人体分析

API 文档

产品文档

腾讯云

文档目录

API 文档
更新历史

最近更新时间：2021-01-22 08:00:43

第 6 次发布

发布时间：2021-01-22 08:00:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- SegmentCustomizedPortraitPic
  - 新增出参：ImageRects

新增数据结构：

- ImageRect

第 5 次发布

发布时间：2020-11-27 08:00:29

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- CreateSegmentationTask
- DescribeSegmentationTask
- TerminateSegmentationTask

新增数据结构：

- VideoBasicInformation

第 4 次发布

发布时间：2020-07-30 08:01:28

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- SegmentCustomizedPortraitPic

新增数据结构：

- SegmentationOptions

第 3 次发布

发布时间：2020-06-30 08:01:29

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- DetectBodyJoints

新增数据结构：

- BodyJointsResult
- BoundRect
- KeyPointInfo

第 2 次发布

发布时间：2020-06-19 08:01:05

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- DetectBody
  - 新增入参：AttributesOptions

新增数据结构：

- Age
- AttributesOptions
- Bag
- BodyAttributeInfo
- Gender
- LowerBodyCloth
- LowerBodyClothColor
- LowerBodyClothLength
- LowerBodyClothType
- Orientation
- UpperBodyCloth
- UpperBodyClothColor
- UpperBodyClothSleeve
- UpperBodyClothTexture

修改数据结构：

- BodyDetectResult
  - 新增成员：BodyAttributeInfo

第 1 次发布

发布时间：2020-03-30 08:31:45

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- CreateGroup

- CreatePerson
- CreateTrace
- DeleteGroup
- DeletePerson
- DetectBody
- GetGroupList
- GetPersonList
- ModifyGroup
- ModifyPersonInfo
- SearchTrace
- SegmentPortraitPic

新增数据结构：

- BodyDetectResult
- BodyRect
- Candidate
- GroupInfo
- PersonInfo
- Trace
- TraceInfo

简介

最近更新时间：2020-03-30 12:37:12

人体分析 API 升级到 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。

腾讯云神图 · 人体分析（Body Analysis）基于腾讯优图领先的人体分析算法，提供人像分割、人体检测、行人重识别（ReID）等服务。支持识别图片或视频中的半身人体轮廓，并将其与背景进行分离，支持通过人体检测，识别行人的穿着、体态、发型等属性信息，实现跨摄像头跨场景下行人的识别与检索。可应用于人像抠图、背景特效、行人搜索、人群密度检测等场景。

人像分割

指对图片或视频中的半身人体轮廓范围进行识别，将其与背景进行分离，返回分割后的二值图、灰度图、前景人像图。支持实现背景图像的替换与合成，支持为人像添加各种设定的背景特效，还可以对分割后的人像进行人脸特效处理。可应用于人像抠图、照片合成、人像特效等场景，极大提升工具效率。

人体检测

指在给定图片中检测出人体位置的矩形框，识别人体的头部、肩部、手部、脚部等多个关键点，并返回人体坐标信息。支持多人体等复杂场景的检测，可应用于互动娱乐中增加互动体验，还可用于智慧安防中预警异常行为，还能用于实现人群密度检测。

行人重识别

也叫人体 ReID。指在人体检测的基础上，通过识别行人的穿着、体态等属性信息，实现跨摄像头、跨场景下行人的识别与检索。配合人脸识别使用，还可以实现行人身份的确认。可应用于智慧零售、安防监控等场景，帮助提升检测效率。

API 概览

最近更新时间：2020-11-27 08:00:34

人体库管理相关接口

接口名称	接口功能
CreateGroup	创建人体库
CreatePerson	创建人员
CreateTrace	增加人员轨迹
DeleteGroup	删除人体库
DeletePerson	删除人员
GetGroupList	获取人体库列表
GetPersonList	获取人员列表
ModifyGroup	修改人体库
ModifyPersonInfo	修改人员信息

人体搜索相关接口

接口名称	接口功能
SearchTrace	人体搜索

人体检测相关接口

接口名称	接口功能
DetectBody	人体检测与属性分析
DetectBodyJoints	人体关键点分析

人像分割相关接口

接口名称	接口功能
CreateSegmentationTask	创建视频人像分割处理任务
DescribeSegmentationTask	查看视频人像分割处理任务信息
SegmentCustomizedPortraitPic	自定义人像分割
SegmentPortraitPic	人像分割
TerminateSegmentationTask	终止视频人像分割处理任务

调用方式
请求结构

最近更新时间：2020-09-28 08:00:40

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 bda.tencentcloudapi.com ，也支持指定地域域名访问，例如广州地域的域名为 bda.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 bda.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）	bda.tencentcloudapi.com
华南地区(广州)	bda.ap-guangzhou.tencentcloudapi.com
华东地区(上海)	bda.ap-shanghai.tencentcloudapi.com
华北地区(北京)	bda.ap-beijing.tencentcloudapi.com
西南地区(成都)	bda.ap-chengdu.tencentcloudapi.com
西南地区(重庆)	bda.ap-chongqing.tencentcloudapi.com
港澳台地区(中国香港)	bda.ap-hongkong.tencentcloudapi.com
亚太东南(新加坡)	bda.ap-singapore.tencentcloudapi.com
亚太东南(曼谷)	bda.ap-bangkok.tencentcloudapi.com
亚太南部(孟买)	bda.ap-mumbai.tencentcloudapi.com
亚太东北(首尔)	bda.ap-seoul.tencentcloudapi.com
亚太东北(东京)	bda.ap-tokyo.tencentcloudapi.com
美国东部(弗吉尼亚)	bda.na-ashburn.tencentcloudapi.com
美国西部(硅谷)	bda.na-siliconvalley.tencentcloudapi.com
北美地区(多伦多)	bda.na-toronto.tencentcloudapi.com
欧洲地区(法兰克福)	bda.eu-frankfurt.tencentcloudapi.com
欧洲地区(莫斯科)	bda.eu-moscow.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：2021-04-28 08:01:48

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的接口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。

签名方法 v3

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 文档。

注意：接口文档中的示例由于目的是展示接口参数用法，简化起见，使用的是签名方法 v1 GET 请求，如果依旧想使用签名方法 v1 请参考下文章节。

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：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-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。**
X-TC-Timestamp	Integer	是	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。**
X-TC-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。
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 bda； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

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--
```

签名方法 v1

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。

参数名称	类型	必选	描述
Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。**注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。**
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

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
Content-Type: application/x-www-form-urlencoded
```

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

地域	取值
西南地区(成都)	ap-chengdu
华南地区(广州)	ap-guangzhou
华东地区(南京)	ap-nanjing
华东地区(上海)	ap-shanghai

签名方法 v3

最近更新时间：2020-12-15 08:01:42

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，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 7 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++。

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请求参数中指定该签名结果并以指定的方式和格式发送请求。

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。如已泄漏，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】即可以创建一对密钥。

签名过程

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

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

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。

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, Sig
nedHeaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c" \
-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 中问号（?）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，字符集 UTF8，推荐使用编程语言标准库，所有特殊字符均需编码，大写形式。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。 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\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
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

content-type;host
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 5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031。

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，可能导致运行一段时间后，请求必定失败，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
5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031
```

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，此处不展示中间结果。

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 2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c 。

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。
Signature	签名值。此示例计算结果是 2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c。

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, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c
```

