

事件总线 API 文档



腾讯云

【 版权声明 】

© 2013–2024 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

事件目标相关接口

创建事件目标

删除事件目标

获取事件目标列表

更新事件目标

事件集相关接口

删除事件集

Event 事件投递

事件投递

创建事件集

获取事件集详情

获取事件集列表

更新事件集

事件规则相关接口

检验规则

创建事件规则

删除事件规则

获取事件规则列表

更新事件规则

获取事件规则详情

事件连接器相关接口

创建事件连接器

删除事件连接器

获取事件连接器列表

更新事件连接器

数据转换器相关接口

检查转换器

创建转换器

删除转换器

获取转换器详情

更新转换器

事件存储相关接口

日志检索

查询日志索引维度值

事件模版相关接口

获取平台产品事件匹配规则

获取平台产品事件名称

获取平台产品列表

获取平台产品事件模板

数据结构

错误码

API 文档

更新历史

最近更新时间：2023-11-24 01:14:16

第 13 次发布

发布时间：2023-11-24 01:14:11

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [TDMQParams](#)

修改数据结构：

- [ConnectionDescription](#)
 - 新增成员：TDMQParams
- [DTSPParams](#)
 - 新增成员：ConsumerGroupName, Account, Password

第 12 次发布

发布时间：2023-09-08 01:48:45

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetPlatformEventTemplate](#)
- [ListPlatformEventNames](#)
- [ListPlatformEventPatterns](#)
- [ListPlatformProducts](#)

新增数据结构：

- [PlatformEventDetail](#)
- [PlatformEventSummary](#)
- [PlatformProduct](#)

第 11 次发布

发布时间：2023-08-22 01:12:28

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CheckRule](#)
 - 新增入参：Event, EventPattern

第 10 次发布

发布时间：2023-03-24 01:40:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeLogTagValue](#)
- [SearchLog](#)

新增数据结构：

- [LogFilter](#)
- [LogFilters](#)
- [SearchLogResult](#)

第 9 次发布

发布时间：2023-02-28 01:29:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateEventBus](#)
 - 新增入参：EnableStore
- [GetEventBus](#)
 - 新增出参：PayMode, SaveDays, LogTopicId, EnableStore, LinkMode

新增数据结构：

- [ConnectionBrief](#)
- [DTSPParams](#)

修改数据结构：

- [ConnectionDescription](#)
 - 新增成员：DTSPParams
- [EventBus](#)
 - 新增成员：PayMode, ConnectionBriefs, TargetBriefs

第 8 次发布

发布时间：2023-02-27 01:23:14

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateEventBus](#)
 - 新增入参：EnableStore

第 7 次发布

发布时间：2023-02-16 01:24:24

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateEventBus](#)
 - 新增入参：SaveDays
- [UpdateEventBus](#)
 - 新增入参：SaveDays, LogTopicId

第 6 次发布

发布时间：2022-08-23 06:27:55

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateRule](#)
 - 新增入参：EventPattern

第 5 次发布

发布时间：2022-06-30 06:06:12

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [ESTargetParams](#)

修改数据结构：

- [TargetDescription](#)
 - 新增成员：ESTargetParams

第 4 次发布

发布时间：2022-06-28 06:09:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ListTargets](#)
 - 修改入参：RuleId

第 3 次发布

发布时间：2022-05-26 06:11:01

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Event](#)
 - 新增成员：Time

第 2 次发布

发布时间：2022-01-20 08:09:25

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [PutEvents](#)

第 1 次发布

发布时间：2022-01-06 16:11:19

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CheckRule](#)
- [CheckTransformation](#)
- [CreateConnection](#)
- [CreateEventBus](#)
- [CreateRule](#)
- [CreateTarget](#)
- [CreateTransformation](#)
- [DeleteConnection](#)
- [DeleteEventBus](#)
- [DeleteRule](#)
- [DeleteTarget](#)
- [DeleteTransformation](#)
- [GetEventBus](#)
- [GetRule](#)
- [GetTransformation](#)
- [ListConnections](#)
- [ListEventBuses](#)
- [ListRules](#)
- [ListTargets](#)
- [PublishEvent](#)
- [UpdateConnection](#)
- [UpdateEventBus](#)
- [UpdateRule](#)
- [UpdateTarget](#)
- [UpdateTransformation](#)

新增数据结构：

- [APIGWParams](#)
- [CkafkaDeliveryParams](#)
- [CkafkaParams](#)
- [CkafkaTargetParams](#)
- [Connection](#)
- [ConnectionDescription](#)
- [DeadLetterConfig](#)
- [EtlFilter](#)
- [Event](#)
- [EventBus](#)
- [Extraction](#)
- [Filter](#)
- [OutputStructParam](#)
- [RetryPolicy](#)
- [Rule](#)
- [SCFParams](#)
- [Target](#)
- [TargetBrief](#)
- [TargetDescription](#)
- [TextParams](#)
- [Transform](#)
- [Transformation](#)

简介

最近更新时间：2022-01-06 16:11:37

欢迎使用 事件总线 API [3.0 版本](#)。全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验。支持全地域就近接入让您更快连接腾讯云产品。更多腾讯云 API 3.0 使用介绍请查看：[快速入门](#)

EventBridge 事件中心是一项独立的 Serverless 服务，让您无需编写代码即可实时了解腾讯云服务、自己的应用程序以及软件即服务 (SaaS) 应用程序中的数据的变化。

API 概览

最近更新时间：2023-10-17 01:16:22

事件目标相关接口

接口名称	接口功能	频率限制（次/秒）
CreateTarget	创建事件目标	20
DeleteTarget	删除事件目标	20
ListTargets	获取事件目标列表	20
UpdateTarget	更新事件目标	20

事件集相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteEventBus	删除事件集	20
PublishEvent	Event事件投递	200
PutEvents	事件投递	1000
CreateEventBus	创建事件集	20
GetEventBus	获取事件集详情	20
ListEventBuses	获取事件集列表	20
UpdateEventBus	更新事件集	20

事件规则相关接口

接口名称	接口功能	频率限制（次/秒）
CheckRule	检验规则	20
CreateRule	创建事件规则	20
DeleteRule	删除事件规则	20
ListRules	获取事件规则列表	20
UpdateRule	更新事件规则	20
GetRule	获取事件规则详情	20

事件连接器相关接口

接口名称	接口功能	频率限制（次/秒）
CreateConnection	创建事件连接器	20
DeleteConnection	删除事件连接器	20
ListConnections	获取事件连接器列表	20
UpdateConnection	更新事件连接器	20

数据转换器相关接口

接口名称	接口功能	频率限制（次/秒）
------	------	-----------

接口名称	接口功能	频率限制（次/秒）
CheckTransformation	检查转换器	20
CreateTransformation	创建转换器	20
DeleteTransformation	删除转换器	20
GetTransformation	获取转换器详情	20
UpdateTransformation	更新转换器	20

事件存储相关接口

接口名称	接口功能	频率限制（次/秒）
SearchLog	日志检索	20
DescribeLogTagValue	查询日志索引维度值	20

事件模版相关接口

接口名称	接口功能	频率限制（次/秒）
GetPlatformEventTemplate	获取平台产品事件模板	20
ListPlatformEventNames	获取平台产品事件名称	20
ListPlatformEventPatterns	获取平台产品事件匹配规则	20
ListPlatformProducts	获取平台产品列表	20

调用方式

请求结构

最近更新时间：2023-02-22 01:24:38

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 `eb.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `eb.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 `eb.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）	eb.tencentcloudapi.com
华南地区(广州)	eb.ap-guangzhou.tencentcloudapi.com
华东地区(上海)	eb.ap-shanghai.tencentcloudapi.com
华北地区(北京)	eb.ap-beijing.tencentcloudapi.com
西南地区(成都)	eb.ap-chengdu.tencentcloudapi.com
西南地区(重庆)	eb.ap-chongqing.tencentcloudapi.com
港澳台地区(中国香港)	eb.ap-hongkong.tencentcloudapi.com
亚太东南(新加坡)	eb.ap-singapore.tencentcloudapi.com
亚太东南(曼谷)	eb.ap-bangkok.tencentcloudapi.com
亚太南部(孟买)	eb.ap-mumbai.tencentcloudapi.com
亚太东北(首尔)	eb.ap-seoul.tencentcloudapi.com
亚太东北(东京)	eb.ap-tokyo.tencentcloudapi.com
美国东部(弗吉尼亚)	eb.na-ashburn.tencentcloudapi.com
美国西部(硅谷)	eb.na-siliconvalley.tencentcloudapi.com
北美地区(多伦多)	eb.na-toronto.tencentcloudapi.com
欧洲地区(法兰克福)	eb.eu-frankfurt.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：2023-12-28 01:14:35

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/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，AKIDEXAMPLE 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 eb；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*****EXAMPLE/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*****EXAMPLE/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*****EXAMPLE/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*****EXAMPLE

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*****EXAMPLE
```

地域列表

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

地域	取值
华北地区（北京）	ap-beijing
西南地区（成都）	ap-chengdu
西南地区（重庆）	ap-chongqing
华南地区（广州）	ap-guangzhou
港澳台地区（中国香港）	ap-hongkong
华东地区（上海）	ap-shanghai
亚太东南（新加坡）	ap-singapore
美国西部（硅谷）	na-siliconvalley

签名方法 v3

最近更新时间：2023-12-27 01:18:08

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：

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

签名版本 v3 签名过程

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

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

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

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

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsj5yKBZQpn74WFkmLPx3***** 和 Gu5t9xGARNpq86cd98joQYCN3*****。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=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3" \
-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 = "Gu5t9xGARNpq86cd98joQYCN3*****"
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

据，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：
f1cb4d518a0eda9d5cbbfdb7850983f1e603eeae484edea76e4dd8d8deb5556e，
e7c609ce81bea53546bed2cc904778bef9ca14082e48e67883443ed64e227cd7，
8aa8ab5755582f576e94bcfe383b8e29325b0ca90c3590d569221c6a63a091ed。

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 Gu5t9xGARNpq86cd98joQYCN3*****。
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 be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3。

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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****。
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 be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-t
```

```
ype:host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3
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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    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 Gu5t9xGARNpq86cd98joQYCN3*****
    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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
$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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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}\nx-tc-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, 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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
```

```
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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, 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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

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

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

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

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

// ***** 步骤 4：拼接 Authorization *****
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;

string curlcmd = "curl -X POST https://" + host + "\n"
```

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

C

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

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

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

void hmac_sha256(char* key, const char* input, char* result)
{
    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, strlen(key), EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )input, strlen(input));
unsigned int len = 32;
HMAC_Final(h, hash, &len);

#ifdef OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif
strncpy(result, (const char*)hash, len);
}

void hex_encode(const char* input, char* output)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = strlen(input);
    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < 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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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%s", "TC3", SECRET_KEY);
char k_date[64] = {0};
hmac_sha256(k_key, date, k_date);
char k_service[64] = {0};
hmac_sha256(k_date, service, k_service);
char k_signing[64] = {0};
hmac_sha256(k_service, "tc3_request", k_signing);
char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, string_to_sign, k_hmac_sha_sign);

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, 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：2024-01-05 01:15:06

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

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

推荐使用 API Explorer

</> 点击调试

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

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

1. 申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3*****

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	AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
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': 'AKIDz8krbsj5yKBZQpn74WFkmLPx3*****',
  '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=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&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-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&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 = 'Gu5t9xGARNpq86cd98joYCN3*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&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：

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

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 zmmjn35mikh6pM3V7sUEuX4wyYM=，最终得到的签名串请求参数（Signature）为：
zmmjn35mikh6pM3V7sUEuX4wyYM%3D，它将用于生成最终的请求 URL。

