 200"
      ],
    },
    {
      "LabelName": "result",
      "LabelValues": [
        "ok"
      ],
    },
  ],
},
```

```
{
  "LabelName": "region",
  "LabelValues": [
    "ap-shanghai"
  ]
},
{
  "MetricName": "pts_engine_num_vus",
  "LabelValuesSet": []
},
{
  "MetricName": "pts_engine_send_bytes_total",
  "LabelValuesSet": []
},
{
  "MetricName": "pts_engine_receive_bytes_total",
  "LabelValuesSet": []
},
{
  "MetricName": "pts_engine_iterations_total",
  "LabelValuesSet": []
},
{
  "MetricName": "pts_engine_iterations_duration_seconds",
  "LabelValuesSet": []
},
{
  "MetricName": "pts_engine_req_send_duration_seconds",
  "LabelValuesSet": [
    {
      "LabelName": "service",
      "LabelValues": [
        "http://httpbin.org/get",
        "http://www.baidu.com"
      ]
    }
  ]
},
{
  "LabelName": "status",
  "LabelValues": [
    "200",
    "502"
  ]
},
{
  "LabelName": "result",
  "LabelValues": [
    "502 Bad Gateway",
    "ok"
  ]
},
{
  "LabelName": "method",
  "LabelValues": [
    "GET"
  ]
}
```

```
},
{
  "LabelName": "proto",
  "LabelValues": [
    "HTTP/1.1"
  ]
},
{
  "LabelName": "region",
  "LabelValues": [
    "ap-shanghai"
  ]
}
],
{
  "MetricName": "pts_engine_req_wait_duration_seconds",
  "LabelValuesSet": [
    {
      "LabelName": "service",
      "LabelValues": [
        "http://httpbin.org/get",
        "http://www.baidu.com"
      ]
    },
    {
      "LabelName": "status",
      "LabelValues": [
        "200",
        "502"
      ]
    },
    {
      "LabelName": "result",
      "LabelValues": [
        "502 Bad Gateway",
        "ok"
      ]
    },
    {
      "LabelName": "method",
      "LabelValues": [
        "GET"
      ]
    },
    {
      "LabelName": "proto",
      "LabelValues": [
        "HTTP/1.1"
      ]
    },
    {
      "LabelName": "region",
      "LabelValues": [
        "ap-shanghai"
      ]
    }
  ]
}
```

```
]
},
{
  "MetricName": "pts_engine_req_receive_duration_seconds",
  "LabelValuesSet": [
    {
      "LabelName": "service",
      "LabelValues": [
        "http://httpbin.org/get",
        "http://www.baidu.com"
      ]
    },
    {
      "LabelName": "status",
      "LabelValues": [
        "200",
        "502"
      ]
    },
    {
      "LabelName": "result",
      "LabelValues": [
        "502 Bad Gateway",
        "ok"
      ]
    },
    {
      "LabelName": "method",
      "LabelValues": [
        "GET"
      ]
    },
    {
      "LabelName": "proto",
      "LabelValues": [
        "HTTP/1.1"
      ]
    },
    {
      "LabelName": "region",
      "LabelValues": [
        "ap-shanghai"
      ]
    }
  ],
  {
    "MetricName": "pts_engine_req_block_duration_seconds",
    "LabelValuesSet": [
      {
        "LabelName": "service",
        "LabelValues": [
          "http://httpbin.org/get",
          "http://www.baidu.com"
        ]
      },
      {
```

```
"LabelName": "status",
"LabelValues": [
  "200",
  "502"
],
{
  "LabelName": "result",
  "LabelValues": [
    "502 Bad Gateway",
    "ok"
  ]
},
{
  "LabelName": "method",
  "LabelValues": [
    "GET"
  ]
},
{
  "LabelName": "proto",
  "LabelValues": [
    "HTTP/1.1"
  ]
},
{
  "LabelName": "region",
  "LabelValues": [
    "ap-shanghai"
  ]
}
},
{
  "MetricName": "pts_engine_req_connect_duration_seconds",
  "LabelValuesSet": [
    {
      "LabelName": "service",
      "LabelValues": [
        "http://httpbin.org/get",
        "http://www.baidu.com"
      ]
    },
    {
      "LabelName": "status",
      "LabelValues": [
        "200",
        "502"
      ]
    },
    {
      "LabelName": "result",
      "LabelValues": [
        "502 Bad Gateway",
        "ok"
      ]
    }
  ],
}
```

```
{
  "LabelName": "method",
  "LabelValues": [
    "GET"
  ]
},
{
  "LabelName": "proto",
  "LabelValues": [
    "HTTP/1.1"
  ]
},
{
  "LabelName": "region",
  "LabelValues": [
    "ap-shanghai"
  ]
}
},
{
  "MetricName": "pts_engine_req_tls_handshake_duration_seconds",
  "LabelValuesSet": [
    {
      "LabelName": "service",
      "LabelValues": [
        "http://httpbin.org/get",
        "http://www.baidu.com"
      ]
    },
    {
      "LabelName": "status",
      "LabelValues": [
        "200",
        "502"
      ]
    }
  ],
  {
    "LabelName": "result",
    "LabelValues": [
      "502 Bad Gateway",
      "ok"
    ]
  },
  {
    "LabelName": "method",
    "LabelValues": [
      "GET"
    ]
  },
  {
    "LabelName": "proto",
    "LabelValues": [
      "HTTP/1.1"
    ]
  },
  {
```

```
"LabelName": "region",
"LabelValues": [
  "ap-shanghai"
]
}
}
},
"RequestId": "abc-123-xyz"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。

查询压测过程日志

最近更新时间：2025-04-25 01:50:53

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

压测过程日志包括引擎输出日志及用户输出日志

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：DescribeNormalLogs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	压测项目ID 示例值：project-xxx
ScenarioId	是	String	测试场景ID 示例值：scenario-xxx
JobId	是	String	压测任务ID 示例值：job-xxxx
Context	否	String	日志上下文，加载更多日志时使用，透传上次返回的Context值，获取后续的日志内容。过期时间1小时 示例值：1234-5678
From	否	Timestamp ISO8601	日志开始时间 示例值：2021-09-02T03:18:35+08:00
To	否	Timestamp ISO8601	日志结束时间 示例值：2021-09-02T03:18:35+08:00
SeverityText	否	String	日志级别，可取debug/info/error 示例值：debug/info/error
Instance	否	String	施压节点IP 示例值：192.168.9.9
InstanceRegion	否	String	施压节点所在地域 示例值：ap-guangzhou/ap-shanghai
LogType	否	String	日志类型，console代表用户输出，engine代表引擎输出 示例值：console/engine
Limit	否	Integer	返回日志条数限制，最大100 示例值：50

3. 输出参数

参数名称	类型	描述
Context	String	日志上下文，加载更多日志时使用，透传上次返回的Context值，获取后续的日志内容。过期时间1小时 示例值：123-456-789

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

4. 示例

示例1 查询压测过程日志

查询压测过程日志

输入示例

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

{
  "ProjectId": "project-fjs583kq",
  "ScenarioId": "scenario-q63bkqsa",
  "JobId": "job-657v4tmi",
  "LogType": "console"
}
```

输出示例

```
{
  "Response": {
    "Context": "123-456-789",
    "NormalLogs": [],
    "RequestId": "nes5mdwkfrbfhbpjrs4ox6ck41xjn9jp"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

查询项目列表

最近更新时间：2025-04-25 01:50:53

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询项目列表

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：DescribeProjects。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Offset	否	Integer	分页offset 示例值：0
Limit	否	Integer	每页limit 示例值：10
ProjectIds.N	否	Array of String	项目ID数组 示例值：["project-xx"]
ProjectName	否	String	项目名 示例值：project-xx
OrderBy	否	String	按字段排序 示例值：id
Ascend	否	Boolean	升序/降序 示例值：true
TagFilters.N	否	Array of TagSpec	标签数组

3. 输出参数

参数名称	类型	描述
ProjectSet	Array of Project	项目数组 注意：此字段可能返回 null，表示取不到有效值。
Total	Integer	项目数量 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 查询项目列表

查询项目列表

输入示例

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

{
  "ProjectIds": [
    "project-xx"
  ]
}
```

输出示例

```
{
  "Response": {
    "ProjectSet": [
      {
        "AppId": 123,
        "Uin": "456",
        "SubAccountUin": "789",
        "CreatedAt": "2021-08-23T20:59:07+08:00",
        "UpdatedAt": "2021-08-23T20:59:07+08:00",
        "ProjectId": "project-ks8lr6m2",
        "Name": "MyProject",
        "Description": "Test project",
        "Tags": [
          {
            "TagKey": "owner",
            "TagValue": "lyla"
          }
        ],
        "Status": 1
      }
    ],
    "RequestId": "abc",
    "Total": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.UnauthorizedOperation	当前请求未经CAM授权。
FailedOperation	操作失败。
FailedOperation.AccessTagFail	获取标签失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
FailedOperation.SendRequest	请求发送失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