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-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c
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"}]}
```

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++

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。

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;
private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";
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";
String signedHeaders = "content-type;host";

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
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

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": [{"Name": "instance-name", "Values": [u"未命名"]}]}

# ************* 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\n" % (ct, host)
signed_headers = "content-type;host"
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"
"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() {
secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secretKey := "Gu5t9xGARNpq86cd98joQYCN3*******"
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 := "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n"
signedHeaders := "content-type;host"
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
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*******";
$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 = "";
$canonicalHeaders = "content-type:application/json; charset=utf-8\n"."host:".$host."\n";
$signedHeaders = "content-type;host";
$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."/".$credentialScope
.", SignedHeaders=content-type;host, Signature=".$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
secret_id = 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******'
secret_key = 'Gu5t9xGARNpq86cd98joQYCN3*******'

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}\n"
signed_headers = 'content-type;host'
# 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 *******, 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}, Signat
ure=#{signature}"
puts authorization

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

DotNet

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

public class Application
{
public static string SHA256Hex(string s)
{
using (SHA256 algo = SHA256.Create())
{
```

```
byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
StringBuilder builder = new StringBuilder();
for (int i = 0; i < hashbytes.Length; ++i)
{
builder.Append(hashbytes[i].ToString("x2"));
}
return builder.ToString();
}
}
public static byte[] HmacSHA256(byte[] key, byte[] msg)
{
using (HMACSHA256 mac = new HMACSHA256(key))
{
return mac.ComputeHash(msg);
}
}

public static Dictionary<String, String> BuildHeaders(string secretid,
string secretkey, string service, string endpoint, string region,
string action, string version, DateTime date, string requestPayload)
{
string datestr = date.ToString("yyyy-MM-dd");
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 10
00;
// ************* 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";
string signedHeaders = "content-type;host";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine("-----------------------------------");

// ************* 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" + hashedCanon
icalRequest;
Console.WriteLine(stringToSign);
Console.WriteLine("-----------------------------------");
```

```
// ************* 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);
Console.WriteLine("----------------------------------");

// ************* 步骤 4：拼接 Authorization *************
string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);
Console.WriteLine("----------------------------------");

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
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

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

// 此处由于示例规范的原因，采用时间戳2019-02-26 00:44:25，此参数作为示例，如果在项目中，您应当使用：
// DateTime date = DateTime.UtcNow;
// 注意时区，建议此时间统一采用UTC时间戳，否则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instan
ce-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
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******"

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 signedHeaders = "content-type;host"

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"

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

// ************* 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)
console.log("-----------------------------")

// ************* 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)
console.log("-----------------------------")

// ************* 步骤 4：拼接 Authorization *************
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)
console.log("-----------------------------")

const Call_Information = '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(Call_Information)
```

```
}
main()
```

C++

```
#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
```

```
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

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 canonicalHeaders = "content-type:application/json; charset=utf-8\nhost:" + host + "\n";
string signedHeaders = "content-type;host";
string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name
\"}]}";
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
cout << canonicalRequest << endl;
cout << "-----------------------" << 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" + hashedCanonicalReques
t;
cout << stringToSign << endl;
cout << "-----------------------" << 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;
cout << "-----------------------" << endl;

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

string headers = "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 << headers << endl;
return 0;
};
```

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：2020-08-11 08:01:51

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 7 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。

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

参数名称	中文	参数值
Version	接口版本号	2017-03-12

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" ，因此格式化后就为 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=AKI
Dz8krbsJ5yKBZQpn74WFkmLPx3*******&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。 签名原文字符串由以下几个参数构成:

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

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

示例的拼接结果为:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Regi
on=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 = 'Gu5t9xGARNpq86cd98joQYCN3*******';
$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

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

最终输出的 url 可能为：https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=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，即使是旧版的 API ，由于存在细节差异也会导致签名计算错误，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
params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"); // 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
params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3*******", "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 time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

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"
"sort"
)

func main() {
secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secretKey := "Gu5t9xGARNpq86cd98joQYCN3*******"
params := map[string]string{
"Nonce": "11886",
"Timestamp": "1465185768",
"Region": "ap-guangzhou",
"SecretId": secretId,
"Version": "2017-03-12",
"Action": "DescribeInstances",
"InstanceIds.0": "ins-09dx96dg",
"Limit": "20",
"Offset": "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
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*******";
$param["Nonce"] = 11886;//rand();
$param["Timestamp"] = 1465185768;//time();
$param["Region"] = "ap-guangzhou";
$param["SecretId"] = $secretId;
$param["Version"] = "2017-03-12";
$param["Action"] = "DescribeInstances";
$param["InstanceIds.0"] = "ins-09dx96dg";
$param["Limit"] = 20;
$param["Offset"] = 0;

ksort($param);

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

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
```

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

Ruby

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

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

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, strin
g 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
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

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

```
param.Add("Offset", "0");
param.Add("InstanceIds.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

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

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

NodeJS

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

function get_req_url(params, endpoint){
params['Signature'] = escape(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
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******"

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

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

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

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

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

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

返回结果

最近更新时间：2020-06-12 08:01:39

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，下表列出了公共错误码。

错误码	错误描述

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	HTTPS 请求方法错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

人体检测相关接口
人体检测与属性分析

最近更新时间：2021-03-05 08:01:41

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

检测给定图片中的人体（Body）的位置信息及属性信息。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer　点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DetectBody。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-nanjing, ap-shanghai
Image	否	String	人体图片 Base64 数据。 图片 base64 编码后大小不可超过5M。 图片分辨率不得超过 1920 * 1080 。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
MaxBodyNum	否	Integer	最多检测的人体数目，默认值为1（仅检测图片中面积最大的那个人体）； 最大值10，检测图片中面积最大的10个人体。
Url	否	String	人体图片 Url 。 Url、Image必须提供一个，如果都提供，只使用 Url。 图片 base64 编码后大小不可超过5M。 图片分辨率不得超过 1920 * 1080 。 图片存储于腾讯云的Url可保障更高下载速度和稳定性，建议图片存储于腾讯云。 非腾讯云存储的Url速度和稳定性可能受一定影响。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
AttributesOptions	否	AttributesOptions	是否返回年龄、性别、朝向等属性。 可选项有 Age、Bag、Gender、UpperBodyCloth、LowerBodyCloth、Orientation。 如果此参数为空则为不需要返回。 需要将属性组成一个用逗号分隔的字符串，属性之间的顺序没有要求。 关于各属性的详细描述，参见下文出参。 最多返回面积最大的 5 个人体属性信息，超过 5 个人体（第 6 个及以后的人体）的 BodyAttributesInfo 不具备参考意义。

3. 输出参数

参数名称	类型	描述
BodyDetectResults	Array of BodyDetectResult	图中检测出来的人体框。
BodyModelVersion	String	人体识别所用的算法模型版本。
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 调用失败示例

输入示例

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

{
"Url": "IamNotUrl"
}
```

输出示例

```
{
"Response": {
"RequestId": "b5f77c78-efaf-42d1-b0ac-419cc70b4994"
}
}
```

示例2 调用成功示例

输入示例

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

{
"Url": "test.jpg"
}
```

输出示例

```
{
"Response": {
"BodyDetectResults": [
```

```
{
"BodyRect": {
"X": 260,
"Y": 1,
"Width": 272,
"Height": 365
},
"Confidence": 0.91490805149078,
"BodyAttributeInfo": {
"UpperBodyCloth": {
"Color": {
"Type": "xx",
"Probability": 0.0
},
"Sleeve": {
"Type": "xx",
"Probability": 0.0
},
"Texture": {
"Type": "xx",
"Probability": 0.0
}
},
"Orientation": {
"Type": "xx",
"Probability": 0.0
},
"LowerBodyCloth": {
"Color": {
"Type": "xx",
"Probability": 0.0
},
"Length": {
"Type": "xx",
"Probability": 0.0
},
"Type": {
"Type": "xx",
"Probability": 0.0
}
},
"Gender": {
"Type": "xx",
"Probability": 0.0
},
"Age": {
"Type": "xx",
"Probability": 0.0
},
"Bag": {
"Type": "xx",
```

```
"Probability": 0.0
}
}
}
],
"BodyModelVersion": "1.0",
"RequestId": "13ce6864-614a-4a9f-8207-c68fc9c552c4"
}
}
```