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

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

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

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

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

5. 签名演示

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

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

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

- [Signature Demo](#)

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

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

guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
        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 Gu5t9xGARNpq86cd98joQYCN3*****
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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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",
        "InstanceId.0": "ins-09dx96dg",
        "Limit": strconv.Itoa(20),
        "Offset": strconv.Itoa(0),
    }

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

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

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

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

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

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
$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["InstanceId.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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
```

```
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 AKIDz8krbsj5yKBZQpn74WfkmLPx3*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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("InstanceId.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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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['InstanceId.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：2024-01-09 01:15:02

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，可以联系我们，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，可以联系我们，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 06:29:34

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。

事件目标相关接口

创建事件目标

最近更新时间：2023-12-15 19:35:54

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

创建事件目标

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：CreateTarget。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxx
Type	是	String	目标类型;取值范围:scf(云函数)/cls(日志服务)/amp(消息推送)/ckafka(消息推送)/es(大数据elastic-se 示例值：scf
TargetDescription	是	TargetDescription	目标描述;scf类型示例: { "ResourceDescription": "qcs::scf:ap-guangzhou:uin/2252646423:namespace/default/fun0326/\$LATEST" }; cls类型示例: { "ResourceDescription": "qcs::cls:ap-guangzhou:uin/12323442323:topic/7103f705-6c38-428af0f2e732" } ckafka类型示例: { "ResourceDescription": "qcs::ckafka:ap-guangzhou:uin/1500000688:ckafkaId/uin/15000018q1nwj", "CkafkaTargetParams": { "TopicName": "alert", "RetryPolicy": { "RetryInterval": 60, "MaxRetryAttempts": 360 } } } amp类型-邮件/短信示例: { "ResourceDescription": "qcs::eb-amp:ap-guangzhou:uin/100012505002:", "AMPParams": { "NotificationTemplateId": 10181, "Lang": "cn", "NoticeReceivers": [{ "UserType": "User", "User": "9424525", "TimeWindow": { "From": "09:30:00", "To": "23:30:00" }, "Channels": ["Email", "SMS"] }], "Channels": ["Email", "SMS"] } } es类型示例: { "ResourceDescription": "qcs::es:ap-guangzhou:uin/1500000688:instance/es-7cplmhsd", "EsVersion": "7.14.2", "UserName": "elastic", "Password": "xxxxx", "NetMode": "privateLink", "test": "IndexSuffixMode": "default", "RotationInterval": "none", "IndexTemplateType": "", "Outp 示例值： { "ResourceDescription": "qcs::scf:ap-guangzhou:uin/2252646423:namespace/default-0326/\$LATEST" }
RuleId	是	String	事件规则ID 示例值：rule-xxxxxxx

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
TargetId	String	目标ID 示例值: target-e9r3a1l0
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建目标

输入示例

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

{
  "EventBusId": "eb-xxxxxx",
  "Type": "scf",
  "TargetDescription": {
    "ResourceDescription": "qcs::scf:ap-guangzhou:uin/xxxxxxx:namespace/xxxxxx/function/xxxxx/x"
  },
  "RuleId": "rule-xxxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b7662cf2-ce20-4b3e-aff2-2cb875cf0b6b",
    "TargetId": "target-xxxxxx"
  }
}
```

示例2 创建自定义投递目标

输入示例

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

{
  "EventBusId": "eb-xxxxxx",
  "Type": "scf",
  "TargetDescription": {
    "ResourceDescription": "qcs::custom_scf:ap-guangzhou:appid/xxxxx/uin/xxxxxxx:namespace/xxxxxx/function/xxxxx/x"
  },
  "RuleId": "rule-xxxxxxx"
}
```

输出示例

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

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
FailedOperation.AddPrivateLink	添加私有链接失败。
FailedOperation.AuthenticateUserFailed	用户鉴权错误。
FailedOperation.CreateTrigger	投递目标创建失败，请检查函数状态。
FailedOperation.ESInternalError	ES集群内部错误。
FailedOperation.ESRequestFailed	ES集群操作失败。
FailedOperation.ESTemplateConflict	ES索引模板冲突错误。
InvalidParameter.Payload	Payload与规范不符，请修正后再试。
InvalidParameterValue.AMPParams	AMPParams取值与规范不符，请修正后再试。
InvalidParameterValue.BatchEventCount	BatchEventCount取值与规范不符，请修正后再试。
InvalidParameterValue.BatchTimeout	BatchTimeout取值与规范不符，请修正后再试。
InvalidParameterValue.CKafkaTargetParams	Ckafka 目标配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.CallbackType	CallbackType取值与规范不符，请修正后再试。

错误码	描述
InvalidParameterValue.CallbackWeComURL	CallbackWeComURL取值与规范不符，请修正后再试。
InvalidParameterValue.ElasticSearchTargetParams	ES目标参数错误。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverChannel	NoticeReceiverChannel取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverTimeWindow	NoticeReceiverTimeWindow取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverUserIds	NoticeReceiverUserIds取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverUserType	NoticeReceiverUserType取值与规范不符，请修正后再试。
InvalidParameterValue.Qualifier	Qualifier取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.TargetDescription	TargetDescription取值与规范不符，请修正后再试。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。
LimitExceeded.ClusterPrivateLinkExceeded	集群私有链接超限。
LimitExceeded.Logset	Logset数量达到限制，可提交工单申请提升限制。
LimitExceeded.Target	Target数量达到限制，可提交工单申请提升限制。
LimitExceeded.Trigger	投递目标创建失败，函数触发器数量达到限制，可提交工单申请提升限制。
LimitExceeded.UserPrivateLinkExceeded	用户私有链接超限。
OperationDenied.AccountNotExists	投递目标创建失败，检测到当前账号不存在，请确认您的账号状态。
OperationDenied.DefaultCLSResourceUnsupported	不允许在自定义事件集上创建基于EB默认日志集的CLS事件目标。
OperationDenied.ESVersionUnsupported	不支持操作当前ES版本。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
OperationDenied.UnsupportedOperation	当前用户账号类型暂不支持操作，请提交工单处理。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Function	未找到指定的Function，请创建后再试。
ResourceNotFound.Namespace	投递目标创建失败，未找到指定的命名空间，请创建后再试。
ResourceNotFound.PrivateLinkResource	未找到privatelink记录。
ResourceNotFound.Role	未找到指定的服务角色，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。
ResourceNotFound.Target	未找到指定的Target，请创建后再试。
ResourceNotFound.Version	投递目标创建失败，未找到指定的服务版本，请创建后再试。
ResourceUnavailable.ESUnhealth	ES集群状态异常。
ResourceUnavailable.Target	投递目标创建失败，当前资源不可用。

删除事件目标

最近更新时间：2023-11-30 03:54:55

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

删除事件目标

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：DeleteTarget。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxx
TargetId	是	String	事件目标ID 示例值：target-xxxxxxxx
RuleId	是	String	事件规则ID 示例值：rule-xxxxxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除目标

输入示例

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

{
  "EventBusId": "eb-753e9ung",
  "TargetId": "target-gxk2ukxy",
  "RuleId": "rule-n0v4zane"
}
```


输出示例

```
{
  "Response": {
    "RequestId": "587a0f0e-8c5d-40c2-874b-eabf103d8004"
  }
}
```

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.AuthenticateUserFailed	用户鉴权错误。
FailedOperation.DeletePrivateLink	删除私有链接失败。
FailedOperation.ESInternalError	ES集群内部错误。
FailedOperation.ESRequestFailed	ES集群操作失败。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.TargetId	TargetId取值与规范不符，请修正后再试。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。
OperationDenied.ESVersionUnsupported	不支持操作当前ES版本。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.NetAssociation	未找到网络资源关联记录。

错误码	描述
ResourceNotFound.PrivateLinkResource	未找到privatelink记录。
ResourceNotFound.Role	未找到指定的服务角色，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。
ResourceNotFound.Target	未找到指定的Target，请创建后再试。

获取事件目标列表

最近更新时间：2023-11-30 03:54:54

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取事件目标列表

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：ListTargets。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxx
OrderBy	否	String	根据哪个字段进行返回结果排序,支持以下字段：AddTime（创建时间），ModTime（修改时间） 示例值：ModTime
RuleId	否	String	事件规则ID 示例值：rule-xxxxxxx
Limit	否	Integer	返回数量，默认为20，最大值为100。 示例值：0
Offset	否	Integer	分页偏移量，默认为0。 示例值：20
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC（升序）和 DESC（降序） 示例值：DESC

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	目标总数 示例值：1
Targets	Array of Target	目标信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取目标列表

输入示例

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

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

```
{
  "EventBusId": "eb-l65v1c2",
  "RuleId": "rule-fdltium8"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "03118ade-8de2-4514-af4b-b9a2f170ddd9",
    "Targets": [
      {
        "EventBusId": "eb-l65v1c2u",
        "RuleId": "rule-fdltium8",
        "TargetDescription": {
          "ResourceDescription": "qcs::scf:ap-guanzhou:uin/3473058547:namespace/default/function/test/1"
        },
        "TargetId": "target-azwj7s7g",
        "Type": "scf",
        "EnableBatchDelivery": false,
        "BatchEventCount": 1,
        "BatchTimeout": 1
      },
      {
        "EventBusId": "eb-l65v1c2u",
        "RuleId": "rule-fdltium8",
        "TargetDescription": {
          "ResourceDescription": "qcs::scf:ap-guanzhou:uin/3473058547:namespace/default/function/test/1"
        },
        "TargetId": "target-o5yx01oq",
        "Type": "scf",
        "EnableBatchDelivery": false,
        "BatchEventCount": 1,
        "BatchTimeout": 1
      },
      {
        "EventBusId": "eb-l65v1c2u",
        "RuleId": "rule-fdltium8",
        "TargetDescription": {
          "ResourceDescription": "qcs::scf:ap-guanzhou:uin/3473058547:namespace/default/function/test/1"
        },
        "TargetId": "target-prp1ovqi",
        "Type": "scf",
        "EnableBatchDelivery": false,
        "BatchEventCount": 1,
        "BatchTimeout": 1
      },
      {
        "EventBusId": "eb-l65v1c2u",
        "RuleId": "rule-fdltium8",
        "TargetDescription": {
          "ResourceDescription": "qcs::scf:ap-guanzhou:uin/3473058547:namespace/default/function/test/1"
        }
      }
    ]
  }
}
```

```
},
"TargetId": "target-krcwchke",
"Type": "scf",
"EnableBatchDelivery": false,
"BatchEventCount": 1,
"BatchTimeout": 1
},
{
"EventBusId": "eb-l65v1c2u",
"RuleId": "rule-fdltium8",
"TargetDescription": {
"ResourceDescription": "qcs::scf:ap-guanzhou:uin/3473058547:namespace/default/function/test/1"
},
"TargetId": "target-e9r3a1l0",
"Type": "scf",
"EnableBatchDelivery": false,
"BatchEventCount": 1,
"BatchTimeout": 1
}
],
"TotalCount": 5
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.Limit	Limit取值与规范不符，请修正后再试。
InvalidParameterValue.Offset	Offset取值与规范不符，请修正后再试。

错误码	描述
InvalidParameterValue.Order	Order取值与规范不符，请修正后再试。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.TargetId	TargetId取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。

更新事件目标

最近更新时间：2023-12-05 01:43:40

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

更新事件目标

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：UpdateTarget。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxx
RuleId	是	String	事件规则ID 示例值：rule-xxxxxxx
TargetId	是	String	事件目标ID 示例值：target-xxxxxxx
EnableBatchDelivery	否	Boolean	开启批量投递使能 示例值：false
BatchTimeout	否	Integer	批量投递最长等待时间 示例值：1
BatchEventCount	否	Integer	批量投递最大事件条数 示例值：1