查询地域列表

最近更新时间：2025-04-25 01:50:53

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询地域列表

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：DescribeRegions。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
LoadType	否	Integer	通过该参数指定不同压测网络环境，在不同网络环境下，PTS可用的地域不一样 示例值：0: 公网压测可用地域， 1: VPC私有网络压测可用地域， 2: DevCloud环境可用地域

3. 输出参数

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

4. 示例

示例1 查询地域列表

输入示例

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

{
  "LoadType": 0
}
```

输出示例

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

```
"RegionSet": [
  {
    "CreatedAt": "2021-08-20T16:33:00+08:00",
    "UpdatedAt": "2021-08-20T16:33:00+08:00",
    "RegionId": 1,
    "Region": "ap-guangzhou",
    "RegionShortName": "gz",
    "RegionState": 1,
    "RegionName": "广州",
    "Area": "华南地区"
  },
  {
    "CreatedAt": "2021-08-20T16:33:00+08:00",
    "UpdatedAt": "2021-08-20T16:33:00+08:00",
    "RegionId": 37,
    "Region": "ap-shenzhen",
    "RegionShortName": "szz",
    "RegionState": 1,
    "RegionName": "深圳",
    "Area": "华南地区"
  }
],
"RequestId": "abc-123-xyz"
}
```

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.DbQueryFailed	数据库查询失败。

错误码	描述
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

查询请求汇总信息

最近更新时间：2025-04-25 01:50:53

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询请求汇总信息

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：DescribeRequestSummary。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	压测任务ID 示例值：job-1a2b3c4d
ScenarioId	是	String	压测场景ID 示例值：scenario-12345
ProjectId	是	String	压测项目ID 示例值：project-12345

3. 输出参数

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

4. 示例

示例1 查询请求汇总信息

输入示例

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

{
  "JobId": "job-kufs4qhc",
  "ScenarioId": "scenario-ih54i7dm",
```

```
"ProjectId": "project-pf8s0daq"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "RequestSummarySet": [
      {
        "Average": 0.0426436516818182,
        "Count": 25,
        "ErrorPercentage": 0,
        "Max": 0.077895098,
        "Method": "GET",
        "Min": 0.007491527,
        "P90": 0.08830936954286554,
        "P95": 0.09415468477143277,
        "P99": 0,
        "RPS": 1.8336389398233037,
        "Result": "",
        "Service": "http://mockhttpbin.pts.svc.cluster.local/get",
        "Status": ""
      }
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。

批量查询指标

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

批量查询指标，返回固定时间点指标内容

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：DescribeSampleBatchQuery。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	压测任务的 ID 示例值：job-i39wgjys
ScenarioId	是	String	场景的 ID 示例值：scenario-4og13yd4
Queries.N	是	Array of InternalMetricQuery	查询指标数组
ProjectId	是	String	项目的 ID 示例值：project-hxmdlc3c

3. 输出参数

参数名称	类型	描述
MetricSampleSet	Array of CustomSample	返回指标内容 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 批量查询指标

查询请求速率&请求总数

输入示例

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

{
```

```
"ProjectId": "project-btksohr0",
"JobId": "job-61ovraba",
"ScenarioId": "scenario-j1u85ixm",
"Queries": [
{
"Metric": "pts_engine_req_total",
"Aggregation": "Rate"
},
{
"Metric": "pts_engine_req_total",
"Aggregation": "Count"
}
]
}
```

输出示例

```
{
"Response": {
"MetricSampleSet": [
{
"Metric": "pts_engine_req_total",
"Aggregation": "Rate",
"Unit": "reqs/s",
"Labels": null,
"Value": "30.34732915216155",
"Timestamp": 1631694009709
},
{
"Metric": "pts_engine_req_total",
"Aggregation": "Count",
"Unit": "reqs",
"Labels": null,
"Value": "1193277",
"Timestamp": 1631694009709
}
],
"RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

批量查询指标矩阵

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

批量查询指标矩阵

默认接口请求频率限制：30次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSampleMatrixBatchQuery。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-12345
ProjectId	是	String	项目ID 示例值：project-12345
ScenarioId	是	String	场景ID 示例值：scenario-12345
Queries.N	是	Array of InternalMetricQuery	查询语句
MaxPoint	否	Integer	最多返回的数据点个数 示例值：500

3. 输出参数

参数名称	类型	描述
MetricSampleMatrixSet	Array of CustomSampleMatrix	批量指标矩阵 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 DescribeSampleMatrixBatchQuery

输入示例

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

```
{
  "JobId": "job-6pmj0oae",
  "ScenarioId": "scenario-oienak6y",
  "Queries": [
    {
      "Metric": "pts_engine_req_duration_seconds",
      "Aggregation": "P50"
    }
  ],
  "ProjectId": "project-m51m6h0a"
}
```

输出示例

```
{
  "Response": {
    "MetricSampleMatrixSet": [
      {
        "Metric": "pts_engine_req_duration_seconds",
        "Aggregation": "P50",
        "Unit": "s",
        "Streams": [
          {
            "Name": "响应时间P50",
            "Labels": null,
            "Values": [
              {
                "Timestamp": 1650517296000,
                "Value": 0.032142349260523326
              },
              {
                "Timestamp": 1650517344000,
                "Value": 0.025
              },
              {
                "Timestamp": 1650518490000,
                "Value": 0.05
              }
            ]
          }
        ]
      }
    ],
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。

查询指标矩阵

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询指标矩阵

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：DescribeSampleMatrixQuery。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-12345
ProjectId	是	String	项目ID 示例值：project-12345
ScenarioId	是	String	场景ID 示例值：scenario-12345
Metric	是	String	指标名。取值范围参见 DescribeMetricLabelWithValues 接口返回的所有指标名 示例值：pts_engine_req_total
Aggregation	是	String	聚合函数。取值范围：Rate,Count,Avg,P90,P95,P99,Gauge 示例值：Rate
Filters.N	否	Array of Filter	用标签过滤规则来过滤指标，规则中包含标签名 LabelName、标签值 LabelValue、操作符 Operator（0代表相等，1代表不等） 示例值：{ "LabelName": "method", "LabelValue": "GET", "Operator": 0 }
GroupBy.N	否	Array of String	分组；取值范围参见 DescribeMetricLabelWithValues 接口返回的指标及其支持的标签名 示例值：["service", "result", "status"]
MaxPoint	否	Integer	返回的最大数据点个数 示例值：500

3. 输出参数

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

4. 示例

示例1 查询指标矩阵

输入示例

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

{
  "ProjectId": "project-btksohr0",
  "JobId": "job-61ovraba",
  "ScenarioId": "scenario-j1u85ixm",
  "Metric": "pts_engine_req_duration_seconds",
  "Aggregation": "P50"
}
```

输出示例

```
{
  "Response": {
    "MetricSampleMatrix": {
      "Metric": "pts_engine_req_duration_seconds",
      "Aggregation": "P50",
      "Unit": "s",
      "Streams": [
        {
          "Name": "响应时间P50",
          "Labels": null,
          "Values": [
            {
              "Timestamp": 1650517296000,
              "Value": 0.032142349260523326
            },
            {
              "Timestamp": 1650517299000,
              "Value": 0.03377059214626565
            },
            {
              "Timestamp": 1650517302000,
              "Value": 0.03206904981580744
            },
            {
              "Timestamp": 1650518490000,
              "Value": 0.05
            }
          ]
        }
      ],
      "RequestId": "70805f6a-d7e1-4247-9d5a-fde3afe2b377"
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。

查询指标

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询指标，返回固定时间点指标内容

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：DescribeSampleQuery。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	job id 示例值：job-1a2b3c4d
ScenarioId	是	String	场景Id 示例值：scenario-1a2b3c4d
Metric	是	String	指标名。取值范围参见 DescribeMetricLabelWithValues 接口返回的所有指标名 示例值：pts_engine_req_total
Aggregation	是	String	聚合函数。取值范围：Rate,Count,Avg,P90,P95,P99,Gauge 示例值：Rate
ProjectId	是	String	项目ID 示例值：project-1a2b3c4d
Labels.N	否	Array of Label	标签过滤条件。各指标支持的标签参见 DescribeMetricLabelWithValues 接口返回的所有指标及其支持的标签

3. 输出参数

参数名称	类型	描述
MetricSample	CustomSample	返回指标内容 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询总请求数

查询某个任务总的请求数

输入示例

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

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

{
  "ProjectId": "project-1a2b3c4d",
  "JobId": "job-1a2b3c4d",
  "ScenarioId": "scenario-1a2b3c4d",
  "Metric": "pts_engine_req_total",
  "Aggregation": "Count"
}
```

输出示例

```
{
  "Response": {
    "MetricSample": {
      "Metric": "pts_engine_req_total",
      "Aggregation": "Count",
      "Unit": "reqs",
      "Labels": [
        {
          "LabelName": "method",
          "LabelValue": "GET"
        }
      ],
      "Value": "1100000",
      "Timestamp": 1631693000000
    },
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询场景及对应的任务