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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.BodyQualityNotQualified	人体质量分过低。
FailedOperation.ImageBodyDetectFailed	人体检测失败。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageNotSupported	不支持的图片文件。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageSizeExceed	base64编码后的图片数据过大。

错误码	描述
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.NoBodyInPhoto	图片中没有人体。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.UnKnowError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.UrlIllegal	URL格式不合法。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

人体关键点分析

最近更新时间：2021-05-18 08:01:16

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

检测图片中人体的14个关键点。建议用于人体图像清晰、无遮挡的场景。支持一张图片中存在多个人体的识别。

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：DetectBodyJoints。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	否	String	公共参数，本接口不需要传递此参数。
Image	否	String	图片 base64 数据，base64 编码后大小不可超过5M。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
Url	否	String	图片的 Url 。对应图片 base64 编码后大小不可超过5M。 Url、Image必须提供一个，如果都提供，只使用 Url。 图片存储于腾讯云的Url可保障更高下载速度和稳定性，建议图片存储于腾讯云。 非腾讯云存储的Url速度和稳定性可能受一定影响。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。

3. 输出参数

参数名称	类型	描述
BodyJointsResults	Array of BodyJointsResult	图中检测出的人体框和人体关键点， 包含14个人体关键点的坐标，建议根据人体框置信度筛选出合格的人体；
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 调用失败示例

输入示例

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

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

{
"Url": "IamNotUrl"
}
```

输出示例

```
{
"Response": {
"RequestId": "b5f77c78-efaf-42d1-b0ac-419cc70b4994"
}
}
```

示例2 调用成功示例

输入示例

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

{
"Url": "test.jpg"
}
```

输出示例

```
{
"Response": {
"BodyJointsResults": [
{
"BoundBox": {
"X": 134,
"Y": 169,
"Width": 270,
"Height": 486
},
"BodyJoints": [
{
"X": 233.458984375,
"Y": 212.494140625,
"KeyPointType": "头部"
},
{
"X": 237.619140625,
"Y": 249.935546875,
```

```
"KeyPointType": "颈部"
},
{
"X": 283.380859375,
"Y": 266.576171875,
"KeyPointType": "右肩"
},
{
"X": 324.982421875,
"Y": 308.177734375,
"KeyPointType": "右肘"
},
{
"X": 295.861328125,
"Y": 349.779296875,
"KeyPointType": "右腕"
},
{
"X": 208.498046875,
"Y": 274.896484375,
"KeyPointType": "左肩"
},
{
"X": 183.537109375,
"Y": 337.298828125,
"KeyPointType": "左肘"
},
{
"X": 158.576171875,
"Y": 387.220703125,
"KeyPointType": "左腕"
},
{
"X": 262.580078125,
"Y": 395.541015625,
"KeyPointType": "右髋"
},
{
"X": 237.619140625,
"Y": 466.263671875,
"KeyPointType": "右膝"
},
{
"X": 216.818359375,
"Y": 557.787109375,
"KeyPointType": "右踝"
},
{
"X": 254.259765625,
"Y": 399.701171875,
"KeyPointType": "左髋"
```

```
},
{
"X": 266.740234375,
"Y": 482.904296875,
"KeyPointType": "左膝"
},
{
"X": 349.943359375,
"Y": 545.306640625,
"KeyPointType": "左踝"
}
],
"Confidence": 0.99999010562897
}
],
"RequestId": "97d85578-6b11-4e7c-beea-65601bc0bc04"
}
}
```


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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.BodyFeatureFail	人体特征检测失败。

错误码	描述
FailedOperation.BodyJointsFail	人体关键点检测失败。
FailedOperation.DownloadError	文件下载失败。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageSizeExceed	base64编码后的图片数据过大。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.NoBodyInPhoto	图片中没有人体。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.UnKnowError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.TooLargeFileError	文件太大。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

人体库管理相关接口
修改人员信息

最近更新时间：2021-03-05 08:01:42

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

修改人员信息。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：ModifyPersonInfo。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID。
PersonName	否	String	人员名称。

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 调用失败示例

输入示例

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

{
"PersonId": "12313123",
```

```
"PersonName": "testG10P1M1"
}
```

输出示例

```
{
"Response": {
"RequestId": "5844914d-b2e1-4afc-9970-8d1cdd6a7138"
}
}
```

示例2 调用成功示例

输入示例

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

{
"PersonId": "testG10P1",
"PersonName": "testG10P1M1"
}
```

输出示例

```
{
"Response": {
"RequestId": "327920d1-f111-463d-b9d3-3eaa0a473508"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP

- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.PersonIdIllegal	人员ID包含非法字符。人员ID只支持英文、数字、-%@#&_。
InvalidParameterValue.PersonIdNotExist	人员ID不存在。
InvalidParameterValue.PersonIdTooLong	人员ID超出长度限制。
InvalidParameterValue.PersonNameIllegal	人员名称包含非法字符。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
UnsupportedOperation.UnknowMethod	未知方法名。

删除人体库

最近更新时间：2021-03-05 08:01:43

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

删除该人体库及包含的所有的人员。

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：DeleteGroup。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人体库ID。

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 调用失败示例

输入示例

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

{
"GroupId": "1231231"
}
```

输出示例

```
{
"Response": {
"RequestId": "cb24cce3-beb9-4f8b-9f61-e79fb075a9a4"
}
}
```

示例2 调用成功示例

输入示例

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

{
"GroupId": "testG10"
}
```

输出示例

```
{
"Response": {
"RequestId": "bc9be78c-6d08-430e-9119-959d903769c9"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

创建人员

最近更新时间：2021-03-05 08:01:43

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

创建人员，添加对应人员的人体轨迹信息。

请注意：

- 我们希望您的输入为 严格符合轨迹图片 要求的图片。如果您输入的图片不符合轨迹图片要求，会对最终效果产生较大负面影响。请您尽量保证一个Trace中的图片人体清晰、无遮挡、连贯；
- 一个人体轨迹（Trace）可以包含1-5张人体图片。提供越多质量高的人体图片有助于提升最终识别结果；
- 无论您在单个Trace中提供了多少张人体图片，我们都将生成一个对应的轨迹（Trace）信息。即，Trace仅和本次输入的图片序列相关，和图片的个数无关；
- 输入的图片组中，若有部分图片输入不合法（如图片大小过大、分辨率过大、无法解码等），我们将舍弃这部分图片，确保合法图片被正确搜索。即，我们将尽可能保证请求成功，去除不合法的输入；
- 构成人体轨迹单张图片大小不得超过2M，分辨率不得超过1920*1080。

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：CreatePerson。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	待加入的人员库ID。
PersonName	是	String	人员名称。[1，60]个字符，可修改，可重复。
PersonId	是	String	人员ID，单个腾讯云账号下不可修改，不可重复。 支持英文、数字、-%@#&_，，长度限制64B。
Trace	是	Trace	人体轨迹信息。