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新目标

输入示例

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

```
{
  "EnableBatchDelivery": "true",
  "RuleId": "rule-51wvf3is",
  "BatchEventCount": "32",
  "TargetId": "target-ceh92ss4",
  "EventBusId": "eb-ohb9x6rg",
  "BatchTimeout": "43"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "72b987c9-8842-47ac-9ad6-fa2bde3e936d"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InternalError	内部错误。
InvalidParameterValue.BatchTimeout	BatchTimeout取值与规范不符，请修正后再试。
InvalidParameterValue.CallbackType	CallbackType取值与规范不符，请修正后再试。
InvalidParameterValue.CallbackWeComURL	CallbackWeComURL取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。

错误码	描述
InvalidParameterValue.NoticeReceiverChannel	NoticeReceiverChannel取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverTimeWindow	NoticeReceiverTimeWindow取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverUserIds	NoticeReceiverUserIds取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverUserType	NoticeReceiverUserType取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.TargetId	TargetId取值与规范不符，请修正后再试。
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。
ResourceNotFound.Target	未找到指定的Target，请创建后再试。

事件集相关接口

删除事件集

最近更新时间：2023-11-30 03:54:47

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

删除事件集

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：DeleteEventBus。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除事件集

输入示例

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

{
  "EventBusId": "eb-l65vlc2u"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b3103b8d-77a6-4b55-b42f-a25e2dc453a9"
  }
}
```

```
}  
}
```

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.UnTagResource	标签解绑操作失败。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
ResourceInUse.EventBus	无法删除，因事件集下存在规则、连接器或归档，请删除所有规则、连接器和归档后重试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Tag	未找到指定标签。

Event事件投递

最近更新时间：2023-11-30 03:54:46

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

（已废弃）用于Event事件投递

默认接口请求频率限制：200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：PublishEvent。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventList.N	是	Array of Event	事件列表 示例值：EventList
EventBusId	是	String	事件集ID 示例值：EventBusId

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 事件投递

输入示例

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

{
  "EventList": [
    {
      "Data": "string",
      "Type": "string",
      "Source": "string",
      "Subject": "qcs::dts:ap-guangzhou:appid12312/uid1250000000:xxx"
    },
    {
```

```
"Data": "string",
"Type": "string",
"Source": "string",
"Subject": "qcs::dts:ap-guangzhou:appid12312/uid1250000000:xxx"
},
{
  "EventBusId": "string"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e3d43926-c2cd-49f2-97f0-53db21e6fcea"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.System	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。
InvalidParameter	参数错误。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.InvalidEvent	事件格式非法，请修正后再试。
InvalidParameterValue.InvalidEventBus	非法的事件集，请检查后重试。
LimitExceeded.ResourceLimit	ResourceLimit数量达到限制，可提交工单申请提升限制。

错误码	描述
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.EventBusNotFound	事件集不存在或未配置规则，请检查后再试。

事件投递

最近更新时间：2023-11-30 03:54:45

.....

1. 接口描述

接口请求域名：`eb.tencentcloudapi.com`。

用于Event事件投递

默认接口请求频率限制：1000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：PutEvents。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventList.N	是	Array of Event	事件列表 示例值：EventList
EventBusId	是	String	事件集ID 示例值：EventBusId

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 事件投递

输入示例

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

{
  "EventList": [
    {
      "Data": "string",
      "Type": "string",
      "Source": "string",
      "Subject": "qcs::dts:ap-guangzhou:appid12312/uid1250000000:xxx"
    },
    {
      "Data": "string",
      "Type": "string",
      "Source": "string",
      "Subject": "qcs::dts:ap-guangzhou:appid12312/uid1250000000:xxx"
    }
  ]
}
```

```
}
],
"EventBusId": "string"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ae684d86-33ec-449e-bc38-9d9552199267"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.System	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。
InternalServerError.UnknownInternalServerError	未知错误。
InvalidParameter	参数错误。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.InvalidEvent	事件格式非法，请修正后再试。
InvalidParameterValue.InvalidEventBus	非法的事件集，请检查后重试。
LimitExceeded.BannedAccount	事件投递失败，因欠费或违规等原因，账号已被禁用，请联系官网账户端客服处理。
LimitExceeded.ResourceLimit	ResourceLimit数量达到限制，可提交工单申请提升限制。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

错误码	描述
ResourceNotFound.EventBusNotFound	事件集不存在或未配置规则，请检查后再试。

创建事件集

最近更新时间：2023-11-30 03:54:48

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

用于创建事件集

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：CreateEventBus。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusName	是	String	事件集名称，只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2~60个字符 示例值：EventBusName
Description	否	String	事件集描述，不限字符类型，200字符描述以内 示例值：这是一个描述
SaveDays	否	Integer	EB存储时长
EnableStore	否	Boolean	EB是否开启存储

3. 输出参数

参数名称	类型	描述
EventBusId	String	事件集ID 示例值：eb-xxxxxxxxx
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建事件集

创建事件集

输入示例

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

{
  "EventBusName": "test"
}
```

输出示例

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

```
"EventBusId": "eb-xxxxxxx",
"RequestId": "e3d43926-c2cd-49f2-97f0-53db21e6fcea"
}
}
```

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.RegisterCLSService	注册CLS服务失败。
FailedOperation.TagResource	标签绑定操作失败。
FailedOperation.TagResourceAllocateQuotas	申请标签配额失败。
InternalServerError	内部错误。
InvalidParameterValue.Description	Description取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusName	EventBusName取值与规范不符，请修正后再试。
InvalidParameterValue.EventTraceConfig	链路追踪配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.LinkMode	LinkMode取值与规范不符，请修正后再试。
InvalidParameterValue.Tags	标签参数错误。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。
LimitExceeded.EventBus	EventBus数量达到限制，可提交工单申请提升限制。
LimitExceeded.InsufficientBalance	资源创建失败，可冻结余额不足，请充值后重新创建。
OperationDenied.DefaultCLSResourceUnsupported	不允许在自定义事件集上创建基于EB默认日志集的CLS事件目标。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
ResourceInUse.DefaultEventBus	同一个地域下，云服务默认事件集不允许重复创建。

错误码	描述
ResourceInUse.EventBus	无法删除，因事件集下存在规则、连接器或归档，请删除所有规则、连接器和归档后重试。
ResourceNotFound	资源不存在。
ResourceNotFound.Role	未找到指定的服务角色，请创建后再试。
ResourceNotFound.Tag	未找到指定标签。
ResourceUnavailable.Target	投递目标创建失败，当前资源不可用。

获取事件集详情

最近更新时间：2023-11-30 03:54:47

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取事件集详情

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：GetEventBus。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxx

3. 输出参数

参数名称	类型	描述
ModTime	Timestamp ISO8601	更新时间 示例值：2021-04-27T14:34:19+08:00
Description	String	事件集描述 示例值：这是一个描述
ClsTopicId	String	日志主题ID 示例值：78a5887e-357d-4bac-baca-1de008fa6ff0
AddTime	Timestamp ISO8601	创建时间 示例值：2021-04-27T14:34:19+08:00
ClsLogsetId	String	日志集ID 示例值：78a5887e-357d-4bac-baca-1de008fa6ff0
EventBusName	String	事件集名称 示例值：eb
EventBusId	String	事件集ID 示例值：eb-xxxxxxxx
Type	String	（已废弃）事件集类型 示例值：Cloud
PayMode	String	计费模式
SaveDays	Integer	EB日志存储时长 注意：此字段可能返回 null，表示取不到有效值。
LogTopicId	String	EB日志主题ID 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
EnableStore	Boolean	是否开启存储 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
LinkMode	String	消息序列，是否有序 注意：此字段可能返回 null，表示取不到有效值。 示例值：Disorder
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取事件集

获取事件集

输入示例

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

{
  "EventBusId": "eb-l65vlc2"
}
```

输出示例

```
{
  "Response": {
    "AddTime": "2022-12-12T20:09:46+08:00",
    "ClsLogsetId": "",
    "ClsTopicId": "",
    "Description": "",
    "EventBusId": "eb-0gdrbv5q",
    "EventBusName": "default",
    "ModTime": "2022-12-12T20:09:46+08:00",
    "PayMode": "",
    "LogTopicId": "xxx",
    "SaveDays": "1",
    "EnableStore": true,
    "LinkMode": "Disorder",
    "RequestId": "ffd4aae2-c29e-40a8-b18c-037a17ed810c",
    "Type": "Cloud"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

获取事件集列表

最近更新时间：2023-12-15 19:35:52

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取事件集列表

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：ListEventBuses。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
OrderBy	否	String	根据哪个字段进行返回结果排序,支持以下字段：AddTime（创建时间），ModTime（修改时间） 示例值：ModTime
Limit	否	Integer	返回数量，默认为20，最大值为100。 示例值：20
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC（升序）和 DESC（降序） 示例值：DESC
Filters.N	否	Array of Filter	过滤字段范围：EventBusName(事件集名称)/EventBusId(事件集Id)/Type(事件集类型:Cloud(云服务);Platform(平台型);Custom(自定义))/TagKey(标签键)。每次请求的Filters的上限为10，Filter.Values的上限为5。 [{"Name":"Type","Values":["Cloud","Platform"]}] 示例值：[{"Name":"Type","Values":["Cloud","Platform"]}]
Offset	否	Integer	分页偏移量，默认为0。 示例值：0

3. 输出参数

参数名称	类型	描述
EventBuses	Array of EventBus	事件集信息
TotalCount	Integer	事件集总数 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取事件集列表

获取事件集列表

输入示例


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

{}
```

输出示例

```
{
  "Response": {
    "EventBuses": [
      {
        "ModTime": "2020-09-22T00:00:00+00:00",
        "Description": "abc",
        "AddTime": "2020-09-22T00:00:00+00:00",
        "EventBusName": "abc",
        "EventBusId": "abc",
        "Type": "abc",
        "PayMode": "abc",
        "ConnectionBriefs": [
          {
            "Type": "abc",
            "Status": "abc"
          }
        ],
        "TargetBriefs": [
          {
            "TargetId": "abc",
            "Type": "abc"
          }
        ],
        "TotalCount": 0,
        "RequestId": "abc"
      }
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameterValue.Filters	Filters取值与规范不符，请修正后再试。
InvalidParameterValue.Limit	Limit取值与规范不符，请修正后再试。
InvalidParameterValue.Offset	Offset取值与规范不符，请修正后再试。
InvalidParameterValue.Order	Order取值与规范不符，请修正后再试。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。
UnauthorizedOperation.CAM	当前账号缺少 EB 操作权限，请登录 CAM 控制台进行授权。

更新事件集

最近更新时间：2023-11-30 03:54:45

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

更新事件集

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：UpdateEventBus。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxx
Description	否	String	事件集描述，不限字符类型，200字符描述以内 示例值：这是一个描述
EventBusName	否	String	事件集名称，只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2~60个字符 示例值：EventBusName
SaveDays	否	Integer	EB日志存储时长
LogTopicId	否	String	EB日志主题ID
EnableStore	否	Boolean	是否开启存储 示例值：true

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新事件集

更新事件集

输入示例

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