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询场景配置并附带已经执行的任务内容

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：DescribeScenarioWithJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Offset	否	Integer	偏移量，默认为0 示例值：1
Limit	否	Integer	返回数量，默认为20，最大为100 示例值：10
ProjectIds.N	否	Array of String	项目ID数组 示例值：["project-xx"]
ScenarioIds.N	否	Array of String	场景ID数组 示例值：["scenario-xx"]
ScenarioName	否	String	场景名 示例值：scenario name
ScenarioStatus	否	Integer	场景状态数组 示例值：[1,2]
OrderBy	否	String	排序的列 示例值：status
Ascend	否	Boolean	是否正序 示例值：true/false
ScenarioRelatedJobsParams	否	ScenarioRelatedJobsParams	job相关参数
IgnoreScript	否	Boolean	是否需要返回场景的脚本内容 示例值：true
IgnoreDataset	否	Boolean	是否需要返回测试数据文件信息 示例值：true
ScenarioType	否	String	场景类型，如pts-http, pts-js, pts-trpc, pts-jmeter 示例值：pts-xx
Owner	否	String	创建人员 示例值：owner

3. 输出参数

参数名称	类型	描述
ScenarioWithJobsSet	Array of ScenarioWithJobs	场景配置以及附带的job内容 注意：此字段可能返回 null，表示取不到有效值。
Total	Integer	场景总数 示例值：3
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询场景及对应的job内容

查询场景及对应的job内容

输入示例

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

{
  "ScenarioIds": [
    "scenario-1a2b3c4d"
  ],
  "ScenarioRelatedJobsParams": {
    "OrderBy": "StartTime",
    "Limit": 10,
    "Ascend": true
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "ScenarioWithJobsSet": [
      {
        "Scenario": {
          "AppId": 251201585,
          "Uin": "700000153395",
          "SubAccountUin": "700000153395",
          "CreatedAt": "2021-09-27T10:46:26+08:00",
          "UpdatedAt": "2021-09-27T10:46:26+08:00",
          "ScenarioId": "scenario-1a2b3c4d",
          "Name": "hello",
          "Type": "pts-js",
          "Load": {
            "LoadSpec": {
              "Concurrency": {
                "Stages": [
                  {
                    "DurationSeconds": 60,
                    "TargetVirtualUsers": 10
                  }
                ]
              }
            }
          }
        }
      }
    ]
  }
}
```

```
}
],
"IterationCount": 5,
"MaxRequestsPerSecond": 1000,
"GracefulStopSeconds": 3,
"Resources": 1
},
"RequestsPerSecond": null,
"ScriptOrigin": null
},
"VpcLoadDistribution": {
"Region": "ap-guangzhou",
"RegionId": 1,
"VpcId": "1",
"SubnetIds": null
},
"GeoRegionsLoadDistribution": [
{
"Region": "1",
"RegionId": 1,
"Percentage": 1
}
]
},
"EncodedScripts": "",
"Description": "",
"Status": 2,
"Configs": [
"a.prop"
],
"Datasets": null,
"Extensions": [
"a.jar"
],
"ProjectId": "project-1a2b3c4d",
"ProjectName": "name",
"SLAId": "",
"CronId": "",
"TestScripts": null,
"Protocols": [],
"RequestFiles": [],
"Plugins": [],
"SLAPolicy": {
"SLARules": [],
"AlertChannel": {
"NoticeId": "",
"AMPConsumerId": ""
}
},
"DomainNameConfig": {
"HostAliases": null,
"DNSConfig": null
},
"NotificationHooks": null,
"Owner": ""
},
"Jobs": [
```

```
{
  "ErrorRate": 0,
  "RequestTotal": 0,
  "RequestsPerSecond": 0,
  "ResponseTimeAverage": 0,
  "ResponseTimeP99": 0,
  "ResponseTimeP95": 0,
  "ResponseTimeP90": 0,
  "ResponseTimeMax": 0,
  "ResponseTimeMin": 0,
  "NetworkReceiveRate": 0,
  "NetworkSendRate": 0,
  "JobId": "job-1a2b3c4d",
  "ProjectId": "project-1a2b3c4d",
  "ProjectName": "name",
  "ScenarioId": "scenario-1a2b3c4d",
  "ScenarioName": "name",
  "CronId": "",
  "Type": "pts-js",
  "Load": {
    "LoadSpec": {
      "Concurrency": {
        "Stages": [
          {
            "DurationSeconds": 60,
            "TargetVirtualUsers": 10
          }
        ],
        "IterationCount": 5,
        "MaxRequestsPerSecond": 1000,
        "Resources": 1,
        "GracefulStopSeconds": 3
      },
      "RequestsPerSecond": null,
      "ScriptOrigin": null
    },
    "VpcLoadDistribution": null,
    "GeoRegionsLoadDistribution": [
      {
        "Region": "ap-guangzhou",
        "RegionId": 1,
        "Percentage": 100
      }
    ],
    "TestScripts": null,
    "Datasets": null,
    "Protocols": [],
    "RequestFiles": [],
    "Plugins": [],
    "StartTime": "2021-09-27T10:46:26+08:00",
    "EndTime": "2021-09-27T10:46:26+08:00",
    "Note": "",
    "JobOwner": "owner",
    "AbortReason": 0,
    "Status": 12,
    "Message": ""
  }
```

```
"DomainNameConfig": {
  "HostAliases": null,
  "DNSConfig": null
},
"Debug": false,
"NotificationHooks": null,
"Duration": 600,
"MaxVirtualUserCount": 1,
"MaxRequestsPerSecond": 8,
"LoadSourceInfos": null,
"CreatedAt": "2021-09-27T10:46:26+08:00",
"Configs": null,
"Extensions": null,
"Scripts": null,
"LoadSources": null
}
}
}
},
"Total": 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。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
InternalError	内部错误。

错误码	描述
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询场景列表

最近更新时间：2025-04-25 01:50:52

-

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询场景列表

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：DescribeScenarios。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ScenarioIds.N	否	Array of String	场景ID数组 示例值：["scenario-xx"]
ScenarioName	否	String	场景名 示例值：scenario-name
ScenarioStatus.N	否	Array of Integer	场景状态数组 示例值：[1,2]
Offset	否	Integer	偏移量，默认为0 示例值：1
Limit	否	Integer	返回数量，默认为20，最大为100 示例值：10
OrderBy	否	String	排序的列 示例值：status
Ascend	否	Boolean	是否正序 示例值：true
ProjectIds.N	否	Array of String	项目ID数组 示例值：["project-xx"]
ScenarioType	否	String	场景类型 示例值：pts-js

3. 输出参数

参数名称	类型	描述
ScenarioSet	Array of Scenario	场景列表 注意：此字段可能返回 null，表示取不到有效值。
Total	Integer	场景总数 示例值：3
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询场景列表

输入示例

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

```
{
  "ScenarioIds": [
    "scenario-xx"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "ScenarioSet": [
      {
        "AppId": 251008000,
        "Configs": null,
        "CreatedAt": "2024-12-25T11:43:42+08:00",
        "CronId": "",
        "Datasets": [],
        "Description": "",
        "DomainNameConfig": {
          "DNSConfig": null,
          "HostAliases": null
        },
        "EncodedScripts": "",
        "Extensions": null,
        "Load": {
          "GeoRegionsLoadDistribution": [
            {
              "Percentage": 80,
              "Region": "ap-guangzhou",
              "RegionId": 1
            },
            {
              "Percentage": 20,
              "Region": "ap-shanghai",
              "RegionId": 4
            }
          ],
          "LoadSpec": {
            "Concurrency": {
              "GracefulStopSeconds": 3,
              "IterationCount": 0,
              "MaxRequestsPerSecond": 40,
              "Resources": 5,
              "Stages": [
                {
                  "DurationSeconds": 60,
                  "TargetVirtualUsers": 5
                }
              ]
            }
          }
        }
      }
    ]
  }
}
```

```

},
{
  "DurationSeconds": 0,
  "TargetVirtualUsers": 5
}
],
{
  "RequestsPerSecond": null,
  "ScriptOrigin": null
},
{
  "VpcLoadDistribution": null
},
{
  "Name": "pts-js(2024-12-25 11:43:40)",
  "NotificationHooks": null,
  "Owner": "",
  "Plugins": [],
  "ProjectId": "project-1a2b3c4d",
  "ProjectName": "name",
  "Protocols": [],
  "RequestFiles": [],
  "SLAId": "",
  "SLAPolicy": {
    "AlertChannel": {
      "AMPConsumerId": "",
      "NoticeId": ""
    },
    "SLARules": []
  },
  "ScenarioId": "scenario-1a2b3c4d",
  "Status": 2,
  "SubAccountUin": "700000200000",
  "TestScripts": [
    {
      "EncodedContent": "EncodedContent",
      "EncodedHttpArchive": "",
      "FileId": "",
      "LoadWeight": 100,
      "Name": "script.js",
      "Size": 896,
      "Type": "js",
      "UpdatedAt": "2024-12-26T10:30:34+08:00"
    }
  ],
  "Type": "pts-js",
  "Uin": "3510500000",
  "UpdatedAt": "2024-12-26T10:30:57+08:00"
}
],
{
  "Total": 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。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

生成临时COS凭证

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

生成临时COS凭证

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：GenerateTmpKey。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目ID 示例值：project-xx
ScenarioId	否	String	场景ID 示例值：scenario-xx

3. 输出参数

参数名称	类型	描述
StartTime	Integer	临时访问凭证获取时刻的时间戳（单位秒） 示例值：1630376705
ExpiredTime	Integer	临时访问凭证超时 时刻的时间戳（单位秒） 示例值：1630377005
Credentials	Credentials	临时访问凭证
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 生成临时COS凭证

输入示例

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

{
  "ProjectId": "project-xx",
```

```
"ScenarioId": "scenario-xx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "StartTime": 1630376705,
    "ExpiredTime": 1630377005,
    "Credentials": {
      "TmpSecretId": "tmp-secretid-value",
      "TmpSecretKey": "tmp-secretkey-value",
      "Token": "token-value"
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

重启定时任务

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

重启状态为已中止的定时任务

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：RestartCronJobs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目ID 示例值：project-xx
CronJobIds.N	是	Array of String	定时任务ID数组 示例值：["cron-xx"]

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 重启定时任务

重启状态为已中止的定时任务

输入示例

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

{
  "ProjectId": "project-xx",
  "CronJobIds": [
    "cron-xx"
  ]
}
```