3. 输出参数

参数名称	类型	描述
TraceId	String	人员轨迹唯一标识。
BodyModelVersion	String	人体识别所用的算法模型版本。

参数名称	类型	描述
InputRetCode	Integer	输入的人体轨迹图片中的合法性校验结果。 只有为0时结果才有意义。 -1001: 输入图片不合法。-1002: 输入图片不能构成轨迹。
InputRetCodeDetails	Array of Integer	输入的人体轨迹图片中的合法性校验结果详情。 -1101:图片无效，-1102:url不合法。-1103:图片过大。-1104:图片下载失败。-1105:图片解码失败。-1109:图片分辨率过高。-2023:轨迹中有非同人图片。-2024: 轨迹提取失败。-2025: 人体检测失败。 RetCode 的顺序和入参中Images 或 Urls 的顺序一致。
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 调用失败示例

输入示例

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

{
"GroupId": "testG3",
"PersonName": "testG3P1",
"PersonId": "testG3P1",
"Trace": {
"Urls": [
"IamNotUrl"
],
"BodyRects": [
{
"X": 1,
"Y": 2,
"Width": 3,
"Height": 4
}
]
}
}
```

输出示例

```
{
"Response": {
"TraceId": "",
"BodyModelVersion": "",
"InputRetCode": -1001,
"InputRetCodeDetails": [
```

```
-1102
],
"RequestId": "78b14df0-9ca6-45e0-b5d4-f053db01f9bb"
}
}
```

示例2 调用成功示例

输入示例

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

{
"GroupId": "testG3",
"PersonName": "testG3P1",
"PersonId": "testG3P1",
"Trace": {
"Urls": [
"http://i2.sinaimg.cn/ty/nba/2015-07-05/U10236P6T12D7648505F44DT20150705114547.jpg"
],
"BodyRects": [
{
"X": 1,
"Y": 2,
"Width": 3,
"Height": 4
}
]
}
}
```

输出示例

```
{
"Response": {
"TraceId": "testCreatePerson123",
"BodyModelVersion": "1.0",
"InputRetCode": 0,
"InputRetCodeDetails": [
0
],
"RequestId": "2d44289d-eeb1-4109-8c2e-3c3e514094b9"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.BodyRectIllegal	输入的人体框不合法。
FailedOperation.BodyRectNumIllegal	输入的人体框数量不合法。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountTraceNumExceed	账号人体轨迹数量超出限制。
InvalidParameterValue.BodyModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.BodyRectsExceed	传入的人体框过多。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
InvalidParameterValue.GroupTraceNumExceed	人体库人体轨迹数量超出限制。
InvalidParameterValue.ImageEmpty	图片为空。

错误码	描述
InvalidParameterValue.PersonIdAlreadyExist	人员ID已经存在。人员ID不可重复。
InvalidParameterValue.PersonIdIllegal	人员ID包含非法字符。人员ID只支持英文、数字、-%@#&_。
InvalidParameterValue.PersonIdTooLong	人员ID超出长度限制。
InvalidParameterValue.PersonNameIllegal	人员名称包含非法字符。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.TraceBodyNumExceed	创建人体轨迹的人体图片数量超出限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。

删除人员

最近更新时间：2021-03-05 08:01:43

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

删除人员。

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：DeletePerson。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID。

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 调用失败示例

输入示例

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

{
"PersonId": "123131231"
}
```

输出示例

```
{
"Response": {
"RequestId": "27304486-69f3-47fd-b8ce-b7436ec8486b"
}
}
```

示例2 调用成功示例

输入示例

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

{
"PersonId": "testG10P1"
}
```

输出示例

```
{
"Response": {
"RequestId": "10f9a902-184e-40d6-b09d-e85f0c2dcfba"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.PersonIdIllegal	人员ID包含非法字符。人员ID只支持英文、数字、-%@#&_。
InvalidParameterValue.PersonIdNotExist	人员ID不存在。
InvalidParameterValue.PersonIdTooLong	人员ID超出长度限制。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

创建人体库

最近更新时间：2021-03-05 08:01:43

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

用于创建一个空的人体库，如果人体库已存在返回错误。

1个APPID下最多有2000W个人体轨迹（Trace），最多1W个人体库（Group）。

单个人体库（Group）最多10W个人体轨迹（Trace）。

单个人员（Person）最多添加 5 个人体轨迹（Trace）。

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：CreateGroup。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupName	是	String	人体库名称，[1,60]个字符，可修改，不可重复。
GroupId	是	String	人体库 ID，不可修改，不可重复。支持英文、数字、-%@#&_，长度限制64B。
Tag	否	String	人体库信息备注，[0，40]个字符。
BodyModelVersion	否	String	人体识别所用的算法模型版本。 目前入参仅支持 “1.0” 1个输入。默认为"1.0"。 不同算法模型版本对应的人体识别算法不同，新版本的整体效果会优于旧版本，后续我们将推出更新版本。

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 调用失败示例

输入示例

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

{
"GroupName": "testG3",
"GroupId": "testG3",
"Tag": "TestG3T3",
"BodyModelVersion": "2.0"
}
```

输出示例

```
{
"Response": {
"RequestId": "e1d5929c-5d72-4b2a-b354-2767056a6929"
}
}
```

示例2 调用成功示例

输入示例

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

{
"GroupName": "testG3",
"GroupId": "testG3",
"Tag": "TestG3T3",
"BodyModelVersion": "1.0"
}
```

输出示例

```
{
"Response": {
"RequestId": "95d7ed2b-4f54-4952-96d0-995981981e37"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountTraceNumExceed	账号人体轨迹数量超出限制。
InvalidParameterValue.BodyModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupIdAlreadyExist	人体库ID已经存在。人体库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
InvalidParameterValue.GroupNameAlreadyExist	人体库名称已经存在。人体库名称不可重复。
InvalidParameterValue.GroupNameTooLong	人体库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人体库数量超出限制。
InvalidParameterValue.GroupTagTooLong	人体库备注超出长度限制。
InvalidParameterValue.GroupTraceNumExceed	人体库人体轨迹数量超出限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。

错误码	描述
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

获取人体库列表

最近更新时间：2021-03-05 08:01:42

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

获取人体库列表。

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：GetGroupList。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
Offset	否	Integer	起始序号，默认值为0。
Limit	否	Integer	返回数量，默认值为10，最大值为1000。

3. 输出参数

参数名称	类型	描述
GroupInfos	Array of GroupInfo	返回的人体库信息。
GroupNum	Integer	人体库总数量。
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 调用失败示例

输入示例

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

{
```

```
"Offset": -1,
"Limit": 10
}
```

输出示例

```
{
"Response": {
"RequestId": "78ec7830-a1c7-42be-bd65-6d0fa25f6fab"
}
}
```

示例2 调用成功示例

输入示例

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

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

输出示例

```
{
"Response": {
"GroupNum": 1,
"GroupInfos": [
{
"GroupName": "testG2",
"GroupId": "testG2",
"Tag": "testG2Tag",
"BodyModelVersion": "1.0",
"CreationTimestamp": 1581673977535
}
],
"RequestId": "9ffccff2-4b52-443f-98f4-eb1f6a30399e"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.LimitExceed	返回数量不在合法范围内。
InvalidParameterValue.OffsetExceed	起始序号过大。请检查需要请求的数组长度。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

获取人员列表

最近更新时间：2021-03-05 08:01:42

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

获取指定人体库中的人员列表。

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：GetPersonList。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人体库ID。
Offset	否	Integer	起始序号，默认值为0。
Limit	否	Integer	返回数量，默认值为10，最大值为1000。

3. 输出参数

参数名称	类型	描述
PersonInfos	Array of PersonInfo	返回的人员信息。
PersonNum	Integer	该人体库的人员数量。
BodyModelVersion	String	人体识别所用的算法模型版本。
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 调用失败示例

输入示例

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