```
{
  "EventBusName": "test111",
  "EventBusId": "eb-l65vlc2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "18484e4f-f6f1-4b87-8827-27422b30c8c4"
  }
}
```

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.RegisterCLSService	注册CLS服务失败。
InvalidParameter.Payload	Payload与规范不符，请修正后再试。
InvalidParameterValue.Description	Description取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusName	EventBusName取值与规范不符，请修正后再试。
InvalidParameterValue.EventTraceConfig	链路追踪配置参数取值与规范不符，请修正后再试。
OperationDenied.DefaultCLSResourceUnsupported	不允许在自定义事件集上创建基于EB默认日志集的CLS事件目标。
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

事件规则相关接口

检验规则

最近更新时间：2023-11-30 03:54:53

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

检验规则

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CheckRule。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Event	是	String	Event信息 示例值：event消息json字符串
EventPattern	是	String	EventPattern信息 示例值：EventPattern json字符串

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 规则校验

输入示例

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

{
  "Event": "xdd",
  "EventPattern": "xddx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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.ErrorFilter	规则与事件不匹配，请修正后再试。
InternalServerError	内部错误。
InternalServerError.System	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。
InternalServerError.UnknownInternalServerError	未知错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidFilterRule	事件模式格式错误，请修正后再试。
InvalidParameterValue.InvalidPattern	非法的匹配规则，请检查后重试。

创建事件规则

最近更新时间：2023-11-30 03:54:53

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

创建事件规则

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateRule。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventPattern	是	String	参考： 事件模式 示例值：{ "detail": { "source-ip": [{ "cidr": "10.0.0.0/24" }] }
EventBusId	是	String	事件集ID。 示例值：eb-xxxxxxx
RuleName	是	String	事件集名称，只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2~60个字符 示例值：rule
Enable	否	Boolean	使能开关。 示例值：true
Description	否	String	事件集描述，不限字符类型，200字符描述以内 示例值：这是一个描述

3. 输出参数

参数名称	类型	描述
RuleId	String	事件规则ID 示例值：rule-xxxxxxx
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建过滤规则

输入示例

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

```
{
  "EventBusId": "eb-anao4i7q",
  "Description": "rule-test1",
  "Enable": true,
  "EventPattern": "{\\n \\\"source\\\": [\\n \\\"ckafka.cloud.tencent\\\"\\n ]\\n}",
  "RuleName": "rule-test1"
}
```

输出示例

```
{
  "Response": {
    "RuleId": "rule-xxxxxxx",
    "RequestId": "xxxxxxxxxxxxxxxxxxxxx"
  }
}
```

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.TagResource	标签绑定操作失败。
FailedOperation.TagResourceAllocateQuotas	申请标签配额失败。
InvalidParameterValue.DeadLetterConfig	死信队列配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.Description	Description取值与规范不符，请修正后再试。
InvalidParameterValue.EventPattern	EventPattern取值与规范不符，请修正后再试。
InvalidParameterValue.RuleName	RuleName取值与规范不符，请修正后再试。

错误码	描述
InvalidParameterValue.Tags	标签参数错误。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。
LimitExceeded.Rule	Rule数量达到限制，可提交工单申请提升限制。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Tag	未找到指定标签。

删除事件规则

最近更新时间：2023-11-30 03:54:52

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

删除事件规则

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteRule。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxx
RuleId	是	String	事件规则ID 示例值：rule-xxxxxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除事件规则

输入示例

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

{
  "RuleId": "rule-xxxxxxxx",
  "EventBusId": "eb-l65vlc2u"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e3d43926-c2cd-49f2-97f0-53db21e6fcea"
  }
}
```

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.DeleteRule	规则删除失败，请检查资源信息，确认是资源是否存在或状态正常。
FailedOperation.UnTagResource	标签解绑操作失败。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
ResourceInUse.Rule	规则下有目标，无法删除，请删除所有目标后重试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。
ResourceNotFound.Tag	未找到指定标签。

获取事件规则列表

最近更新时间：2023-11-30 03:54:51

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取事件规则列表

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：ListRules。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxx
OrderBy	否	String	根据哪个字段进行返回结果排序,支持以下字段：AddTime（创建时间），ModTime（修改时间） 示例值：ModTime
Limit	否	Integer	返回数量，默认为20，最大值为100。 示例值：20
Offset	否	Integer	分页偏移量，默认为0。 示例值：0
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC（升序）和 DESC（降序） 示例值：DESC

3. 输出参数

参数名称	类型	描述
Rules	Array of Rule	事件规则信息
TotalCount	Integer	事件规则总数 示例值：8
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取事件规则列表

输入示例

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

<公共请求参数>

```
{
  "EventBusId": "eb-l65v1c2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "bfbf943a-0032-461c-89e6-260f7a3d520d",
    "Rules": [
      {
        "AddTime": "2021-04-29T00:06:22+08:00",
        "Description": "",
        "Enable": true,
        "EventBusId": "eb-l65v1c2u",
        "ModTime": "2021-04-28T16:06:22+08:00",
        "RuleId": "rule-fdltium8",
        "RuleName": "sssss",
        "Status": "Active",
        "Targets": null,
        "DeadLetterConfig": null
      },
      {
        "AddTime": "2021-04-28T16:00:14+08:00",
        "Description": "",
        "Enable": true,
        "EventBusId": "eb-l65v1c2u",
        "ModTime": "2021-04-28T16:00:14+08:00",
        "RuleId": "rule-m5ut05gs",
        "RuleName": "rule",
        "Status": "Active",
        "Targets": null,
        "DeadLetterConfig": null
      }
    ],
    "TotalCount": 2
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.Filters	Filters取值与规范不符，请修正后再试。
InvalidParameterValue.Limit	Limit取值与规范不符，请修正后再试。
InvalidParameterValue.Order	Order取值与规范不符，请修正后再试。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

更新事件规则

最近更新时间：2023-11-30 03:54:51

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

更新事件规则

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：UpdateRule。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
RuleId	是	String	事件规则ID 示例值：rule-xxxxxxxx
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxx
Enable	否	Boolean	使能开关。 示例值：true
Description	否	String	规则描述，不限字符类型，200字符描述以内。 示例值：这是一个描述
EventPattern	否	String	参考： 事件模式 示例值：{ "detail": { "source-ip": [{ "cidr": "10.0.0.0/24" }] }
RuleName	否	String	事件规则名称，只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2~60个字符 示例值：rule

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新规则

输入示例

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

```
{
  "RuleId": "rule-fdltium8",
  "EventBusId": "eb-l65v1c2",
  "RuleName": "sssss"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e5d57478-9baf-483d-a5db-b1067424e469"
  }
}
```

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.UpdateRule	规则更新失败，请检查资源信息，确认是资源否存在或状态正常。
InvalidParameter.Payload	Payload与规范不符，请修正后再试。
InvalidParameterValue.Description	Description取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.EventPattern	EventPattern取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.RuleName	RuleName取值与规范不符，请修正后再试。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。

错误码	描述
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。

获取事件规则详情

最近更新时间：2023-11-30 03:54:52

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取事件规则详情

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：GetRule。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxx
RuleId	是	String	事件规则ID 示例值：rule-xxxxxxxx

3. 输出参数

参数名称	类型	描述
EventBusId	String	事件集id 示例值：eb-xxxxxxxx
RuleId	String	事件规则id 示例值：rule-xxxxxxxx
RuleName	String	事件规则名称 示例值：rule
Status	String	事件规则状态 示例值：Active
Enable	Boolean	使能开关 示例值：true
Description	String	事件规则描述 示例值：这是一个描述
EventPattern	String	事件模式 示例值：{ "detail": { "source-ip": [{ "cidr": "10.0.0.0/24" }] }
AddTime	Timestamp ISO8601	创建时间 示例值：2021-04-27T14:34:19+08:00
ModTime	Timestamp ISO8601	更新时间 示例值：2021-04-27T14:34:19+08:00
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取规则详情

获取规则详情

输入示例

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

{
  "EventBusId": "eb-l65v1c2",
  "RuleId": "rule-qr2wsqqy"
}
```

输出示例

```
{
  "Response": {
    "AddTime": "2021-04-29T20:01:27+08:00",
    "Description": "",
    "Enable": true,
    "EventBusId": "eb-l65v1c2u",
    "EventPattern": "{\"data\":{\"prefix\":{\"1\",\"2\"}}}",
    "ModTime": "2021-04-29T20:01:28+08:00",
    "RequestId": "9babdbaf-85d9-4d79-8eb7-14fd989cf99a",
    "RuleId": "rule-qr2wsqqy",
    "RuleName": "test",
    "Status": "Active"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

• [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalError	内部错误。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。

事件连接器相关接口

创建事件连接器

最近更新时间：2023-12-14 01:43:59

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

创建事件连接器

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：CreateConnection。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ConnectionDescription	是	ConnectionDescription	连接器描述
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxxx
ConnectionName	是	String	连接器名称 示例值：connection
Description	否	String	描述 示例值：这是一个描述
Enable	否	Boolean	使能开关 示例值：true
Type	否	String	连接器类型，目前支持以下类型：apigw/ckafka/dts/tdmq 示例值：apigw

3. 输出参数

参数名称	类型	描述
ConnectionId	String	连接器ID 示例值：connection-xxxxxxxxx
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建连接器

输入示例

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

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

{
  "EventBusId": "eb-l65v1c2",
  "Type": "tdmq",
  "ConnectionDescription": {
    "ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1",
    "APIGWParams": null
  },
  "ConnectionName": "conn"
}
```

输出示例

```
{
  "Response": {
    "ConnectionId": "connection-5t492ybt",
    "RequestId": "99d8d400-2bde-49d6-99f4-7367907e5964"
  }
}
```

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.ServiceError	ServiceError操作失败，请检查资源信息。
InternalServerError.OssException	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。

错误码	描述
InvalidParameterValue.ConnectionDescription	ConnectionDescription取值与规范不符，请修正后再试。
InvalidParameterValue.ConnectionName	ConnectionName取值与规范不符，请修正后再试。
InvalidParameterValue.DTSParams	DTSParams取值与规范不符，请修正后再试。
InvalidParameterValue.Description	Description取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusName	EventBusName取值与规范不符，请修正后再试。
InvalidParameterValue.InvalidApiRequestConfig	InvalidApiRequestConfig取值与规范不符，请修正后再试。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。
LimitExceeded.Connection	Connection数量达到限制，可提交工单申请提升限制。
LimitExceeded.RouteOverLimit	RouteOverLimit数量达到限制，可提交工单申请提升限制。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
OperationDenied.UnsupportedOperation	当前用户账号类型暂不支持操作，请提交工单处理。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.InvalidService	连接器操作失败，未找到指定 API 网关服务。
ResourceNotFound.Role	未找到指定的服务角色，请创建后再试。
ResourceNotFound.Target	未找到指定的Target，请创建后再试。
ResourceUnavailable.Connection	连接器创建失败，当前资源不可用。
UnsupportedOperation	操作不支持。
UnsupportedOperation.InvalidAction	连接器操作失败，接口不支持当前操作。
UnsupportedOperation.InvalidEndpointType	连接器创建失败，不支持当前后端服务类型。

删除事件连接器

最近更新时间：2023-11-30 03:54:49

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

删除事件连接器

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：DeleteConnection。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ConnectionId	是	String	连接器ID 示例值：connection-xxxxxxxxx
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除连接器

输入示例

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

{
  "ConnectionId": "connection-ljwdx09",
  "EventBusId": "eb-l65v1c2"
}
```

输出示例