输出示例

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

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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](#)。

创建并启动任务

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建并启动任务

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：StartJob。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ScenarioId	是	String	场景ID 示例值：scenario-xx
JobOwner	是	String	任务发起人 示例值：tom
ProjectId	是	String	项目ID 示例值：project-xx
Debug	否	Boolean	是否调试 示例值：true
Note	否	String	备注 示例值：备注

3. 输出参数

参数名称	类型	描述
JobId	String	任务ID 示例值：job-xx
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 创建并启动任务

创建并启动任务

输入示例

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

<公共请求参数>

```
{
  "ProjectId": "project-xx",
  "JobOwner": "xiaohong",
  "ScenarioId": "scenario-xx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "req-xx",
    "JobId": "job-xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.NoTasksInJob	任务中没有task。
FailedOperation.NotSupportedInEnv	当前环境不支持。
FailedOperation.ResourceNotFound	资源不存在。

错误码	描述
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceInsufficient	资源不足。

更新定时任务

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

更新定时任务

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：UpdateCronJob。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目ID 示例值：project-xx
CronJobId	是	String	定时任务ID 示例值：cron-xx
Note	是	String	备注 示例值：备注
CronExpression	是	String	cron表达式 示例值：*/1 * * * *
FrequencyType	是	Integer	执行频率类型，1:只执行一次；2:日粒度；3:周粒度；4:高级 示例值：1
Name	是	String	定时任务名字 示例值：cronjob
ScenarioId	是	String	场景ID 示例值：scenario-xx
ScenarioName	是	String	场景名称 示例值：scenario name
JobOwner	是	String	任务发起人 示例值：tom
EndTime	否	Timestamp ISO8601	结束时间 示例值：2014-08-03T12:00:00+08:00
NoticeId	否	String	Notice ID 示例值：notice-xx

3. 输出参数

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

4. 示例

示例1 更新定时任务

更新定时任务

输入示例

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

{
  "ProjectId": "project-xx",
  "CronJobId": "cron-xx",
  "CronExpression": "*/1 * * * *",
  "FrequencyType": 1,
  "EndTime": "2020-09-22T00:00:00+00:00",
  "NoticeId": "notice-xx",
  "Name": "cron name",
  "ScenarioId": "scenario-xx",
  "ScenarioName": "scenario name",
  "Note": "notes",
  "JobOwner": "tom"
}
```

输出示例

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

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。

更新关联文件场景

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

更新关联文件场景

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：UpdateFileScenarioRelation。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
FileId	是	String	文件 ID。其值应为前序步骤上传该文件到 cos 桶后，文件在 cos 桶中的相应目录 示例值：file-03874f00
ProjectId	是	String	项目 ID 示例值：project-btksohr0
ScenarioIds.N	否	Array of String	场景 ID 数组 示例值：["scenario-j1u85ixm"]

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 重新关联文件与场景

输入示例

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

{
  "ProjectId": "project-ksqyndry",
  "ScenarioIds": [
    "scenario-jp98fx4g"
  ],
}
```

```
"FileId": "file-03874f00"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "9139da45-84a5-4212-aff6-a05fd1a2e7f7"
  }
}
```

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.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。

更新任务

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

更新任务

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：UpdateJob。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-xx
ProjectId	是	String	项目ID 示例值：project-xx
ScenarioId	是	String	场景ID 示例值：scenario-xx
Note	否	String	任务备注信息 示例值：this is a test job

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 创建任务

输入示例

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

{
  "ProjectId": "project-xx",
  "ScenarioId": "scenario-xx",
```

```
"Note": "this is my note",
"JobId": "job-xx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

更新项目

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

更新项目

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：UpdateProject。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	项目ID 示例值：project-xx
Name	否	String	项目名 示例值：project name
Description	否	String	项目描述 示例值：this is a test project
Status	否	Integer	项目状态，默认传递1 示例值：1
Tags.N	否	Array of TagSpec	标签数组

3. 输出参数

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

4. 示例

示例1 更新项目

输入示例

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

{
  "ProjectId": "project-1a2b3c4d",
```

```
"Name": "project name"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.AccessTagFail	获取标签失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

更新场景

最近更新时间：2025-04-25 01:50:50

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

更新场景

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：UpdateScenario。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ScenarioId	是	String	场景ID 示例值：scenario-1a2b3c4d
Name	否	String	场景名；调用该接口时，请将 Scenario 中不需要修改的字段保持原样也作为接口的入参，否则场景可能会不可用。 示例值：scenario name
Description	否	String	场景描述 示例值：this is a test scenario
Type	否	String	压测场景的模式类型。取值范围：pts-http 代表简单模式，pts-js 代表脚本模式，pts-jmeter 代表 JMeter 模式。 示例值：pts-http
Load	否	Load	施压配置
EncodedScripts	否	String	deprecated 示例值：WyJhYmMiLCAiZGVmIl0K
Configs.N	否	Array of String	deprecated 示例值：["a.prop","b.prop"]
Datasets.N	否	Array of TestData	测试数据集
Extensions.N	否	Array of String	deprecated 示例值：["a.jar","b.jar"]
SLAId	否	String	SLA规则ID 示例值：sla-1a2b3c4d
CronId	否	String	cron job ID 示例值：cron-1a2b3c4d
Status	否	Integer	场景状态（注：现已无需传递该参数） 示例值：2
ProjectId	否	String	项目ID 示例值：project-1a2b3c4d
TestScripts.N	否	Array of ScriptInfo	测试脚本路径
Protocols.N	否	Array of ProtocolInfo	协议文件路径
RequestFiles.N	否	Array of FileInfo	请求文件路径
SLAPolicy	否	SLAPolicy	SLA 策略
Plugins.N	否	Array of FileInfo	拓展包文件路径
DomainNameConfig	否	DomainNameConfig	域名解析配置
NotificationHooks.N	否	Array of Notification	WebHook请求配置

参数名称	必选	类型	描述
Owner	否	String	创建人名 示例值：owner-name
EnvId	否	String	环境ID 示例值：env-1a2b3c4d

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）。定位问题时需提供该次请求的 RequestId。

4. 示例

示例1 更新场景

更新场景

输入示例

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

{
  "ScenarioId": "scenario-1a2b3c4d",
  "Name": "pts-js(2023-05-12 16:12:30)",
  "Description": "",
  "Type": "pts-js",
  "Load": {
    "LoadSpec": {
      "Concurrency": {
        "IterationCount": 0,
        "MaxRequestsPerSecond": 0,
        "Stages": [
          {
            "DurationSeconds": 120,
            "TargetVirtualUsers": 2
          },
          {
            "DurationSeconds": 120,
            "TargetVirtualUsers": 4
          },
          {
            "DurationSeconds": 120,
            "TargetVirtualUsers": 5
          },
          {
            "DurationSeconds": 240,
            "TargetVirtualUsers": 5
          }
        ],
        "Resources": 1
      }
    }
  }
}
```

输出示例

5. 开发者资源

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 API 服务。

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。

版权所有：腾讯云计算（北京）有限责任公司

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

创建环境

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

创建环境

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：CreateEnvironment。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。