```
<公共请求参数>

{
"GroupId": "testG10",
"Offset": 0,
"Limit": 1001
}
```

输出示例

```
{
"Response": {
"RequestId": "c803e3eb-25b4-48d9-b4df-b74c570d110b"
}
}
```

示例2 调用成功示例

输入示例

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

{
"GroupId": "testG10",
"Offset": 0,
"Limit": 10
}
```

输出示例

```
{
"Response": {
"PersonNum": 1,
"BodyModelVersion": "1.0",
"PersonInfos": [
{
"PersonName": "testG10P1",
"PersonId": "testG10P1",
"TraceInfos": [
{
"TraceId": "3524775577730961229",
"BodyIds": [
"3524775577730961229-0"
]
}
]
```

```
}
],
"RequestId": "2e1841c2-81ee-42db-98ac-6c056aafe9b3"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
InvalidParameterValue.LimitExceed	返回数量不在合法范围内。
InvalidParameterValue.OffsetExceed	起始序号过大。请检查需要请求的数组长度。

错误码	描述
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

修改人体库

最近更新时间：2021-03-05 08:01:42

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

修改人体库名称、备注。

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：ModifyGroup。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人体库ID。
GroupName	否	String	人体库名称。
Tag	否	String	人体库信息备注。

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 调用失败示例

输入示例

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

{
"GroupId": "12312312",
```

```
"GroupName": "testG3M3",
"Tag": "testG3M3"
}
```

输出示例

```
{
"Response": {
"RequestId": "3661682c-414a-48d0-86fd-939d504bab70"
}
}
```

示例2 调用成功示例

输入示例

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

{
"GroupId": "testG3",
"GroupName": "testG3M3",
"Tag": "testG3M3"
}
```

输出示例

```
{
"Response": {
"RequestId": "32ac4a6a-7a58-4968-b32e-b8d7772eb26e"
}
}
```

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

- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
InvalidParameterValue.GroupNameAlreadyExist	人体库名称已经存在。人体库名称不可重复。
InvalidParameterValue.GroupNameTooLong	人体库名称超出长度限制。
InvalidParameterValue.GroupTagTooLong	人体库备注超出长度限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。
UnsupportedOperation.UnknowMethod	未知方法名。

增加人员轨迹

最近更新时间：2021-03-05 08:01:43

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

将一个人体轨迹添加到一个人员中。一个人员最多允许包含 5 个人体轨迹。同一人的人体轨迹越多，搜索识别效果越好。

请注意：

- 我们希望您的输入为 严格符合轨迹图片 要求的图片。如果您输入的图片不符合轨迹图片要求，会对最终效果产生较大负面影响。请您尽量保证一个Trace中的图片人体清晰、无遮挡、连贯。
- 一个人体轨迹（Trace）可以包含1-5张人体图片。提供越多质量高的人体图片有助于提升最终识别结果。
- 无论您在单个Trace中提供了多少张人体图片，我们都将生成一个对应的轨迹（Trace）信息。即，Trace仅和本次输入的图片序列相关，和图片的个数无关。
- 输入的图片组中，若有部分图片输入不合法（如图片大小过大、分辨率过大、无法解码等），我们将舍弃这部分图片，确保合法图片被正确搜索。即，我们将尽可能保证请求成功，去除不合法的输入；
- 构成人体轨迹单张图片大小限制为2M，分辨率限制为1920*1080。

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：CreateTrace。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID。
Trace	是	Trace	人体轨迹信息。

3. 输出参数

参数名称	类型	描述
TraceId	String	人员轨迹唯一标识。
BodyModelVersion	String	人体识别所用的算法模型版本。

参数名称	类型	描述
InputRetCode	Integer	输入的人体轨迹图片中的合法性校验结果。 只有为0时结果才有意义。 -1001: 输入图片不合法。-1002: 输入图片不能构成轨迹。
InputRetCodeDetails	Array of Integer	输入的人体轨迹图片中的合法性校验结果详情。 -1101:图片无效，-1102:url不合法。-1103:图片过大。-1104:图片下载失败。-1105:图片解码失败。-1109:图片分辨率过高。-2023:轨迹中有非同人图片。-2024: 轨迹提取失败。-2025: 人体检测失败。
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 调用失败示例

输入示例

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

{
"PersonId": "3333333333333",
"Trace": {
"Urls": [
"http://i2.sinaimg.cn/ty/nba/2015-07-05/U10236P6T12D7648505F44DT20150705114547.jpg"
],
"BodyRects": [
{
"X": 1,
"Y": 2,
"Width": 3,
"Height": 4
}
]
}
}
```

输出示例

```
{
"Response": {
"RequestId": "81ee3a85-1273-4d39-b7b2-2dea15859039"
}
}
```

示例2 调用成功示例

输入示例

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

{
"PersonId": "testG3P1",
"Trace": {
"Urls": [
"http://i2.sinaimg.cn/ty/nba/2015-07-05/U10236P6T12D7648505F44DT20150705114547.jpg"
],
"BodyRects": [
{
"X": 1,
"Y": 2,
"Width": 3,
"Height": 4
}
]
}
}
```

输出示例

```
{
"Response": {
"TraceId": "testTrace123",
"BodyModelVersion": "1.0",
"InputRetCode": 0,
"InputRetCodeDetails": [
0
],
"RequestId": "d86cada6-1b1a-4ac1-86a4-c1a6bbb3e54c"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.BodyRectIllegal	输入的人体框不合法。
FailedOperation.BodyRectNumIllegal	输入的人体框数量不合法。
FailedOperation.CreateTraceFailed	创建轨迹失败，请选择符合要求的人体图片。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountTraceNumExceed	账号人体轨迹数量超出限制。
InvalidParameterValue.BodyRectsExceed	传入的人体框过多。
InvalidParameterValue.GroupTraceNumExceed	人体库人体轨迹数量超出限制。
InvalidParameterValue.PersonIdIllegal	人员ID包含非法字符。人员ID只支持英文、数字、-%@#&_。
InvalidParameterValue.PersonIdNotExist	人员ID不存在。
InvalidParameterValue.PersonIdTooLong	人员ID超出长度限制。
InvalidParameterValue.PersonTraceNumExceed	人员人体轨迹数量超出限制。
InvalidParameterValue.TraceBodyNumExceed	创建人体轨迹的人体图片数量超出限制。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

人体搜索相关接口
人体搜索

最近更新时间：2021-03-05 08:01:42

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

本接口用于对一组待识别的人体轨迹（Trace）图片，在人体库中识别出最相似的 TopK 人体，按照相似度从大到小排列。

人体轨迹（Trace）图片要求：图片中当且仅包含一个人体。人体完整、无遮挡。

请注意：

- 我们希望您的输入为严格符合轨迹图片要求的图片。如果您输入的图片不符合轨迹图片要求，会对最终效果产生较大负面影响；
- 人体轨迹，是一个包含1-5张图片的图片序列。您可以输入1张图片作为轨迹，也可以输入多张。单个轨迹中包含越多符合质量的图片，搜索效果越好。
- 构成人体轨迹单张图片大小不得超过2M，分辨率不得超过1920*1080。

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：SearchTrace。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	希望搜索的人体库ID。
Trace	是	Trace	人体轨迹信息。
MaxPersonNum	否	Integer	单张被识别的人体轨迹返回的最相似人员数量。 默认值为5，最大值为100。 例，设MaxPersonNum为8，则返回Top8相似的人员信息。值越大，需要处理的时间越长。建议不要超过10。
TraceMatchThreshold	否	Float	出参Score中，只有超过TraceMatchThreshold值的结果才会返回。 默认为0。范围[0, 100.0]。