```
{
  "Response": {
    "RequestId": "4ec69b78-1e6a-4c39-b6e0-217d496ada7d"
  }
}
```

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.DeleteConnection	删除连接器失败，请检查资源状态。
FailedOperation.ServiceError	ServiceError操作失败，请检查资源信息。
InternalServerError.OssException	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。
InvalidParameterValue.ConnectionDescription	ConnectionDescription取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
ResourceNotFound	资源不存在。
ResourceNotFound.Connection	未找到指定的Connection，请创建后再试。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.InvalidApi	连接器删除失败，未找到指定 API 。
ResourceNotFound.InvalidService	连接器操作失败，未找到指定 API 网关服务。
UnsupportedOperation.InvalidAction	连接器操作失败，接口不支持当前操作。

获取事件连接器列表

最近更新时间：2023-11-30 03:54:49

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取事件连接器列表

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：ListConnections。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxx
OrderBy	否	String	根据哪个字段进行返回结果排序，目前支持如下以下字段：AddTime, ModTime 示例值：ModTime
Limit	否	Integer	返回数量，默认为20，最大值为100。 示例值：20
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC 和 DESC 示例值：DESC
Offset	否	Integer	偏移量，默认为0。 示例值：0

3. 输出参数

参数名称	类型	描述
Connections	Array of Connection	连接器信息
TotalCount	Integer	连接器总数 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取连接器列表

输入示例

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

<公共请求参数>

```
{
  "EventBusId": "eb-l65v1c2"
}
```

输出示例

```
{
  "Response": {
    "Connections": [],
    "RequestId": "cec4c711-4e68-43ac-988f-00171f2b2146",
    "TotalCount": 0
  }
}
```

示例2 获取连接器列表1

输入示例

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

{
  "EventBusId": "eb-l65v1c2"
}
```

输出示例

```
{
  "Response": {
    "Connections": [
      {
        "AddTime": "2021-04-29T13:13:43+08:00",
        "ConnectionDescription": {
          "APIGWParams": null,
          "ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
        },
        "ConnectionId": "connection-5t492ybt",
        "ConnectionName": "conn",
        "Description": "",
        "Enable": false,
        "EventBusId": "eb-l65v1c2u",
        "ModTime": "2021-04-29T13:13:44+08:00",
        "Status": "Active",
        "Type": "tdmq"
      },
      {
        "AddTime": "2021-04-29T13:00:32+08:00",
        "ConnectionDescription": {
          "APIGWParams": null,
          "ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
        }
      }
    ]
  }
}
```

```
},
"ConnectionId": "connection-7mpfojtt",
"ConnectionName": "conn",
"Description": "",
"Enable": false,
"EventBusId": "eb-l65v1c2u",
"ModTime": "2021-04-29T13:00:33+08:00",
"Status": "Failed",
"Type": "tdmq"
},
{
"AddTime": "2021-04-29T12:58:24+08:00",
"ConnectionDescription": {
"APIGWParams": null,
"ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
},
"ConnectionId": "connection-il11bb0h",
"ConnectionName": "conn",
"Description": "",
"Enable": false,
"EventBusId": "eb-l65v1c2u",
"ModTime": "2021-04-29T12:58:24+08:00",
"Status": "Failed",
"Type": "tdmq"
},
{
"AddTime": "2021-04-29T12:01:34+08:00",
"ConnectionDescription": {
"APIGWParams": null,
"ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
},
"ConnectionId": "connection-6s29eqob",
"ConnectionName": "conn",
"Description": "",
"Enable": false,
"EventBusId": "eb-l65v1c2u",
"ModTime": "2021-04-29T12:01:35+08:00",
"Status": "Failed",
"Type": "tdmq"
},
{
"AddTime": "2021-04-29T11:46:27+08:00",
"ConnectionDescription": {
"APIGWParams": null,
"ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
},
"ConnectionId": "connection-dka58wiv",
"ConnectionName": "conn",
"Description": "",
"Enable": false,
"EventBusId": "eb-l65v1c2u",
"ModTime": "2021-04-29T11:46:28+08:00",
"Status": "Failed",
"Type": "tdmq"
},
{
"AddTime": "2021-04-29T11:42:44+08:00",
```

```
"ConnectionDescription": {
  "APIGWParams": null,
  "ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
},
"ConnectionId": "connection-r1o76r2d",
"ConnectionName": "conn",
"Description": "",
"Enable": false,
"EventBusId": "eb-l65v1c2u",
"ModTime": "2021-04-29T11:42:44+08:00",
"Status": "Failed",
"Type": "tdmq"
},
{
  "AddTime": "2021-04-29T11:35:12+08:00",
  "ConnectionDescription": {
    "APIGWParams": null,
    "ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
  },
  "ConnectionId": "connection-ay3ix129",
  "ConnectionName": "conn",
  "Description": "",
  "Enable": false,
  "EventBusId": "eb-l65v1c2u",
  "ModTime": "2021-04-29T11:35:13+08:00",
  "Status": "Failed",
  "Type": "tdmq"
},
{
  "AddTime": "2021-04-29T11:24:45+08:00",
  "ConnectionDescription": {
    "APIGWParams": null,
    "ResourceDescription": "qcs::tdmq:ap-guangzhou:uin/3473058547:subscriptionName/pulsar-5r5drqo7j8/user/test/sub1"
  },
  "ConnectionId": "connection-heno97b7",
  "ConnectionName": "conn",
  "Description": "",
  "Enable": false,
  "EventBusId": "eb-l65v1c2u",
  "ModTime": "2021-04-29T11:24:45+08:00",
  "Status": "Failed",
  "Type": "tdmq"
}
],
"RequestId": "e576a375-88ae-4c3f-84a9-0bf28dc7b0ee",
"TotalCount": 8
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.Limit	Limit取值与规范不符，请修正后再试。
InvalidParameterValue.Offset	Offset取值与规范不符，请修正后再试。
InvalidParameterValue.Order	Order取值与规范不符，请修正后再试。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

更新事件连接器

最近更新时间：2023-11-30 03:54:48

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

更新事件连接器

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：UpdateConnection。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ConnectionId	是	String	连接器ID 示例值：connection-xxxxxxx
EventBusId	是	String	事件集ID 示例值：eb-xxxxxxx
Enable	否	Boolean	使能开关 示例值：true
Description	否	String	描述 示例值：这是一个描述
ConnectionName	否	String	连接器名称 示例值：connection

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新连接器

输入示例

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

{
  "EventBusId": "eb-l65vlc2u",
```

```
"ConnectionId": "connection-l65vlc2u"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b3103b8d-77a6-4b55-b42f-a25e2dc453a9"
  }
}
```

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.UpdateConnection	连接器更新失败，请检查资源信息，确认是资源是否存在或状态正常。
InvalidParameter.EnableAPIGateway	APIGateway连接器不支持开启或关闭操作，请修正后重试。
InvalidParameterValue.ConnectionId	ConnectionId取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
ResourceNotFound.Connection	未找到指定的Connection，请创建后再试。

数据转换器相关接口

检查转换器

最近更新时间：2023-11-30 03:54:44

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

用于在ETL配置页面, 测试规则和数据。

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：CheckTransformation。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Input	是	String	待处理的json字符串 示例值：{"key": "value"}
Transformations.N	是	Array of Transformation	一个转换规则列表

3. 输出参数

参数名称	类型	描述
Output	String	经过Transformations处理之后的数据 示例值：{"key2": "value2"}
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 检查转换器

输入示例

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

{
  "Input": "{\"data\":{\"msgBody\":{\"city\":\"shenzhen\",\"temp\":\"32\",\"weather\":\"sunny\"}}}",
  "Transformations": [
    {
      "Extraction": {
        "ExtractionInputPath": "$.data.msgBody",
```

```
"Format": "JSON",
"TextParams": {
  "Separator": ":",
  "Regex": "shenzhen"
},
"EtlFilter": {
  "Filter": "{\"city\":{\"contain\":\"shen\"}}\""
},
"Transform": {
  "OutputStructs": [
    {
      "Key": "city",
      "Value": ".$city",
      "ValueType": "JSONPATH"
    },
    {
      "Key": "const",
      "Value": "11",
      "ValueType": "NUMBER"
    },
    {
      "Key": "sdate",
      "Value": "date",
      "ValueType": "SYS_VARIABLE"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "589fd30c-49c9-11ec-9ad1-5254006e5bc5",
    "Output": "{\"sdate\":\"2021-11-20T14:16:06+08:00\",\"city\":\"shenzhen\",\"const\":\"11\"}"
  }
}
```

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](#)。

创建转换器

最近更新时间：2023-12-15 19:35:51

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

用于创建转换器

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：CreateTransformation。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件总线 id 示例值：eb-xxxxxx
RuleId	是	String	规则id 示例值：rule-xxxxxxx
Transformations.N	是	Array of Transformation	一个转换规则列表，当前仅限定一个;示例如下： [{"Extraction": {"ExtractionInputPath": "\$.data.payload", "Format": "JSON", "EtlFilter": {"Filter": {"source": "ckafka.cloud.tencent"}}, "Transform": {"OutputStructs": [{"Key": "op", "Value": "\$.op", "ValueType": "JSONPATH"}, {"Key": "table", "Value": "\$.source.table", "ValueType": "JSONPATH"}, {"Key": "id", "Value": "\$.after.id", "ValueType": "JSONPATH"}, {"Key": "app_id", "Value": "\$.after.app_id", "ValueType": "JSONPATH"}, {"Key": "spu_id", "Value": "\$.after.spu_id", "ValueType": "JSONPATH"}]}}] 示例值： [{"Extraction": {"ExtractionInputPath": "\$.data.payload", "Format": "JSON", "EtlFilter": {"Filter": {"source": "ckafka.cloud.tencent"}}, "Transform": {"OutputStructs": [{"Key": "op", "Value": "\$.op", "ValueType": "JSONPATH"}, {"Key": "table", "Value": "\$.source.table", "ValueType": "JSONPATH"}, {"Key": "id", "Value": "\$.after.id", "ValueType": "JSONPATH"}, {"Key": "app_id", "Value": "\$.after.app_id", "ValueType": "JSONPATH"}, {"Key": "spu_id", "Value": "\$.after.spu_id", "ValueType": "JSONPATH"}]}}]

3. 输出参数

参数名称	类型	描述
TransformationId	String	生成的转换器id 示例值：tsfm-xxxxxxxxxxxxx
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建转换器

输入示例

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

{
  "EventBusId": "eb-xxxxx",
  "RuleId": "rule-xxxxx",
  "Transformations": [
    {
      "Extraction": {
        "ExtractionInputPath": "$",
        "Format": "JSON"
      },
      "EtlFilter": {
        "Filter": "{\"source\":\"ckafka.cloud.tencent\"}"
      },
      "Transform": {
        "OutputStructs": [
          {
            "Key": "data",
            "Value": "$.data",
            "ValueType": "STRING"
          },
          {
            "Key": "age",
            "Value": "$.age",
            "ValueType": "NUMBER"
          }
        ]
      }
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "TransformationId": "tsfm-xxxxxxx",
    "RequestId": "e3d43926-c2cd-49f2-97f0-53db21e6fcea"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.Transformations	数据转换任务创建失败，配置参数取值与规范不符，请修正后再试。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。

删除转换器

最近更新时间：2023-11-30 03:54:43

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

用于删除转换器

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：DeleteTransformation。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxx
RuleId	是	String	规则ID 示例值：rule-xxxxxxx
TransformationId	是	String	转换器id 示例值：tsfm-xxxxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除转换器

输入示例

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

{
  "EventBusId": "eb-xxxxx",
  "RuleId": "rule-xxxxxxxxx",
  "TransformationId": "tsfm-xxxxxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xxxxxxx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.TransformationID	数据转换配置参数取值与规范不符，请修正后再试。
ResourceNotFound.Transformation	未找到指定的转换任务，请创建后再试。

获取转换器详情

最近更新时间：2023-11-30 03:54:42

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

用于获取转换器详情

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：GetTransformation。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxx
RuleId	是	String	规则ID 示例值：rule-xxxxxx
TransformationId	是	String	转换器id 示例值：tsfm-xxxxxx

3. 输出参数

参数名称	类型	描述
Transformations	Array of Transformation	转换规则列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取转换器详情

输入示例

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

{
  "EventBusId": "eb-xxxxx",
  "RuleId": "rule-xxxxxxxxx",
  "TransformationId": "tsfm-xxxxxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx",
    "Transformations": [
      {
        "EtlFilter": {
          "Filter": "{\"source\":\"ckafka.cloud.tencent\"}"
        },
        "Extraction": {
          "ExtractionInputPath": "$",
          "Format": "JSON",
          "TextParams": {
            "Separator": ","
          }
        },
        "Transform": {
          "OutputStructs": [
            {
              "Key": "age",
              "Value": "$.age",
              "ValueType": "NUMBER"
            }
          ]
        }
      }
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.TransformationID	数据转换配置参数取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。
ResourceNotFound.Transformation	未找到指定的转换任务，请创建后再试。

更新转换器

最近更新时间：2023-11-30 03:54:42

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

用于更新转换器

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：UpdateTransformation。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventBusId	是	String	事件集ID 示例值：eb-xxxxxx
RuleId	是	String	规则ID 示例值：rule-xxxxxxxxxx
TransformationId	是	String	转换器id 示例值：tsfm-xxxxxxxxxx
Transformations.N	是	Array of Transformation	一个转换规则列表，当前仅限定一个

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新转换器

输入示例

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

{
  "EventBusId": "eb-xxxxx",
  "RuleId": "rule-xxxxx",
  "TransformationId": "tsfm-xxxxx",
  "Transformations": [
```

```
{
  "Extraction": {
    "ExtractionInputPath": "$",
    "Format": "JSON"
  },
  "EtlFilter": {
    "Filter": "{\"source\":\"ckafka.cloud.tencent\"}"
  },
  "Transform": {
    "OutputStructs": [
      {
        "Key": "data",
        "Value": "$.data",
        "ValueType": "STRING"
      },
      {
        "Key": "age",
        "Value": "$.age",
        "ValueType": "NUMBER"
      }
    ]
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e3d43926-c2cd-49f2-97f0-53db21e6fcea"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InvalidParameterValue.TransformationID	数据转换配置参数取值与规范不符，请修正后再试。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

事件存储相关接口

日志检索

最近更新时间：2023-12-15 19:35:55

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

日志检索

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：SearchLog。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
StartTime	是	Integer	起始时间unix 毫秒时间戳 示例值：1702538123708
EndTime	是	Integer	结束时间unix 毫秒时间戳 示例值：1702539023708
EventBusId	是	String	事件集ID 示例值：eb-xxxxxx
Page	是	Integer	页码 示例值：1
Limit	是	Integer	每页数据大小 示例值：10
Filter.N	否	Array of LogFilter	事件查询筛选条件；示例如下： [{"key":"host","operator":"eq","value":"106.53.106.243"}, {"type":"AND","filters":[{"key":"region","operator":"like","value":"guangzhou"}, {"key":"type","operator":"eq","value":"cvm:ErrorEvent:GuestReboot"}]}, {"type":"OR","filters":[{"key":"field1","operator":"like","value":"access"}, {"key":"field2","operator":"eq","value":"custom"}]}] 示例值： [{"key":"host","operator":"eq","value":"106.53.106.243"}, {"type":"AND","filters":[{"key":"region","operator":"like","value":"guangzhou"}, {"key":"type","operator":"eq","value":"cvm:ErrorEvent:GuestReboot"}]}, {"type":"OR","filters":[{"key":"field1","operator":"like","value":"access"}, {"key":"field2","operator":"eq","value":"custom"}]}]
OrderFields.N	否	Array of String	事件查询结果排序，["timestamp","subject"] 示例值：["timestamp","subject"]
OrderBy	否	String	排序方式，asc 从旧到新，desc 从新到旧 示例值：desc

3. 输出参数

参数名称	类型	描述
Total	Integer	日志总数 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
Limit	Integer	每页日志条数 注意：此字段可能返回 null，表示取不到有效值。
Page	Integer	页码 注意：此字段可能返回 null，表示取不到有效值。
Results	Array of SearchLogResult	日志检索结果 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，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 日志检索

日志检索

输入示例

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

{
  "StartTime": 1673233483024,
  "EndTime": 1673838283024,
  "EventBusId": "eb-xxxxx",
  "Page": 1,
  "Limit": 1000
}
```