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 创建环境

创建环境

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz"
  }
}
```

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 API 服务。

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

删除环境

最近更新时间：2025-04-25 01:50:56

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

删除环境

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：DeleteEnvironments。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。

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 删除环境

删除环境

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查看环境列表

最近更新时间：2025-04-25 01:50:54

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查看环境列表

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：DescribeEnvironments。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。

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 环境列表查询

环境列表查询

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询错误详情汇总信息

最近更新时间：2025-04-25 01:50:54

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询错误详情汇总信息

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：DescribeErrorSummary。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务ID 示例值：job-1a2b3c4d
ScenarioId	是	String	场景ID 示例值：scenario-1a2b3c4d
ProjectId	是	String	项目ID 示例值：project-1a2b3c4d
Filters.N	否	Array of Filter	过滤参数

3. 输出参数

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

4. 示例

示例1 查询错误汇总信息

查询错误汇总信息

输入示例

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

{
```

```
"ProjectId": "project-xx",
"JobId": "job-xx",
"ScenarioId": "scenario-xx",
"Filters": []
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz",
    "ErrorSummarySet": [
      {
        "Status": "404",
        "Result": "404 Not Found",
        "Proto": "HTTP/1.1",
        "Count": 17,
        "Rate": 47.222222222222,
        "Message": "该状态码表明，服务器找不到客户端请求的资源"
      }
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。

错误码	描述
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询采样日志

最近更新时间：2025-04-25 01:50:52

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

查询采样日志

默认接口请求频率限制：30次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSampleLogs。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ProjectId	是	String	测试项目ID 示例值：project-xxx
ScenarioId	是	String	测试场景ID 示例值：scenario-xxx
JobId	是	String	测试任务ID 示例值：job-xxxx
Context	否	String	加载更多日志时使用，透传上次返回的Context值，获取后续的日志内容。过期时间1小时，不与 Offset 参数同时使用 示例值：csdfeee-dsflI09ds-sdsf
From	否	Timestamp ISO8601	日志开始时间 示例值：2021-09-02T03:18:35+08:00
To	否	Timestamp ISO8601	日志结束时间 示例值：2021-09-02T03:18:35+08:00
SeverityText	否	String	日志级别debug,info,error 示例值：info
InstanceRegion	否	String	地域 示例值：ap-shanghai
Instance	否	String	施压引擎节点IP 示例值：192.168.0.1
LogType	否	String	request 代表采样日志,engine 代表引擎日志，console 代表用户打印日志 示例值：request/engine/console
Offset	否	Integer	日志偏移量，不与Context 参数同时使用 示例值：10
Limit	否	Integer	返回日志条数，最大100 示例值：50
ReactionTimeRange	否	ReactionTimeRange	采样日志响应时间范围
Status	否	String	采样请求状态码 示例值：200
Result	否	String	采样请求结果码

参数名称	必选	类型	描述
			示例值：ok
Method	否	String	采样请求方法 示例值：Get
Service	否	String	采样服务API 示例值：http://www.baidu.com

3. 输出参数

参数名称	类型	描述
Total	Integer	日志总数 示例值：100
Context	String	日志上下文，加载更多日志时使用，透传上次返回的Context值，获取后续的日志内容。过期时间1小时 示例值：2340-324234-32423
SampleLogs	Array of SampleLog	采样日志数组 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询采样日志

查询采样日志

输入示例

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

{
  "ProjectId": "project-fjs583kq",
  "ScenarioId": "scenario-q63bkqsa",
  "JobId": "job-657v4tmi",
  "LogType": "request"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "85f22814-36b9-44e0-bfaa-18fad9acc8f8",
    "Total": 10,
    "Context": "12353-2432423",
    "SampleLogs": []
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。

更新环境

最近更新时间：2025-04-25 01:50:51

1. 接口描述

接口请求域名：pts.tencentcloudapi.com。

更新环境

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：UpdateEnvironment。
Version	是	String	公共参数 ，本接口取值：2021-07-28。
Region	否	String	公共参数 ，本接口不需要传递此参数。

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 更新环境

更新环境

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "abc-123-xyz"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.ResourceNotFound	资源不存在。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

数据结构

最近更新时间：2025-05-20 01:26:48

AggregationLegend

聚合函数

被如下接口引用：DescribeAvailableMetrics。

名称	类型	描述
Aggregation	String	指标支持的聚合函数 示例值：Rate
Legend	String	聚合函数作用于指标后对应的描述 示例值：请求RPS
Unit	String	聚合之后的指标单位 示例值：reqs/s

AlertChannel

告警通知渠道

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
Noticeld	String	否	通知模板ID 示例值：notice-xx
AMPConsumerId	String	否	AMP consumer ID 示例值：consumer-xx

AlertChannelRecord

告警通知接收组

被如下接口引用：DescribeAlertChannels。

名称	类型	必选	描述
Noticeld	String	否	Notice ID 示例值：notice-xx
AMPConsumerId	String	否	Consumer ID 示例值：consumer-xx
ProjectId	String	否	项目 ID 示例值：project-xx
Status	Integer	否	状态 示例值：1
CreatedAt	String	否	创建时间 示例值：2021-09-02T03:18:35+08:00
UpdatedAt	String	否	更新时间 示例值：2021-09-02T03:18:35+08:00
AppId	Integer	否	App ID 示例值：251200000
Uin	String	否	主账号 示例值：700000400000

名称	类型	必选	描述
SubAccountUin	String	否	子账号 示例值：700000400000

AlertRecord

告警历史记录项

被如下接口引用：DescribeAlertRecords。

名称	类型	必选	描述
AlertRecordId	String	否	告警历史记录项 ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：alert-xxx
ProjectId	String	否	项目 ID 示例值：project-xx
ScenarioId	String	否	场景 ID 示例值：scenario-xx
Status	AlertRecordStatus	否	状态
CreatedAt	String	否	创建时间 示例值：2021-09-02T03:18:35+08:00
UpdatedAt	String	否	修改时间 示例值：2021-09-02T03:18:35+08:00
JobId	String	否	任务 ID 示例值：job-xx
AppId	Integer	否	App ID 示例值：251200000
Uin	String	否	主账号 示例值：700000400000
SubAccountUin	String	否	子账号 示例值：700000400000
ScenarioName	String	否	场景名称 示例值：scenario-name
Target	String	否	告警对象 示例值：target-value
JobSLAId	String	否	告警规则 ID 示例值：sla-xx
JobSLADescription	String	否	告警规则描述 示例值：SLA规则: 请求的错误百分比 > 5.00 % 当前值: 7.61 %

AlertRecordStatus

告警历史项的状态

被如下接口引用：DescribeAlertRecords。

名称	类型	必选	描述
AbortJob	Integer	否	停止压测任务成功与否 示例值：1
SendNotice	Integer	否	发送告警通知成功与否 示例值：1

Attributes

采样日志附带属性

被如下接口引用：DescribeSampleLogs。

名称	类型	描述
Status	String	采用请求返回码 示例值：200
Result	String	采样请求结果码 示例值：ok
Service	String	采样请求API 示例值：http://www.baidu.com
Method	String	采样请求调用方法 示例值：get
Duration	String	采样请求延时时间 示例值：20

CheckSummary

检查点汇总结果

被如下接口引用：DescribeCheckSummary。

名称	类型	描述
Name	String	检查点名字 示例值：status is ok
Step	String	检查点所在步骤名字 示例值：buy dog
SuccessCount	Integer	检查点成功次数 示例值：99
FailCount	Integer	检查失败次数 示例值：1
ErrorRate	Float	错误比例 示例值：0.01

Concurrency

并发模式的施压配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
Stages	Array of Stage	否	多阶段配置数组 注意：此字段可能返回 null，表示取不到有效值。