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Candidates	Array of Candidate	识别出的最相似候选人。
InputRetCode	Integer	输入的人体轨迹图片中的合法性校验结果。 只有为0时结果才有意义。 -1001: 输入图片不合法。-1002: 输入图片不能构成轨迹。
InputRetCodeDetails	Array of Integer	输入的人体轨迹图片中的合法性校验结果详情。 -1101:图片无效，-1102:url不合法。-1103:图片过大。-1104:图片下载失败。-1105:图片解码失败。-1109:图片分辨率过高。-2023:轨迹中有非同人图片。-2024: 轨迹提取失败。-2025: 人体检测失败。
BodyModelVersion	String	人体识别所用的算法模型版本。
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 调用失败示例

输入示例

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

{
"GroupId": "12311111111",
"Trace": {
"Urls": [
"http://i2.sinaimg.cn/ty/nba/2015-07-05/U10236P6T12D7648505F44DT20150705114547.jpg"
],
"BodyRects": [
{
"X": 260,
"Y": 1,
"Width": 272,
"Height": 365
}
]
},
"MaxPersonNum": 2
}
```

输出示例

```
{
"Response": {
"RequestId": "1a5de4e8-02a0-4b4e-8c9a-e66e6f26d12a"
```

```
}
}
```

示例2 调用成功示例

输入示例

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

{
"GroupId": "12311111111",
"Trace": {
"Urls": [
"http://i2.sinaimg.cn/ty/nba/2015-07-05/U10236P6T12D7648505F44DT20150705114547.jpg"
],
"BodyRects": [
{
"X": 260,
"Y": 1,
"Width": 272,
"Height": 365
}
]
},
"MaxPersonNum": 2
}
```

输出示例

```
{
"Response": {
"Candidates": [
{
"PersonId": "testG10P1",
"TraceId": "3517452339320503246",
"Score": 99.99
},
{
"PersonId": "testG10P40",
"TraceId": "3516549744519331668",
"Score": 99.99
}
],
"BodyModelVersion": "1.0",
"InputRetCode": 0,
"InputRetCodeDetails": [
```

```
0
],
"RequestId": "0301e30d-e83c-43c7-b78c-29eafea4762d"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.BodyRectIllegal	输入的人体框不合法。
FailedOperation.BodyRectNumIllegal	输入的人体框数量不合法。
FailedOperation.GroupEmpty	搜索的人体库为空。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageSizeExceed	base64编码后的图片数据过大。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
InvalidParameter.InvalidParameter	参数不合法。

错误码	描述
InvalidParameterValue.BodyRectsExceed	传入的人体框过多。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.TraceBodyNumExceed	创建人体轨迹的人体图片数量超出限制。
InvalidParameterValue.TraceMatchThresholdIllegal	TraceMatchThreshold参数不合法。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

人像分割相关接口
创建视频人像分割处理任务

最近更新时间：2021-03-05 08:01:41

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

本接口为离线人像分割处理接口组中的提交任务接口，可以对提交的资源进行处理视频流/图片流识别视频作品中的人像区域，进行一键抠像、背景替换、人像虚化等后期处理。

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：CreateSegmentationTask。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表，本接口仅支持其中的: ap-guangzhou
VideoUrl	是	String	需要分割的视频URL，可外网访问。
BackgroundImageUrl	否	String	背景图片URL。 可以将视频背景替换为输入的图片。 如果不输入背景图片，则输出人像区域mask。
Config	否	String	预留字段，后期用于展示更多识别信息。

3. 输出参数

参数名称	类型	描述
TaskId	String	任务标识ID,可以用与追溯任务状态，查看任务结果
EstimatedProcessingTime	Float	预估处理时间，单位为秒
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 创建任务成功

创建视频人像分割处理任务成功

输入示例

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

{
"VideoUrl": "test.video.url",
"BackgroundImageUrl": "test.image.url"
}
```

输出示例

```
{
"Response": {
"RequestId": "33c35bff-27b6-408f-9ca7-19191303fa07",
"TaskID": "fakeTaskID",
"EstimatedProcessingTime": 30
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.DownloadError	文件下载失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.JobQueueFull	任务队列已满。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.TaskLimitExceeded	任务超过上限。
FailedOperation.TooLargeFileError	文件太大。
FailedOperation.UnKnowError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。

查看视频人像分割处理任务信息

最近更新时间：2021-03-05 08:01:41

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

可以查看单条任务的处理情况，包括处理状态，处理结果。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeSegmentationTask。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表，本接口仅支持其中的: ap-guangzhou
TaskID	是	String	在提交分割任务成功时返回的任务标识ID。

3. 输出参数

参数名称	类型	描述
TaskStatus	String	当前任务状态： QUEUING 排队中 PROCESSING 处理中 FINISHED 处理完成
ResultVideoUrl	String	分割后视频URL, 存储于腾讯云COS 注意：此字段可能返回 null，表示取不到有效值。
ResultVideoMD5	String	分割后视频MD5，用于校验 注意：此字段可能返回 null，表示取不到有效值。
VideoBasicInformation	VideoBasicInformation	视频基本信息 注意：此字段可能返回 null，表示取不到有效值。
ErrorMsg	String	分割任务错误信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，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 查询任务成功

查询视频人像分割处理任务成功

输入示例

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

{
"TaskID": "taskID"
}
```

输出示例

```
{
"Response": {
"RequestId": "0352ed67-66b0-4515-a04f-ddc0ab129658",
"TaskStatus": "FINISHED",
"ErrorMsg": "",
"ResultVideoUrl": "http://resulturl.com/a.mp4",
"ResultVideoMD5": "somemd5",
"VideoBasicInformation": {
"FrameWidth": 1280,
"FrameHeight": 590,
"FramesPerSecond": 28,
"Duration": 21,
"TotalFrames": 630
}
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS

- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.AudioDecodeFailed	音频解码失败。
FailedOperation.AudioEncodeFailed	音频编码失败。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.TaskNotExist	任务不存在。
FailedOperation.VideoDecodeFailed	视频解码失败。
InvalidParameter.InvalidParameter	参数不合法。

终止视频人像分割处理任务

最近更新时间：2021-03-05 08:01:40

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

终止指定视频人像分割处理任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：TerminateSegmentationTask。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表，本接口仅支持其中的: ap-guangzhou
TaskID	是	String	在提交分割任务成功时返回的任务标识ID。

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 取消任务成功

取消任务成功

输入示例

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

{
"TaskID": "fakeID"
}
```

输出示例

```
{
"Response": {
"RequestId": "0352ed67-66b0-4515-a04f-ddc0ab129658"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.RequestTimeout	后端服务超时。
FailedOperation.TaskNotExist	任务不存在。
FailedOperation.TerminateTaskFailed	任务取消失败。
InvalidParameter.InvalidParameter	参数不合法。

自定义人像分割

最近更新时间：2021-03-16 08:01:00

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

在前后景分割的基础上优化多分类分割，支持对头发、五官等的分割，既作为换发型、挂件等底层技术，也可用于抠人头、抠人脸等玩法

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：SegmentCustomizedPortraitPic。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-shanghai
SegmentationOptions	是	SegmentationOptions	此参数为分割选项，请根据需要选择自己所想从图片中分割的部分。注意所有选项均为非必选，如未选择则值默认为false，但是必须要保证多于一个选项的描述为true。
Image	否	String	图片 base64 数据，base64 编码后大小不可超过5M。 图片分辨率须小于2000*2000。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
Url	否	String	图片的 Url 。 Url、Image必须提供一个，如果都提供，只使用 Url。 图片分辨率须小于2000*2000 ，图片 base64 编码后大小不可超过 5M。 图片存储于腾讯云的Url可保障更高下载速度和稳定性，建议图片存储于腾讯云。 非腾讯云存储的Url速度和稳定性可能受一定影响。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。

3. 输出参数

参数名称	类型	描述
PortraitImage	String	根据指定标签分割输出的透明背景人像图片的 base64 数据。
MaskImage	String	指定标签处理后的Mask。一个通过 Base64 编码的文件，解码后文件由 Float 型浮点数组成。这些浮点数代表原图从左上角开始的每一行的每一个像素点，每一个浮点数的值是原图相应像素点位于人体轮廓内的置信度（0-1）转化的灰度值（0-255）

参数名称	类型	描述
ImageRects	Array of ImageRect	坐标信息。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，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 调用失败示例

输入示例

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

{
"Url": "IamNotAUrl",
"SegmentationOptions": {
"Background": false
}
}
```