输出示例

```
{
  "Response": {
    "RequestId": "584caa6b-26d8-4ba5-858d-df1182730075",
    "Results": [
      {
        "Timestamp": "xxx",
        "Message": "xxx-1",
        "Source": "xxx",
        "Type": "xzz",
        "RuleIds": "xxx",
        "Subject": "xxx",
        "Region": "xxx",
        "Status": "xxx"
      }
    ],
    "Total": 1000,
    "Limit": 500,
    "Page": 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
InternalServerError	内部错误。
InvalidParameterValue.Action	Action 操作参数取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。

查询日志索引维度值

最近更新时间：2023-12-15 19:35:56

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

查询日志索引维度值

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：DescribeLogTagValue。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
StartTime	是	Integer	起始时间
EndTime	是	Integer	结束时间
EventBusId	是	String	事件集ID 示例值：eb-xxxxxx
GroupField	是	String	聚合字段,取值范围如下：Source(事件源),RuleIds(命中规则),Subject(实例ID),Region(地域) 示例值：RuleIds
Page	是	Integer	页数 示例值：1
Limit	是	Integer	每页数据大小 示例值：100
Filter.N	否	Array of LogFilter	筛选条件 示例值：[{ "key": "host", "operator": "eq", "value": "106.53.106.243" }, { "type": "AND", "filters": [{ "key": "region", "operator": "like", "value": "guangzhou" }, { "key": "type", "operator": "eq", "value": "cvm:ErrorEvent:GuestReboot" }] }, { "type": "OR", "filters": [{ "key": "field1", "operator": "like", "value": "access" }, { "key": "field2", "operator": "eq", "value": "custom" }] }] }

3. 输出参数

参数名称	类型	描述
Results	Array of String	索引检索维度值 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，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询日志索引字段值

查询日志索引字段值

输入示例

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

```
{
  "StartTime": 1673233483024,
  "EndTime": 1673838283024,
  "EventBusId": "eb-xxxxx",
  "GroupField": "Source",
  "Page": 1,
  "Limit": 1000
}
```

输出示例

```
{
  "Response": {
    "Results": [
      "eb-l65v1c2u"
    ],
    "RequestId": "cfa5ba79-4169-44d5-a1c4-a95b8a8b4481"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。

错误码	描述
InvalidParameterValue.Action	Action 操作参数取值与规范不符，请修正后再试。
ResourceNotFound	资源不存在。

最近更新时间：2023-12-14 01:44:00

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取平台产品事件匹配规则

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ListPlatformEventPatterns。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ProductType	是	String	平台产品类型 示例值：eb_platform_test,该值根据ListPlatformProducts接口获取

3. 输出参数

参数名称	类型	描述
EventPatterns	Array of PlatformEventSummary	平台产品事件匹配规则
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询平台产品事件匹配规则

输入示例

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

{
  "ProductType": "eb_platform_test"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "584caa6b-26d8-4ba5-858d-df1182730075",
    "EventPatterns": [
      {
```

腾讯云 API 平台

API Inspector

SDK

- 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](#)

- Tencent Cloud CLI 3.0

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](#)。

获取平台产品事件名称

最近更新時間：2023-11-30 03:54:57

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取平台产品事件名称

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ListPlatformEventNames。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ProductType	是	String	平台产品类型 示例值：eb_platform_test

3. 输出参数

参数名称	类型	描述
EventNames	Array of PlatformEventDetail	平台产品列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询平台事件中文名称

输入示例

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

{
  "ProductType": "eb_platform_test"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "584caa6b-26d8-4ba5-858d-df1182730075",
```

```
"EventNames": [
{
  "EventName": "TEST",
  "EventType": "eb_platform_test:TEST:ALL"
}
]
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。

获取平台产品列表

最近更新时间：2023-11-30 03:54:56

1. 接口描述

接口请求域名：eb.tencentcloudapi.com。

获取平台产品列表

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：ListPlatformProducts。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

3. 输出参数

参数名称	类型	描述
PlatformProducts	Array of PlatformProduct	平台产品列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取平台产品列表

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "ffd4aae2-c29e-40a8-b18c-037a17ed810c",
    "PlatformProducts": [
      {
        "ProductName": "告警平台事件测试",
        "ProductType": "alarm"
      }
    ]
  }
}
```

```
}  
}  
}  
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameterValue.Action	Action 操作参数取值与规范不符，请修正后再试。

最近更新时间: 2023-12-14 01:44:01

接口请求域名: `eb.tencentcloudapi.com`。

获取平台产品事件模板

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。

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：GetPlatformEventTemplate。
Version	是	String	公共参数 ，本接口取值：2021-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EventType	是	String	平台产品事件类型 示例值：eb_platform_test:TEST:ALL，此值根据ListPlatformEventNames接口获取

参数名称	类型	描述
EventTemplate	String	平台产品事件模板
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

示例1 获取平台产品事件模板

输入示例

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

{
  "EventType": "eb_platform_test:TEST:ALL"
}
```