IterationCount	Integer	否	运行次数 示例值：10
MaxRequestsPerSecond	Integer	否	最大RPS 示例值：1000
GracefulStopSeconds	Integer	否	优雅终止任务的等待时间 示例值：3
Resources	Integer	否	资源数 示例值：6

Credentials

COS临时凭证

被如下接口引用：GenerateTmpKey。

名称	类型	描述
TmpSecretId	String	临时secret ID 示例值：tmp-secret-id-value
TmpSecretKey	String	临时secret key 示例值：tmp-secret-key-value
Token	String	临时token 示例值：tmp-token-value

CronJob

定时任务

被如下接口引用：DescribeCronJobs。

名称	类型	描述
CronJobId	String	定时任务ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：cron-xx
Name	String	定时任务名字 示例值：cron-name
ProjectId	String	项目ID 示例值：project-xx
ScenarioId	String	场景ID 示例值：scenario-xx
ScenarioName	String	场景名称 示例值：scenario-name
CronExpression	String	cron 表达式 示例值：*/1 ****
EndTime	Timestamp ISO8601	结束时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2014-08-03T12:00:00+08:00
AbortReason	Integer	中止原因 示例值：1
Status	Integer	定时任务状态 示例值：1
NoticeId	String	Notice ID 示例值：notice-xx
CreatedAt	Timestamp ISO8601	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2014-08-03T12:00:00+08:00
UpdatedAt	Timestamp ISO8601	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2014-08-03T12:00:00+08:00
FrequencyType	Integer	执行频率类型 示例值：1
Note	String	备注 示例值：备注

名称	类型	描述
JobOwner	String	tom 示例值：任务发起人
AppId	Integer	App ID 示例值：251000000
Uin	String	主账号 示例值：3510500000
SubAccountUin	String	子账号 示例值：3510500000

CustomSample

sample附带原始查询语句中的metric, aggregation

被如下接口引用：DescribeSampleBatchQuery, DescribeSampleQuery。

名称	类型	描述
Metric	String	指标名 示例值：pts_engine_req_total
Aggregation	String	聚合条件 示例值：Rate
Labels	Array of Label	过滤条件 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"LabelName":"Method", "LabelValue":"GET"}]
Value	Float	查询值 示例值：123.34
Timestamp	Integer	Time is the number of milliseconds since the epoch // (1970-01-01 00:00 UTC) excluding leap seconds. 示例值：12345433
Unit	String	指标对应的单位，当前单位有：s,bytes,bytes/s,reqs,reqs/s,checks,checks/s,itors,itors/s,VUs, % 示例值：ms
Name	String	指标序列名字 示例值：请求RPS

CustomSampleMatrix

指标矩阵，可包含多条指标序列

被如下接口引用：DescribeSampleMatrixBatchQuery, DescribeSampleMatrixQuery。

名称	类型	描述
Metric	String	指标名字 示例值：pts_engine_req_total
Aggregation	String	聚合函数 示例值：Rate
Unit	String	指标单位 示例值：reqs/s
Streams	Array of SampleStream	指标序列数组 注意：此字段可能返回 null，表示取不到有效值。
Step	Integer	两个时间点的时间间隔，单位纳秒 示例值：100

DNSConfig

施压机 DNS 配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
Nameservers	Array of String	否	DNS IP 列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：["0.0.0.0"]

DomainNameConfig

施压机的域名解析相关配置

被如下接口引用：CreateScenario, DescribeFiles, DescribeJobs, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
HostAliases	Array of HostAlias	否	域名绑定配置 注意：此字段可能返回 null，表示取不到有效值。
DNSConfig	DNSConfig	否	DNS 配置 注意：此字段可能返回 null，表示取不到有效值。

ErrorSummary

错误信息汇总

被如下接口引用：DescribeErrorSummary。

名称	类型	描述
Status	String	状态码 示例值：200
Result	String	结果码 示例值：200 ok
Count	Integer	错误出现次数 示例值：1000
Rate	Float	错误率 示例值：0.11
Message	String	错误信息 示例值：成功
Proto	String	请求协议类型 示例值：HTTP/1.1

File

文件列表

被如下接口引用：DescribeFiles。

名称	类型	描述
FileId	String	文件 ID 示例值：file-xx
Kind	Integer	文件种类，参数文件-1，协议文件-2，请求文件-3 示例值：1
Name	String	文件名 示例值：uin.csv
Size	Integer	文件字节数 示例值：3896

名称	类型	描述
Type	String	文件类型 示例值：csv
UpdatedAt	Timestamp ISO8601	更新时间 示例值：2006-01-02 15:04:05
LineCount	Integer	文件行数 示例值：1001
HeadLines	Array of String	头部数据行 注意：此字段可能返回 null，表示取不到有效值。 示例值：["uin", "1", "2"]
TailLines	Array of String	尾部数据行 注意：此字段可能返回 null，表示取不到有效值。 示例值：["998", "999", "1000"]
HeaderInFile	Boolean	首行是否为参数名 示例值：true
HeaderColumns	Array of String	参数名数组 注意：此字段可能返回 null，表示取不到有效值。 示例值：["uin"]
FileInfos	Array of FileInfo	文件夹中的文件 注意：此字段可能返回 null，表示取不到有效值。
ScenarioSet	Array of Scenario	关联场景 注意：此字段可能返回 null，表示取不到有效值。
Status	Integer	文件状态 示例值：1
CreatedAt	Timestamp ISO8601	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2006-01-02 15:04:05
ProjectId	String	项目 ID 示例值：project-xx
AppID	Integer	此字段不再使用 示例值：251000000
Uin	String	用户主账号 示例值：3510500000
SubAccountUin	String	用户子账号 示例值：3510500000
AppId	Integer	用户账号的 App ID 示例值：251000000

FileInfo

文件基本信息

被如下接口引用：CreateFile, CreateScenario, DescribeFiles, DescribeJobs, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
Name	String	否	文件名 示例值：a.log
Size	Integer	否	文件大小 示例值：100
Type	String	否	文件类型 示例值：csv

名称	类型	必选	描述
UpdatedAt	Timestamp ISO8601	否	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-09-02T03:18:35+08:00
FileId	String	否	文件 ID 示例值：file-xx

Filter

指标查询过滤

被如下接口引用：DescribeErrorSummary, DescribeSampleBatchQuery, DescribeSampleMatrixBatchQuery, DescribeSampleMatrixQuery。

名称	类型	必选	描述
Operator	Integer	是	等于：0，不等于：1 示例值：0
LabelName	String	是	标签名，可选值包括： 1. method：请求方法名； 2. proto：协议名； 3. service：服务名； 4. status：响应状态码； 5. result：响应详情； 6. check：检查名。 示例值：service
LabelValue	String	是	标签值： 1. method：请求方法名，以 http 协议为例，method 为 GET、POST、PUT 等； 2. proto：协议名，以 http 协议为例，proto 为 HTTP/1.1、HTTP/2 等； 3. service：服务名，以 http 协议为例，service 为请求 url，如 http://httpbin.org/get 等； 4. status：响应状态码，以 http 协议为例，状态码包括 200、404、500 等； 5. result：响应详情，通过 result 判断请求成功或失败；请求正常，result 标签值为 ok；请求失败，result 标签携带错误码和描述； 6. check：检查名，标签值为用户设置的检查点名称。 示例值：www.baidu.com

GeoRegionsLoadItem

压力分布配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
RegionId	Integer	是	地域ID 示例值：1
Region	String	否	地域 示例值：ap-guangzhou
Percentage	Integer	否	百分比 示例值：50

HostAlias

施压机域名绑定配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
HostNames	Array of String	否	需绑定的域名列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：["example.com"]

名称	类型	必选	描述
IP	String	否	需绑定的 IP 地址 示例值：0.0.0.0

InternalMetricQuery

查询结构封装

被如下接口引用：DescribeSampleBatchQuery, DescribeSampleMatrixBatchQuery。

名称	类型	必选	描述
Metric	String	是	指标名 示例值：pts_engine_req_total
Aggregation	String	是	聚合函数。取值范围：Rate,Count,Avg,P90,P95,P99,Gauge 示例值：Rate
Labels	Array of Label	否	deprecated, 请使用Filters 示例值：[{"LabelName":"Method", "LabelValue":"GET"}]
Filters	Array of Filter	否	用标签过滤规则来过滤指标，规则中包含标签名 LabelName、标签值 LabelValue、操作符 Operator（0代表相等，1代表不等） 示例值：[{"LabelName":"Method","Operator":0, "LabelValue":"GET"}]
GroupBy	Array of String	否	指标分组 示例值：["status"]

Job

任务

被如下接口引用：DescribeJobs, DescribeScenarioWithJobs。

名称	类型	必选	描述
JobId	String	否	任务的JobID 注意：此字段可能返回 null，表示取不到有效值。 示例值：job-xx
ScenarioId	String	否	任务的场景ID 示例值：scenario-xx
Load	Load	否	任务的施压配置
Configs	Array of String	否	此字段不再使用 注意：此字段可能返回 null，表示取不到有效值。 示例值：["a.prop", "b.prop"]
Datasets	Array of TestData	否	任务的数据集文件 注意：此字段可能返回 null，表示取不到有效值。
Extensions	Array of String	否	此字段不再使用 注意：此字段可能返回 null，表示取不到有效值。 示例值：["a.jar", "b.jar"]
Status	Integer	否	任务的运行状态, JobUnknown: 0, JobCreated:1, JobPending:2, JobPreparing:3, JobS JobRunning:11, JobFinished:12, JobPrepareException:13, JobFinishException:14, JobSelectClusterException:19, JobCreateTaskException:20, JobSyncTaskExcep 示例值：1