输出示例

```
{
"Response": {
"RequestId": "46ed6f32-549f-4377-8116-2d55e4574528"
}
}
```

示例2 调用成功示例

输入示例

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

{
"Url": "test.jpg",
"SegmentationOptions": {
"Head": true
}
}
```

输出示例


```
{
"Response": {
"PortraitImage": "iVBORw0KGgoAAAANSUhEUgAAAoAAAAHJCAYAAAAVcogaAAAgAElEQVR4AdThgZZlRRYYybpH3ELr/f/fNnm2D5FMVCcMXJVUdOvJzIXxGxUVlUNFZa3FoXKpXPqNd2aGo6LiqKiA4YXAoKLycqGicizkUpHfqRyuxVfJHzzEkVBxVFx7+K5ihIpriGOAiqPiGuKqOCquiqviqPgrFUfFVfHfonxS/pLKoXKttXin4h15T+VSOVQuFZW1Fmst1lqowHDsvThUVHBQUTmW37gqjoprZnhv8bNUDpWj4qhQOSoqZoaKimOKY2aomBkqKipUroqrAgcViL+j8vekEVj871J5R+VnVPwMlaOi4lBRgeHb64XGWou9N3tv1lqstVDZe6Oy92bvzVqLtRYqxyTHzDAzPM/D8zw8z0PF8zwcM0PFzFAxMxwzUHFUHCqHispVcVT8lYqr4qioqKj4aiE8IL9TOVQulb+TwzjkcFRcFcfMcFUcFd/14p2KQ+UrlWO/fkFFRWWthcpai6NCRUVF5auRT478TiqM38TMgB8wgR8c8cGK3wzw4lhrsffG...",
"MaskImage": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/wAALCAHJAoABAREA/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx...",
"ImageRects": [
{
"Y": 1,
"X": 2,
"Height": 3,
"Width": 4,
"Label": "Head"
}
],
"RequestId": "07d71af0-d3af-4d36-8217-1e6836f38007"
}
}
```


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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageFacedetectFailed	人脸检测失败。
FailedOperation.ImageNotSupported	不支持的图片文件。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageSizeExceed	base64编码后的图片数据过大。
FailedOperation.ProfileNumExceed	人像数过多。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SegmentFailed	人像分割失败。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

人像分割

最近更新时间：2021-03-05 08:01:40

1. 接口描述

接口请求域名： bda.tencentcloudapi.com 。

即二分类人像分割，识别传入图片中人体的完整轮廓，进行抠像。

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：SegmentPortraitPic。
Version	是	String	公共参数，本接口取值：2020-03-24。
Region	是	String	公共参数，详见产品支持的 地域列表，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-shanghai
Image	否	String	图片 base64 数据，base64 编码后大小不可超过5M。 图片分辨率须小于2000*2000。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
Url	否	String	图片的 Url 。 Url、Image必须提供一个，如果都提供，只使用 Url。 图片分辨率须小于2000*2000 ，图片 base64 编码后大小不可超过5M。 图片存储于腾讯云的Url可保障更高下载速度和稳定性，建议图片存储于腾讯云。 非腾讯云存储的Url速度和稳定性可能受一定影响。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。

3. 输出参数

参数名称	类型	描述
ResultImage	String	处理后的图片 base64 数据，透明背景图
ResultMask	String	一个通过 Base64 编码的文件，解码后文件由 Float 型浮点数组成。这些浮点数代表原图从左上角开始的每一行的每一个像素点，每一个浮点数的值是原图相应像素点位于人体轮廓内的置信度（0-1）转化的灰度值（0-255）
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 调用失败示例

输入示例

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

{
"Url": "IamNotAUrl"
}
```

输出示例

```
{
"Response": {
"RequestId": "46ed6f32-549f-4377-8116-2d55e4574528"
}
}
```