输出示例

```
{  
  "Response": {  
    "EventTemplate": "{\n\t\t\t\t\t\"specversion\": \"1.0\",\n\t\t\t\t\t\"id\": \"752e37c8-3528-47f1-8311-98c01e2e89c4\",\n\t\t\t\t\t\"source\": \"platform.c
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameterValue.Action	Action 操作参数取值与规范不符，请修正后再试。

数据结构

最近更新时间：2023-12-15 19:35:56

APIGWParams

APIGWParams描述

被如下接口引用：CreateConnection, ListConnections。

名称	类型	必选	描述
Protocol	String	是	HTTPS 示例值：HTTPS
Method	String	是	POST 示例值：POST

CkafkaDeliveryParams

用来描述需要投递到kafka topic的参数

被如下接口引用：ListRules。

名称	类型	必选	描述
TopicName	String	是	ckafka topic name 示例值：topic-xxxxxx
ResourceDescription	String	是	ckafka资源qcs六段式 示例值：qcs::ckafka:ap-chongqing:uin/123456:ckafkald/uin/456789/ckafka-x

CkafkaParams

Ckafka 连接器参数

被如下接口引用：CreateConnection, ListConnections。

名称	类型	必选	描述
Offset	String	是	kafka offset 示例值：latest
TopicName	String	是	ckafka topic 示例值：topic

CkafkaTargetParams

用来描述ckafka投递目标

被如下接口引用：CreateTarget。

名称	类型	必选	描述
TopicName	String	是	要投递到的ckafka topic 示例值：topic-xxxx
RetryPolicy	RetryPolicy	是	重试策略

Connection

Connection信息

被如下接口引用：ListConnections。

名称	类型	描述
----	----	----

名称	类型	描述
Status	String	状态 示例值: Active
ModTime	Timestamp ISO8601	更新时间 示例值: 2021-04-27T14:34:19+08:00
Enable	Boolean	使能开关 示例值: true
Description	String	描述 示例值: 这是一个描述
AddTime	Timestamp ISO8601	创建时间 示例值: 2021-04-27T14:34:19+08:00
ConnectionId	String	连接器ID 示例值: connection-xxxxxxxxx
EventBusId	String	事件集ID 示例值: eb-xxxxxxxxx
ConnectionDescription	ConnectionDescription	连接器描述
ConnectionName	String	连接器名称 示例值: connection
Type	String	类型 示例值: apigw

ConnectionBrief

连接器基础信息

被如下接口引用: ListEventBuses。

名称	类型	描述
Type	String	连接器类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: tdmq
Status	String	连接器状态 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: Normal

ConnectionDescription

ConnectionDescription描述

被如下接口引用: CreateConnection, ListConnections。

名称	类型	必选	描述
ResourceDescription	String	是	资源qcs六段式, 更多参考 资源六段式 示例值: qcs::apigw:ap-guangzhou:uin/0000000:serviceid/service-id
APIGWParams	APIGWParams	否	apigw参数 注意: 此字段可能返回 null, 表示取不到有效值。
CkafkaParams	CkafkaParams	否	ckafka参数 注意: 此字段可能返回 null, 表示取不到有效值。
DTSParams	DTSParams	否	data transfer service (DTS)参数 注意: 此字段可能返回 null, 表示取不到有效值。
TDMQParams	TDMQParams	否	tdmq参数 注意: 此字段可能返回 null, 表示取不到有效值。

DTSParams

Data Transfer Service参数

被如下接口引用：CreateConnection, ListConnections。

名称	类型	必选	描述
ConsumerGroupName	String	是	Consumer Group Name 示例值：consumer-grp-subs-abcde12345-consumer-group-name
Account	String	是	账户名 示例值：user-account
Password	String	是	密码 注意：此字段可能返回 null，表示取不到有效值。 示例值：user-pwd

DeadLetterConfig

rule对应的dlq配置

被如下接口引用：ListRules。

名称	类型	必选	描述
DisposeMethod	String	是	支持dlq、丢弃、忽略错误继续传递三种模式，分别对应：DLQ,DROP,IGNORE_ERROR 示例值：DLQ
CkafkaDeliveryParams	CkafkaDeliveryParams	否	设置了DLQ方式后,此选项必填。 错误消息会被投递到对应的kafka topic中 注意：此字段可能返回 null，表示取不到有效值。

ESTargetParams

描述Es规则目标

被如下接口引用：CreateTarget。

名称	类型	必选	描述
NetMode	String	是	网络连接类型 示例值：privatelink
IndexPrefix	String	是	索引前缀 示例值：esindex
RotationInterval	String	是	es日志轮换粒度 示例值：none
OutputMode	String	是	DTS事件配置 示例值：default
IndexSuffixMode	String	是	DTS索引配置 示例值：default
IndexTemplateType	String	否	es模版类型 示例值：_type

EtFilter

描述如何过滤数据

被如下接口引用：CheckTransformation, CreateTransformation, GetTransformation, UpdateTransformation。

名称	类型	必选	描述
Filter	String	是	语法规则保持一致 示例值：{"source":"ckafka.cloud.tencent"}

Event

eb event信息

被如下接口引用：PublishEvent, PutEvents。

名称	类型	必选	描述
Source	String	是	事件源的信息,新产品上报必须符合EB的规范 示例值：cvm.cloud.tencent
Data	String	是	事件数据，内容由创建事件的系统来控制，当前datacontenttype仅支持application/json;charset=utf-8，所以该字段是json字符串 示例值：{"a":"b","c":"d","e":1}"
Type	String	是	事件类型，可自定义，选填。云服务默认写 COS:Created:PostObject，用 “: ” 分割类型字段 示例值：cvm:ErrorEvent:PingUnreachable
Subject	String	是	事件来源详细描述，可自定义，选填。云服务默认为标准qcs资源表示语法：qcs::dts:ap-guangzhou:appid/uin:xxx 示例值：qcs::dts:ap-guangzhou:appid/uin:xxx
Time	Integer	否	事件发生的毫秒时间戳， time.Now().UnixNano()/1e6 示例值：1631186355147

EventBus

事件集信息

被如下接口引用：ListEventBuses。

名称	类型	描述
ModTime	Timestamp ISO8601	更新时间 示例值：2021-04-27T14:34:19+08:00
Description	String	事件集描述，不限字符类型，200字符描述以内 示例值：这是一个描述
AddTime	Timestamp ISO8601	创建时间 示例值：2021-04-27T14:34:19+08:00
EventBusName	String	事件集名称，只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2~60个字符 示例值：EventBus
EventBusId	String	事件集ID 示例值：eb-xxxxxxxxx
Type	String	事件集类型 示例值：Cloud
PayMode	String	计费模式 注意：此字段可能返回 null，表示取不到有效值。 示例值：Postpaid(后付费)
ConnectionBriefs	Array of ConnectionBrief	连接器基础信息 注意：此字段可能返回 null，表示取不到有效值。
TargetBriefs	Array of TargetBrief	目标简要信息 注意：此字段可能返回 null，表示取不到有效值。

Extraction

描述如何提取数据

被如下接口引用：CheckTransformation, CreateTransformation, GetTransformation, UpdateTransformation。

名称	类型	必选	描述
----	----	----	----

名称	类型	必选	描述
ExtractionInputPath	String	是	JsonPath, 不指定则使用默认值\$。 示例值: \$
Format	String	是	取值: TEXT/JSON 示例值: TEXT
TextParams	TextParams	否	仅在Text需要传递 注意: 此字段可能返回 null, 表示取不到有效值。

Filter

描述键值对过滤器，用于条件过滤查询。例如过滤ID、名称、状态等

- 若存在多个Filter时，Filter间的关系为逻辑与（AND）关系。
- 若同一个Filter存在多个Values，同一Filter下Values间的关系为逻辑或（OR）关系。

被如下接口引用：ListEventBuses。

名称	类型	必选	描述
Values	Array of String	是	一个或者多个过滤值。 示例值: Cloud
Name	String	是	过滤键的名称。EventBusName(事件集名称)/EventBusId(事件集Id)/Type(事件集类型:Cloud(云服务);Platform(平台型);Custom(自定义))/TagKey(标签键) 示例值: Type

LogFilter

日志查询相关接口filter参数定义

被如下接口引用：DescribeLogTagValue, SearchLog。

名称	类型	必选	描述
Key	String	否	过滤字段名称 示例值: status
Operator	String	否	运算符，全等 eq，不等 neq，相似 like，排除相似 not like，小于 lt，小于且等于 lte，大于 gt，大于且等于 gte，在范围内 range，不在范围内 norange 示例值: eq
Value	String	否	过滤值,范围运算需要同时输入两个值，以英文逗号分隔 示例值: "0"
Type	String	否	该层级filters逻辑关系，取值 "AND" 或 "OR" 示例值: AND
Filters	Array of LogFilters	否	LogFilters数组

LogFilters

日志存储过滤条件

被如下接口引用：DescribeLogTagValue, SearchLog。

名称	类型	必选	描述
Key	String	是	过滤字段名称，取值范围如下:region(地域)，type(事件类型)，source(事件源)，status(事件状态) 示例值: status

名称	类型	必选	描述
Operator	String	是	运算符, 全等 eq, 不等 neq, 相似 like, 排除相似 not like, 小于 lt, 小于且等于 lte, 大于 gt, 大于且等于 gte, 在范围内 range, 不在范围内 norange 示例值: eq
Value	String	是	过滤值, 范围运算需要同时输入两个值, 以英文逗号分隔 示例值: "0"

OutputStructParam

Transform输出参数

被如下接口引用: CheckTransformation, CreateTransformation, GetTransformation, UpdateTransformation。

名称	类型	必选	描述
Key	String	是	对应输出json中的key 示例值: name
Value	String	是	可以填json-path也可以支持常量或者内置关键字date类型 示例值: \$.name
ValueType	String	是	value的数据类型, 可选值: STRING, NUMBER,BOOLEAN,NULL,SYS_VARIABLE,JSONPATH 示例值: JSONPATH

PlatformEventDetail

平台事件名称

被如下接口引用: ListPlatformEventNames。

名称	类型	描述
EventName	String	事件名称 注意: 此字段可能返回 null, 表示取不到有效值。
EventType	String	事件类型 注意: 此字段可能返回 null, 表示取不到有效值。

PlatformEventSummary

平台事件匹配规则

被如下接口引用: ListPlatformEventPatterns。

名称	类型	描述
EventName	String	平台事件名称 注意: 此字段可能返回 null, 表示取不到有效值。
EventPattern	String	平台事件匹配规则 注意: 此字段可能返回 null, 表示取不到有效值。

PlatformProduct

平台产品基础信息

被如下接口引用: ListPlatformProducts。

名称	类型	描述
ProductName	String	平台产品名称
ProductType	String	平台产品类型

RetryPolicy

用来描述一个ckafka投递目标的重试策略

被如下接口引用：CreateTarget。

名称	类型	必选	描述
RetryInterval	Integer	是	重试间隔 单位:秒 示例值：30
MaxRetryAttempts	Integer	是	最大重试次数 示例值：3

Rule

规则信息

被如下接口引用：ListRules。

名称	类型	描述
Status	String	状态 示例值：Active
ModTime	Timestamp ISO8601	修改时间 示例值：2021-04-27T14:34:19+08:00
Enable	Boolean	使能开关 示例值：true
Description	String	描述 示例值：这是一个描述
RuleId	String	规则ID 示例值：rule-xxxxxxx
AddTime	Timestamp ISO8601	创建时间 示例值：2021-04-27T14:34:19+08:00
EventBusId	String	事件集ID 示例值：eb-xxxxxxx
RuleName	String	规则名称 示例值：rule
Targets	Array of TargetBrief	Target 简要信息 注意：此字段可能返回 null，表示取不到有效值。
DeadLetterConfig	DeadLetterConfig	rule设置的dlq规则。可能为null 注意：此字段可能返回 null，表示取不到有效值。

SCFParams

云函数参数

被如下接口引用：CreateTarget。

名称	类型	必选	描述
BatchTimeout	Integer	否	批量投递最长等待时间 示例值：1
BatchEventCount	Integer	否	批量投递最大事件条数 示例值：1
EnableBatchDelivery	Boolean	否	开启批量投递使能 示例值：false

SearchLogResult

日志检索详情

被如下接口引用：SearchLog。

名称	类型	描述
Timestamp	String	单条日志上报时间 注意：此字段可能返回 null，表示取不到有效值。
Message	String	日志内容详情 注意：此字段可能返回 null，表示取不到有效值。
Source	String	事件来源 注意：此字段可能返回 null，表示取不到有效值。
Type	String	事件类型 注意：此字段可能返回 null，表示取不到有效值。
RuleIds	String	事件匹配规则 注意：此字段可能返回 null，表示取不到有效值。 示例值：rule-xxxx
Subject	String	实例ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：ins-xxxx
Region	String	地域 注意：此字段可能返回 null，表示取不到有效值。 示例值：ap-guangzhou
Status	String	事件状态 注意：此字段可能返回 null，表示取不到有效值。

TDMQParams

TDMQ参数详情

被如下接口引用：CreateConnection, ListConnections。

名称	类型	必选	描述
ClusterType	String	否	集群类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：default:虚拟集群; exclusive:专业集群
ClusterEndPoint	String	否	集群支撑网接入点 注意：此字段可能返回 null，表示取不到有效值。

Target

Target信息

被如下接口引用：ListTargets。

名称	类型	描述
Type	String	目标类型 示例值：scf
EventBusId	String	事件集ID 示例值：eb-xxxxxxxx
TargetId	String	目标ID 示例值：target-xxxxxxxx