StartTime	Timestamp ISO8601	否	任务的开始时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-09-08T16:08:50+08:00
EndTime	Timestamp ISO8601	否	任务的结束时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-09-08T16:08:50+08:00

名称	类型	必选	描述
MaxVirtualUserCount	Integer	否	任务的最大VU数 示例值：10
Note	String	否	任务的备注信息 示例值：this is a test
ErrorRate	Float	否	错误率百分比 示例值：10.5
JobOwner	String	否	任务发起人 示例值：tom
LoadSources	LoadSource	否	此字段不再使用 注意：此字段可能返回 null，表示取不到有效值。
Duration	Integer	否	任务时长 示例值：10
MaxRequestsPerSecond	Integer	否	最大每秒请求数 示例值：1000
RequestTotal	Float	否	总请求数 示例值：1000
RequestsPerSecond	Float	否	平均每秒请求数 示例值：1000
ResponseTimeAverage	Float	否	平均响应时间 示例值：1
ResponseTimeP99	Float	否	响应时间第99百分位 示例值：1
ResponseTimeP95	Float	否	响应时间第95百分位 示例值：1
ResponseTimeP90	Float	否	响应时间第90百分位 示例值：1
Scripts	Array of String	否	此字段不再使用 注意：此字段可能返回 null，表示取不到有效值。 示例值：["a.js", "b.js"]
ResponseTimeMax	Float	否	最大响应时间 示例值：1
ResponseTimeMin	Float	否	最小响应时间 示例值：1
LoadSourceInfos	Array of LoadSource	否	发压host信息 注意：此字段可能返回 null，表示取不到有效值。
TestScripts	Array of ScriptInfo	否	测试脚本信息 注意：此字段可能返回 null，表示取不到有效值。
Protocols	Array of ProtocolInfo	否	协议脚本信息 注意：此字段可能返回 null，表示取不到有效值。
RequestFiles	Array of FileInfo	否	请求文件信息 注意：此字段可能返回 null，表示取不到有效值。
Plugins	Array of FileInfo	否	拓展包文件信息 注意：此字段可能返回 null，表示取不到有效值。
CronId	String	否	定时任务ID 示例值：cron-xx
Type	String	否	场景类型 示例值：js

名称	类型	必选	描述
DomainNameConfig	DomainNameConfig	否	域名绑定配置
Debug	Boolean	否	false 示例值：是否为调试型的压测任务
AbortReason	Integer	否	中断原因 示例值：1
CreatedAt	Timestamp ISO8601	否	任务的创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2022-08-09T10:27:22+08:00
ProjectId	String	否	项目ID 示例值：project-xx
NotificationHooks	Array of NotificationHook	否	通知事件回调 注意：此字段可能返回 null，表示取不到有效值。
NetworkReceiveRate	Float	否	每秒接收字节数 示例值：123.456
NetworkSendRate	Float	否	每秒发送字节数 示例值：123.456
Message	String	否	任务状态描述 示例值：已完成
ProjectName	String	否	test-project 示例值：project-name
ScenarioName	String	否	test-scenario 示例值：scenario-name
PayMode	Integer	否	付费类型 示例值：1 预付费，2 后付费，3 免费调试任务
Usage	Integer	否	job VUM用量 示例值：1000

Label

包含labelName 和labelValue

被如下接口引用：DescribeSampleBatchQuery, DescribeSampleMatrixBatchQuery, DescribeSampleMatrixQuery, DescribeSampleQuery。

名称	类型	必选	描述
LabelName	String	是	标签名 示例值：service
LabelValue	String	是	标签值 示例值：http://mockhttpbin.pts.svc.cluster.local/get

LabelWithValues

标签及对应的值

被如下接口引用：DescribeMetricLabelWithValues。

名称	类型	描述
LabelName	String	标签名称 示例值：service
LabelValues	Array of String	标签值 示例值：["500", "200"]

Load

施压配置

被如下接口引用：CreateScenario, DescribeFiles, DescribeJobs, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
LoadSpec	LoadSpec	否	施压配置
VpcLoadDistribution	VpcLoadDistribution	否	压力来源 注意：此字段可能返回 null，表示取不到有效值。
GeoRegionsLoadDistribution	Array of GeoRegionsLoadItem	否	多地域压力分布 注意：此字段可能返回 null，表示取不到有效值。

LoadSource

发压host来源

被如下接口引用：DescribeJobs。

名称	类型	必选	描述
IP	String	否	发压host的IP 示例值：1.1.1.1
PodName	String	否	发压host所在的pod 示例值：pod-name-value
Region	String	否	所属地域 示例值：ap-guangzhou

LoadSpec

施压配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
Concurrency	Concurrency	否	并发施压模式的配置 注意：此字段可能返回 null，表示取不到有效值。
RequestsPerSecond	RequestsPerSecond	否	RPS施压模式的配置 注意：此字段可能返回 null，表示取不到有效值。
ScriptOrigin	ScriptOrigin	否	脚本内置压力模式 注意：此字段可能返回 null，表示取不到有效值。

MetricInfo

指标结构

被如下接口引用：DescribeAvailableMetrics。

名称	类型	描述
Metric	String	后台指标 示例值：pts_engine_reqs_total
Alias	String	前台展示指标名称 示例值：RequestCount
Description	String	指标描述 示例值：请求数量
MetricType	String	指标类型 示例值：Counter

名称	类型	描述
Unit	String	默认指标单位 示例值：s
Aggregations	Array of AggregationLegend	指标支持的聚合函数
InnerMetric	Boolean	是否内部指标，内部指标不可在前台提供用户自由选择 示例值：true

MetricLabelWithValues

PTS提供的指标名，指标对应的labels及values

被如下接口引用：DescribeMetricLabelWithValues。

名称	类型	描述
MetricName	String	metric 名字 示例值：RequestCount
LabelValuesSet	Array of LabelWithValues	label及values 数组

NormalLog

通用日志

被如下接口引用：DescribeNormalLogs。

名称	类型	描述
Timestamp	String	毫秒时间戳 示例值：23423423
SeverityText	String	日志级别 示例值：debug/info/error
Body	String	日志输出内容 示例值：this is log body

Notification

测试启动前后的消息通知

被如下接口引用：UpdateScenario。

名称	类型	必选	描述
Events	Array of String	否	发生事件 示例值：["name", "age"]
URL	String	否	webhook的网址 示例值："http://abc.com"

NotificationHook

事件通知回调

被如下接口引用：DescribeFiles, DescribeJobs, DescribeScenarios。

名称	类型	必选	描述
Events	Array of String	否	通知事件 注意：此字段可能返回 null，表示取不到有效值。 示例值：["BeforeRunning", "AfterFinished", "AfterAborted"]
URL	String	否	回调 URL

名称	类型	必选	描述
			示例值：http://xxx.com/notify

Project

项目

被如下接口引用：DescribeProjects。

名称	类型	描述
ProjectId	String	项目ID 示例值：project-xx
Name	String	项目名 示例值：project-name
Description	String	项目描述 示例值：this is a test project
Tags	Array of TagSpec	标签数组 注意：此字段可能返回 null，表示取不到有效值。
Status	Integer	项目状态 示例值：1
CreatedAt	Timestamp ISO8601	创建时间 示例值：2021-08-23T20:59:07+08:00
UpdatedAt	Timestamp ISO8601	修改时间 示例值：2021-08-23T20:59:07+08:00
AppId	Integer	App ID 示例值：251000000
Uin	String	用户ID 示例值：3510500000
SubAccountUin	String	子用户ID 示例值：3510500000

ProtocolInfo

协议文件详情

被如下接口引用：CreateScenario, DescribeFiles, DescribeJobs, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
Name	String	否	协议详情 示例值：hello.proto
Size	Integer	否	文件大小 示例值：100
Type	String	否	文件类型 示例值：proto
UpdatedAt	Timestamp ISO8601	否	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-09-02T03:18:35+08:00
FileId	String	否	文件 ID 示例值：file-xx

ReactionTimeRange

采用日志响应时间RT范围

被如下接口引用：DescribeSampleLogs。

名称	类型	必选	描述
Min	String	否	最小响应时间，单位ms 示例值：10
Max	String	否	最大响应时间，单位ms 示例值：100

RegionDetail

地域

被如下接口引用：DescribeRegions。

名称	类型	必选	描述
Region	String	是	地域代码 示例值：ap-guangzhou
RegionId	Integer	是	地域ID 示例值：1
Area	String	否	地域所在的地区 示例值：华南地区
RegionName	String	否	地域名称 示例值：广州
RegionState	Integer	否	地域状态 示例值：1
RegionShortName	String	否	地域简称 示例值：gz
CreatedAt	Timestamp ISO8601	否	创建时间 示例值：2021-08-20T16:33:00+08:00
UpdatedAt	Timestamp ISO8601	否	更新时间 示例值：2021-08-20T16:33:00+08:00

RequestSummary

压测请求明细

被如下接口引用：DescribeRequestSummary。

名称	类型	描述
Service	String	请求URL 示例值：http://www.baidu.com
Method	String	请求方法 示例值：GET
Count	Integer	请求次数 示例值：100
Average	Float	请求响应平均耗时，单位秒 示例值：0.50
P90	Float	请求p90耗时，单位秒 示例值：0.90
P95	Float	请求p95耗时，单位秒 示例值：0.95
Min	Float	请求最小耗时，单位秒 示例值：0.10

名称	类型	描述
Max	Float	请求最大耗时，单位秒 示例值：0.99
ErrorPercentage	Float	请求错误率 示例值：1，代表错误率1%
P99	Float	请求p99耗时，单位秒 示例值：0.55
Status	String	响应状态码 示例值：200
Result	String	响应详情 示例值：ok
RPS	Float	平均rps，单位req/s 示例值：100

RequestsPerSecond

RPS模式的施压配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
MaxRequestsPerSecond	Integer	否	最大RPS 示例值：1000
DurationSeconds	Integer	否	施压时间 示例值：60
TargetVirtualUsers	Integer	否	deprecated 示例值：30
Resources	Integer	否	资源数 示例值：6
StartRequestsPerSecond	Integer	否	起始RPS 示例值：500