TargetDescription	TargetDescription	目标描述

名称	类型	描述
RuleId	String	事件规则ID 示例值：rule-xxxxxxxx
EnableBatchDelivery	Boolean	开启批量投递使能 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
BatchTimeout	Integer	批量投递最长等待时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
BatchEventCount	Integer	批量投递最大事件条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

TargetBrief

目标简要信息

被如下接口引用：ListEventBuses, ListRules。

名称	类型	描述
TargetId	String	目标ID 示例值：target-xxxxxxxx
Type	String	目标类型 示例值：scf

TargetDescription

TargetDescription描述

被如下接口引用：CreateTarget, ListTargets。

名称	类型	必选	描述
ResourceDescription	String	是	QCS资源六段式，更多参考 资源六段式 ；scf资源六段式示例[qcs::scf:ap-guangzhou:uin/123:namespace/test(函数命名空间)/function/test(函数名)/\$LATEST(函数版本)] amp资源六段式示例[qcs::eb-amp:ap-guangzhou:uin/123:] ckafka资源六段式示例[qcs::ckafka:ap-guangzhou:uin/123:ckafkald/uin/123/ckafka-123(ckafka实例Id)] cls资源六段式示例[qcs::cls:ap-guangzhou:uin/123:topic/122332442(topicId)] es资源六段式示例[qcs::es:ap-guangzhou:appid/123/uin/456:instance/es-7cplmhsd(es实例Id)] 示例值：qcs::es:ap-guangzhou:uin/1500000688:instance/es-7cplmhsd
SCFParams	SCFParams	否	云函数参数
CkafkaTargetParams	CkafkaTargetParams	否	Ckafka参数
ESTargetParams	ESTargetParams	否	ElasticSearch参数

TextParams

描述如何切分数据

被如下接口引用：CheckTransformation, CreateTransformation, GetTransformation, UpdateTransformation。

名称	类型	必选	描述
Separator	String	否	逗号、 、制表符、空格、换行符、%、#，限制长度为 1。 注意：此字段可能返回 null，表示取不到有效值。 示例值：,

名称	类型	必选	描述
Regex	String	否	填写正则表达式：长度128 注意：此字段可能返回 null，表示取不到有效值。 示例值：a(.*)bc

Transform

描述如何数据转换

被如下接口引用：CheckTransformation, CreateTransformation, GetTransformation, UpdateTransformation。

名称	类型	必选	描述
OutputStructs	Array of OutputStructParam	是	描述如何数据转换

Transformation

一个转换器

被如下接口引用：CheckTransformation, CreateTransformation, GetTransformation, UpdateTransformation。

名称	类型	必选	描述
Extraction	Extraction	否	描述如何提取数据，{"ExtractionInputPath":"\$.data.payload","Format":"JSON"} 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"ExtractionInputPath":"\$.data.payload","Format":"JSON"}
EtlFilter	EtlFilter	否	描述如何过滤数据；{"Filter":{"source":"ckafka.cloud.tencent"}} 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Filter":{"source":"ckafka.cloud.tencent"}}
Transform	Transform	否	描述如何数据转换;"OutputStructs":[{"Key":"op","Value":"\$.op","ValueType":"JSONPATH"}] 注意：此字段可能返回 null，表示取不到有效值。 示例值："OutputStructs":[{"Key":"op","Value":"\$.op","ValueType":"JSONPATH"}]

错误码

最近更新时间：2023-04-04 01:16:33

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.AddPrivateLink	添加私有链接失败。
FailedOperation.AuthenticateUserFailed	用户鉴权错误。
FailedOperation.CreateTrigger	投递目标创建失败，请检查函数状态。
FailedOperation.DeleteConnection	删除连接器失败，请检查资源状态。
FailedOperation.DeletePrivateLink	删除私有链接失败。
FailedOperation.DeleteRule	规则删除失败，请检查资源信息，确认是资源是否存在或状态正常。
FailedOperation.ESInternalError	ES集群内部错误。
FailedOperation.ESRequestFailed	ES集群操作失败。
FailedOperation.ESTemplateConflict	ES索引模板冲突错误。
FailedOperation.ErrorFilter	规则与事件不匹配，请修正后再试。
FailedOperation.RegisterCLSService	注册CLS服务失败。
FailedOperation.ServiceError	ServiceError操作失败，请检查资源信息。
FailedOperation.TagResource	标签绑定操作失败。
FailedOperation.TagResourceAllocateQuotas	申请标签配额失败。
FailedOperation.UnTagResource	标签解绑操作失败。

错误码	说明
FailedOperation.UpdateConnection	连接器更新失败，请检查资源信息，确认是资源是否存在或状态正常。
FailedOperation.UpdateRule	规则更新失败，请检查资源信息，确认是资源是否存在或状态正常。
InternalServerError.OssException	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。
InternalServerError.System	服务处理出错，请稍后重试。若无法解决，请联系智能客服或提交工单。
InternalServerError.UnknownInternalServerError	未知错误。
InvalidParameter.EnableAPIGateway	APIGateway连接器不支持开启或关闭操作，请修正后重试。
InvalidParameter.Payload	Payload与规范不符，请修正后再试。
InvalidParameterValue.AMPPParams	AMPPParams取值与规范不符，请修正后再试。
InvalidParameterValue.Action	Action 操作参数取值与规范不符，请修正后再试。
InvalidParameterValue.BatchEventCount	BatchEventCount取值与规范不符，请修正后再试。
InvalidParameterValue.BatchTimeout	BatchTimeout取值与规范不符，请修正后再试。
InvalidParameterValue.CKafkaTargetParams	Ckafka 目标配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.CallbackType	CallbackType取值与规范不符，请修正后再试。
InvalidParameterValue.CallbackWeComURL	CallbackWeComURL取值与规范不符，请修正后再试。
InvalidParameterValue.ConnectionDescription	ConnectionDescription取值与规范不符，请修正后再试。
InvalidParameterValue.ConnectionId	ConnectionId取值与规范不符，请修正后再试。
InvalidParameterValue.ConnectionName	ConnectionName取值与规范不符，请修正后再试。
InvalidParameterValue.DTSPParams	DTSPParams取值与规范不符，请修正后再试。
InvalidParameterValue.DeadLetterConfig	死信队列配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.Description	Description取值与规范不符，请修正后再试。
InvalidParameterValue.ElasticSearchTargetParams	ES目标参数错误。
InvalidParameterValue.EventBusId	EventBusId取值与规范不符，请修正后再试。
InvalidParameterValue.EventBusName	EventBusName取值与规范不符，请修正后再试。
InvalidParameterValue.EventPattern	EventPattern取值与规范不符，请修正后再试。
InvalidParameterValue.EventTraceConfig	链路追踪配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.Filters	Filters取值与规范不符，请修正后再试。
InvalidParameterValue.InvalidApiRequestConfig	InvalidApiRequestConfig取值与规范不符，请修正后再试。
InvalidParameterValue.InvalidEvent	事件格式非法，请修正后再试。
InvalidParameterValue.InvalidEventBus	非法的事件集，请检查后重试。
InvalidParameterValue.InvalidFilterRule	事件模式格式错误，请修正后再试。
InvalidParameterValue.InvalidPattern	非法的匹配规则，请检查后重试。
InvalidParameterValue.Limit	Limit取值与规范不符，请修正后再试。
InvalidParameterValue.LinkMode	LinkMode取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverChannel	NoticeReceiverChannel取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverTimeWindow	NoticeReceiverTimeWindow取值与规范不符，请修正后再试。
InvalidParameterValue.NoticeReceiverUserIds	NoticeReceiverUserIds取值与规范不符，请修正后再试。

错误码	说明
InvalidParameterValue.NoticeReceiverUserType	NoticeReceiverUserType取值与规范不符，请修正后再试。
InvalidParameterValue.Offset	Offset取值与规范不符，请修正后再试。
InvalidParameterValue.Order	Order取值与规范不符，请修正后再试。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。
InvalidParameterValue.Qualifier	Qualifier取值与规范不符，请修正后再试。
InvalidParameterValue.RuleId	RuleId取值与规范不符，请修正后再试。
InvalidParameterValue.RuleName	RuleName取值与规范不符，请修正后再试。
InvalidParameterValue.Tags	标签参数错误。
InvalidParameterValue.TargetDescription	TargetDescription取值与规范不符，请修正后再试。
InvalidParameterValue.TargetId	TargetId取值与规范不符，请修正后再试。
InvalidParameterValue.TransformationID	数据转换配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.Transformations	数据转换任务创建失败，配置参数取值与规范不符，请修正后再试。
InvalidParameterValue.Type	Type取值与规范不符，请修正后再试。
LimitExceeded.BannedAccount	事件投递失败，因欠费或违规等原因，账号已被禁用，请联系官网账户端客服处理。
LimitExceeded.ClusterPrivateLinkExceeded	集群私有链接超限。
LimitExceeded.Connection	Connection数量达到限制，可提交工单申请提升限制。
LimitExceeded.EventBus	EventBus数量达到限制，可提交工单申请提升限制。
LimitExceeded.InsufficientBalance	资源创建失败，可冻结余额不足，请充值后重新创建。
LimitExceeded.Logset	Logset数量达到限制，可提交工单申请提升限制。
LimitExceeded.ResourceLimit	ResourceLimit数量达到限制，可提交工单申请提升限制。
LimitExceeded.RouteOverLimit	RouteOverLimit数量达到限制，可提交工单申请提升限制。
LimitExceeded.Rule	Rule数量达到限制，可提交工单申请提升限制。
LimitExceeded.Target	Target数量达到限制，可提交工单申请提升限制。
LimitExceeded.Trigger	投递目标创建失败，函数触发器数量达到限制，可提交工单申请提升限制。
LimitExceeded.UserPrivateLinkExceeded	用户私有链接超限。
OperationDenied.AccountNotExists	投递目标创建失败，检测到当前账号不存在，请确认您的账号状态。
OperationDenied.DefaultCLSResourceUnsupported	不允许在自定义事件集上创建基于EB默认日志集的CLS事件目标。
OperationDenied.ESVersionUnsupported	不支持操作当前ES版本。
OperationDenied.EventBusResourceIsLocked	技术架构升级，该资源临时锁定中，预计持续3~5分钟，事件推送流程无影响。
OperationDenied.ResourceImmutable	操作失败，该资源不可修改或删除。
OperationDenied.UnsupportedOperation	当前用户账号类型暂不支持操作，请提交工单处理。
ResourceInUse.DefaultEventBus	同一个地域下，云服务默认事件集不允许重复创建。
ResourceInUse.EventBus	无法删除，因事件集下存在规则、连接器或归档，请删除所有规则、连接器和归档后重试。
ResourceInUse.Rule	规则下有目标，无法删除，请删除所有目标后重试。
ResourceNotFound.Connection	未找到指定的Connection，请创建后再试。
ResourceNotFound.EventBus	未找到指定事件集，请创建后再试。

错误码	说明
ResourceNotFound.EventBusNotFound	事件集不存在或未配置规则，请检查后再试。
ResourceNotFound.Function	未找到指定的Function，请创建后再试。
ResourceNotFound.InvalidApi	连接器删除失败，未找到指定 API 。
ResourceNotFound.InvalidService	连接器操作失败，未找到指定 API 网关服务。
ResourceNotFound.Namespace	投递目标创建失败，未找到指定的命名空间，请创建后再试。
ResourceNotFound.NetAssociation	未找到网络资源关联记录。
ResourceNotFound.PrivateLinkResource	未找到privatelink记录。
ResourceNotFound.Role	未找到指定的服务角色，请创建后再试。
ResourceNotFound.Rule	未找到指定的Rule，请创建后再试。
ResourceNotFound.Tag	未找到指定标签。
ResourceNotFound.Target	未找到指定的Target，请创建后再试。
ResourceNotFound.Transformation	未找到指定的转换任务，请创建后再试。
ResourceNotFound.Version	投递目标创建失败，未找到指定的服务版本，请创建后再试。
ResourceUnavailable.Connection	连接器创建失败，当前资源不可用。
ResourceUnavailable.ESUnhealth	ES集群状态异常。
ResourceUnavailable.Target	投递目标创建失败，当前资源不可用。
UnauthorizedOperation.CAM	当前账号缺少 EB 操作权限，请登录 CAM 控制台进行授权。
UnsupportedOperation.InvalidAction	连接器操作失败，接口不支持当前操作。
UnsupportedOperation.InvalidEndpointType	连接器创建失败，不支持当前后端服务类型。