TargetRequestsPerSecond	Integer	否	目标RPS，入参无效 示例值：500
GracefulStopSeconds	Integer	否	优雅关停的等待时间 示例值：3

SLALabel

SLA 标签

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
LabelName	String	否	标签名 示例值：service
LabelValue	String	否	标签值 示例值：http://httpbin.org/get

SLAPolicy

SLA 策略

被如下接口引用：CreateScenario, DescribeFiles, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
SLARules	Array of SLARule	否	SLA 规则 注意：此字段可能返回 null，表示取不到有效值。
AlertChannel	AlertChannel	否	告警通知渠道 注意：此字段可能返回 null，表示取不到有效值。

SLARule

SLA 规则

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
Metric	String	否	压测指标 示例值：pts_engine_req_total
Aggregation	String	否	压测指标聚合方法 示例值：avg
Condition	String	否	压测指标条件判断符号 示例值：>
Value	Float	否	阈值 示例值：1.00
LabelFilter	Array of SLALabel	否	标签 注意：此字段可能返回 null，表示取不到有效值。
AbortFlag	Boolean	否	是否停止压测任务 示例值：true
For	String	否	持续时长 示例值：1m

SampleLog

采样日志

被如下接口引用：DescribeSampleLogs。

名称	类型	描述
Timestamp	String	日志毫秒时间戳 示例值：3242342332432
Attributes	Attributes	采样日志属性
Body	String	har格式的采样请求 示例值：{"log":{"version":"","creator":{"name":"PTS-har","version":""},"entries":[]}}

SamplePair

sample采样值

被如下接口引用：DescribeSampleMatrixBatchQuery, DescribeSampleMatrixQuery。

名称	类型	描述
Timestamp	Integer	is the number of milliseconds since the epoch (1970-01-01 00:00 UTC) excluding leap seconds. 示例值：2342342
Value	Float	is a representation of a value for a given sample at a given time. 示例值：3242.23

SampleStream

连续指标采样内容

被如下接口引用：DescribeSampleMatrixBatchQuery, DescribeSampleMatrixQuery。

名称	类型	描述
Labels	Array of Label	labels描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"LabelName":"Method", "LabelValue":"GET"}]
Values	Array of SamplePair	指标采样数组 示例值：[{"Timestamp":24355, "Value":3243.45}]
Name	String	指标序列名字 示例值：请求RPS

Scenario

场景列表

被如下接口引用：DescribeFiles, DescribeScenarioWithJobs, DescribeScenarios。

名称	类型	描述
ScenarioId	String	场景ID 示例值：scenario-xx
Name	String	场景名 示例值：scenario-test
Description	String	场景描述 示例值：this is a test scenario
Type	String	场景类型，如pts-http, pts-js, pts-trpc, pts-jmeter 示例值：pts-http
Status	Integer	场景状态 示例值：1
Load	Load	施压配置
EncodedScripts	String	deprecated 示例值：WyJhYmMiLCAiZGVmZyJdCg==
Configs	Array of String	deprecated 注意：此字段可能返回 null，表示取不到有效值。 示例值：["a.prop", "b.prop"]
Extensions	Array of String	deprecated 注意：此字段可能返回 null，表示取不到有效值。 示例值：["a.jar", "b.jar"]
Datasets	Array of TestData	测试数据集 注意：此字段可能返回 null，表示取不到有效值。
SLAId	String	SLA规则的ID 示例值：sla-xx
CronId	String	Cron Job规则的ID 示例值：cron-xx
CreatedAt	String	场景创建时间 示例值：2021-08-16 15:09:56
UpdatedAt	String	场景修改时间 示例值：2021-08-16 15:09:56
ProjectId	String	项目ID 示例值：project-xx
AppId	Integer	App ID

名称	类型	描述
		示例值：251000000
Uin	String	用户ID 示例值：3510500000
SubAccountUin	String	子用户ID 示例值：3510500000
TestScripts	Array of ScriptInfo	测试脚本信息 注意：此字段可能返回 null，表示取不到有效值。
Protocols	Array of ProtocolInfo	协议文件信息 注意：此字段可能返回 null，表示取不到有效值。
RequestFiles	Array of FileInfo	请求文件信息 注意：此字段可能返回 null，表示取不到有效值。
SLAPolicy	SLAPolicy	SLA 策略
Plugins	Array of FileInfo	扩展包信息 注意：此字段可能返回 null，表示取不到有效值。
DomainNameConfig	DomainNameConfig	域名解析配置
NotificationHooks	Array of NotificationHook	通知事件回调 注意：此字段可能返回 null，表示取不到有效值。
Owner	String	创建人员 示例值：owner-name
ProjectName	String	场景所在的项目的名字 示例值：project-name

ScenarioRelatedJobsParams

查询与特定scenario关联的job的参数

被如下接口引用：DescribeScenarioWithJobs。

名称	类型	必选	描述
Offset	Integer	否	job偏移量 示例值：0
Limit	Integer	否	限制最多查询的job数 示例值：10
OrderBy	String	否	排序字段 示例值：start_time
Ascend	Boolean	否	是否升序 示例值：true/false

ScenarioWithJobs

带已执行任务的scenario

被如下接口引用：DescribeScenarioWithJobs。

名称	类型	描述
Scenario	Scenario	sceario结果
Jobs	Array of Job	job结果 注意：此字段可能返回 null，表示取不到有效值。

ScriptInfo

脚本信息

被如下接口引用：CreateScenario, DescribeFiles, DescribeJobs, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
Name	String	否	文件名 示例值：hello.js
Size	Integer	否	文件大小 示例值：100
Type	String	否	文件类型 示例值：js
UpdatedAt	Timestamp ISO8601	否	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-09-02T03:18:35+08:00
EncodedContent	String	否	base64编码后的文件内容 示例值：sfadsfdfsdfdsfdsgerd
EncodedHttpArchive	String	否	base64编码后的har结构体 示例值：sfadsfdfsdfdsfdsgerd
LoadWeight	Integer	否	脚本权重，范围 1-100 示例值：100
FileId	String	否	文件 ID 示例值：file-xxx
Uploaded	Boolean	否	文件是否已上传，如果已上传，则可以不填写 EncodedContent, EncodedHar 等内容。 主要用于较大长度脚本上传。 示例值：true/false

ScriptOrigin

脚本内置压力模型

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
MachineNumber	Integer	是	机器数量 示例值：10
MachineSpecification	String	是	机器规格 示例值：1c2g, 2c4g, 4g8g
DurationSeconds	Integer	是	压测时长 示例值：1200

Stage

分阶段施压时，对单个阶段的配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
DurationSeconds	Integer	否	施压时间 示例值：30
TargetVirtualUsers	Integer	否	虚拟用户数 示例值：30

TagSpec

标签

被如下接口引用：CreateProject, DescribeProjects, UpdateProject。

名称	类型	必选	描述
TagKey	String	否	标签键 示例值：owner
TagValue	String	否	标签值 示例值：tom

TestData

测试数据集

被如下接口引用：CreateScenario, DescribeFiles, DescribeJobs, DescribeScenarios, UpdateScenario。

名称	类型	必选	描述
Name	String	是	测试数据集所在的文件名 示例值：info.csv
Split	Boolean	是	测试数据集是否分片 示例值：true
HeaderInFile	Boolean	是	首行是否为参数名 示例值：true
HeaderColumns	Array of String	否	参数名数组 注意：此字段可能返回 null，表示取不到有效值。 示例值：["name", "age"]
LineCount	Integer	否	文件行数 示例值：10
UpdatedAt	Timestamp ISO8601	否	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2021-09-02T03:18:35+08:00
Size	Integer	否	文件字节数 示例值：10
HeadLines	Array of String	否	头部数据行 注意：此字段可能返回 null，表示取不到有效值。 示例值：["head-line-value"]
TailLines	Array of String	否	尾部数据行 注意：此字段可能返回 null，表示取不到有效值。 示例值：["tail-line-value"]
Type	String	否	文件类型 示例值：CSV
FileId	String	否	文件 ID 示例值：file-xx

VpcLoadDistribution

压力来源配置

被如下接口引用：CreateScenario, UpdateScenario。

名称	类型	必选	描述
RegionId	Integer	是	地域ID 示例值：1
Region	String	否	地域 示例值：ap-guangzhou

名称	类型	必选	描述
VpcId	String	否	VPC ID 示例值：vpc-xx
SubnetIds	Array of String	否	子网ID列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：["subnet-xxx"]

错误码

最近更新时间：2024-12-20 01:48:29

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
ActionOffline	接口已下线。
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在 控制台 检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP 地址在黑名单中。
IpNotInWhitelist	IP 地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。

错误码	说明
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP 限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
AuthFailure	CAM签名/鉴权错误。
FailedOperation.AccessTagFail	获取标签失败。
FailedOperation.DbQueryFailed	数据库查询失败。
FailedOperation.DbRecordCreateFailed	数据库创建记录失败。
FailedOperation.DbRecordUpdateFailed	数据库记录更新失败。
FailedOperation.JobStatusNotRunning	任务状态不是运行中。
FailedOperation.NoTasksInJob	任务中没有task。
FailedOperation.NotSupportedInEnv	当前环境不支持。
FailedOperation.ResourceNotFound	资源不存在。
FailedOperation.SendRequest	请求发送失败。