示例2 调用成功示例

输入示例

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

{
"Url": "test.jpg"
}
```

输出示例


```
{
"Response": {
"ResultImage": "iVBORw0KGgoAAAANSUhEUgAAAiYAAAFuCAYAAAC1LdPaAAAgAElEQVR4AbTB0ZJlyXUYy4g8DT1hICNl5P9/50WdFdJqWIJ7io
WahnDl7r/99me+MzM8VTwNh39FxWcqS+Wcw3cqLpUnlUtFZamorMPvqaiorJlBReWcg4rKUuk9/DMqKirWaVCpWBUVFRVXRcWTysi3KpbKUlkq68Tf
qaioLJWZQWWpqCyVz1SeVN7vN/+IgA4UIaui4vp4x1UxHlbFer//ylWxKr5SsSpWRULFk8pSUXlS+YrKUlkqS0VFZVUsFZWOLBUVFRWVpaKiUlGxVF
R+eKioqFBRUVGZGb6jsSqeKtbM8KTyJN8757BUVkVFxTrn8KTyVKHypH...",
"ResultMask": "...XWPBHiy5s9MsfGWqRxIlxe+DdQbyLK9a6lDSLpaFbtC21I3GK/okoooooooooooooooooooooooor+Qb/g73/bN1j4HfsbeC
P2evCt7qum6x8e9cuDrN5YnT3spfDWi8T6bfrOHvImnnbfHJbKmQMM/av8AMAJJJJOSeST1J9TRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RXV+BvHPi/wCGni3QPHfgPxDqfhXxf4X1K11jQPEGjXUlnqWl6lZyrNbXdrcREPHLFIoYHocYIIJFf7FX/BAn9qf41fthf8E2fgz8Y/j74oXxl8RdR
ivtL1LxE1lBZXOow6XcPaW096tviOa8aKNTPcbVaaTLsMk1+0dFFFFFFFFFFFFFFFFFFFFFFFf5jX/B5jqOoSft8fCfTZL68fToPg3p88Fg9zO1lDP
JqFwJJorUuYI5ZAAHkSMOwADE4r+Oyiiiiv/Z",
"RequestId": "9cf173a5-4ae5-46fb-9898-6c876263780d"
}
}
```

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
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++
- Tencent Cloud SDK 3.0 for Ruby

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.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageNotSupported	不支持的图片文件。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionInsufficient	图片分辨率过小。
FailedOperation.ImageSizeExceed	base64编码后的图片数据过大。
FailedOperation.ProfileNumExceed	人像数过多。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SegmentFailed	人像分割失败。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.ImageEmpty	图片为空。

错误码	描述
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.TooLargeFileError	文件太大。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。

数据结构

最近更新时间：2021-01-22 08:00:42

Age

人体年龄信息。AttributesType 不含 Age 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	人体年龄信息，返回值为以下集合中的一个{小孩,青年,中年,老年}。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

AttributesOptions

返回人体属性选项，此值不填则为不需要返回，可以选择的值为以下六个。 Age、Bag、Gender、Orientation、UpperBodyCloth、LowerBodyCloth，详细的解释请看对象描述 需注意本接口最多返回面积最大的 5 个人体属性信息，超过 5 个人体（第 6 个及以后的人体）的人体属性不具备参考意义。

被如下接口引用：DetectBody。

名称	类型	必选	描述
Age	Boolean	否	返回年龄信息
Bag	Boolean	否	返回随身挎包信息
Gender	Boolean	否	返回性别信息
Orientation	Boolean	否	返回朝向信息
UpperBodyCloth	Boolean	否	返回上装信息
LowerBodyCloth	Boolean	否	返回下装信息

Bag

人体是否挎包。AttributesType 不含 Bag 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	挎包信息，返回值为以下集合中的一个{双肩包, 斜挎包, 手拎包, 无包}。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

BodyAttributeInfo

图中检测出的人体属性信息。

被如下接口引用：DetectBody。

名称	类型	描述

名称	类型	描述
Age	Age	人体年龄信息。 AttributesType 不含 Age 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。 注意：此字段可能返回 null，表示取不到有效值。
Bag	Bag	人体是否挎包。 AttributesType 不含 Bag 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。 注意：此字段可能返回 null，表示取不到有效值。
Gender	Gender	人体性别信息。 AttributesType 不含 Gender 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。 注意：此字段可能返回 null，表示取不到有效值。
Orientation	Orientation	人体朝向信息。 AttributesType 不含 UpperBodyCloth 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。 注意：此字段可能返回 null，表示取不到有效值。
UpperBodyCloth	UpperBodyCloth	人体上衣属性信息。 AttributesType 不含 Orientation 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。 注意：此字段可能返回 null，表示取不到有效值。
LowerBodyCloth	LowerBodyCloth	人体下衣属性信息。 AttributesType 不含 LowerBodyCloth 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。 注意：此字段可能返回 null，表示取不到有效值。

BodyDetectResult

图中检测出来的人体框。

被如下接口引用：DetectBody。

名称	类型	描述
Confidence	Float	检测出的人体置信度。 误识率百分之十对应的阈值是0.14；误识率百分之五对应的阈值是0.32；误识率百分之二对应的阈值是0.62；误识率百分之一对应的阈值是0.81。 通常情况建议使用阈值0.32，可适用大多数情况。
BodyRect	BodyRect	图中检测出来的人体框
BodyAttributeInfo	BodyAttributeInfo	图中检测出的人体属性信息。

BodyJointsResult

人体框和人体关键点信息。

被如下接口引用：DetectBodyJoints。

名称	类型	描述
BoundBox	BoundRect	图中检测出来的人体框。
BodyJoints	Array of KeyPointInfo	14个人体关键点的坐标，人体关键点详见KeyPointInfo。
Confidence	Float	检测出的人体置信度，0-1之间，数值越高越准确。

BodyRect

人体框

被如下接口引用：CreatePerson, CreateTrace, DetectBody, SearchTrace。

名称	类型	必选	描述
X	Integer	是	人体框左上角横坐标。
Y	Integer	是	人体框左上角纵坐标。
Width	Integer	是	人体宽度。
Height	Integer	是	人体高度。

BoundRect

人体框

被如下接口引用：DetectBodyJoints。

名称	类型	描述
X	Integer	人体框左上角横坐标。
Y	Integer	人体框左上角纵坐标。
Width	Integer	人体宽度。
Height	Integer	人体高度。

Candidate

识别出的最相似候选人。

被如下接口引用：SearchTrace。

名称	类型	描述
PersonId	String	人员ID。
TraceId	String	人体轨迹ID。
Score	Float	候选者的匹配得分。 十万人体库下，误识率百分之五对应的分数为70分；误识率百分之二对应的分数为80分；误识率百分之一对应的分数为90分。 二十万人体库下，误识率百分之五对应的分数为80分；误识率百分之二对应的分数为90分；误识率百分之一对应的分数为95分。 通常情况建议使用分数80分（保召回）。若希望获得较高精度，建议使用分数90分（保准确）。

Gender

人体性别信息。 AttributesType 不含 Gender 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。

被如下接口引用：DetectBody。

名称	类型	描述

名称	类型	描述
Type	String	性别信息，返回值为以下集合中的一个 {男性, 女性}
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

GroupInfo

返回的人员库信息。

被如下接口引用：GetGroupList。

名称	类型	描述
GroupName	String	人体库名称。
GroupId	String	人体库ID。
Tag	String	人体库信息备注。
BodyModelVersion	String	人体识别所用的算法模型版本。
CreationTimestamp	Integer	Group的创建时间和日期 CreationTimestamp。CreationTimestamp 的值是自 Unix 纪元时间到Group创建时间的毫秒数。 Unix 纪元时间是 1970 年 1 月 1 日星期四，协调世界时 (UTC) 。

ImageRect

图像坐标信息。

被如下接口引用：SegmentCustomizedPortraitPic。

名称	类型	描述
X	Integer	左上角横坐标。
Y	Integer	左上角纵坐标。
Width	Integer	人体宽度。
Height	Integer	人体高度。
Label	String	分割选项名称。

KeyPointInfo

人体关键点信息

被如下接口引用：DetectBodyJoints。

名称	类型	描述
KeyPointType	String	代表不同位置的人体关键点信息，返回值为以下集合中的一个 [头部,颈部,右肩,右肘,右腕,左肩,左肘,左腕,右髋,右膝,右踝,左髋,左膝,左踝]
X	Float	人体关键点横坐标
Y	Float	人体关键点纵坐标

LowerBodyCloth

下衣属性信息

被如下接口引用：DetectBody。

名称	类型	描述
Color	LowerBodyClothColor	下衣颜色信息。
Length	LowerBodyClothLength	下衣长度信息 。
Type	LowerBodyClothType	下衣类型信息。

LowerBodyClothColor

下衣颜色信息

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	下衣颜色信息，返回值为以下集合中的一个{ 黑色系, 灰白色系, 彩色} 。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

LowerBodyClothLength

下衣长度信息

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	下衣长度信息，返回值为以下集合中的一个，{长, 短} 。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

LowerBodyClothType

下衣类型信息

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	下衣类型，返回值为以下集合中的一个 {裤子,裙子} 。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

Orientation

人体朝向信息。
AttributesType 不含 Orientation 或检测超过 5 个人体时，此参数仍返回，但不具备参考意义。

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	人体朝向信息，返回值为以下集合中的一个 {正向, 背向, 左, 右}。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

PersonInfo

人员信息。

被如下接口引用：GetPersonList。

名称	类型	描述
PersonName	String	人员名称。
PersonId	String	人员ID。
TraceInfos	Array of TraceInfo	包含的人体轨迹图片信息列表。

SegmentationOptions

此参数为分割选项，请根据需要选择自己所想从图片中分割的部分。注意所有选项均为非必选，如未选择则值默认为false，但是必须要保证多于一个选项的描述为true。

被如下接口引用：SegmentCustomizedPortraitPic。

名称	类型	必选	描述
Background	Boolean	否	分割选项-背景
Hair	Boolean	否	分割选项-头发
LeftEyebrow	Boolean	否	分割选项-左眉
RightEyebrow	Boolean	否	分割选项-右眉
LeftEye	Boolean	否	分割选项-左眼
RightEye	Boolean	否	分割选项-右眼
Nose	Boolean	否	分割选项-鼻子
UpperLip	Boolean	否	分割选项-上唇
LowerLip	Boolean	否	分割选项-下唇
Tooth	Boolean	否	分割选项-牙齿
Mouth	Boolean	否	分割选项-口腔（不包含牙齿）
LeftEar	Boolean	否	分割选项-左耳
RightEar	Boolean	否	分割选项-右耳
Face	Boolean	否	分割选项-面部(不包含眼、耳、口、鼻等五官及头发。)
Head	Boolean	否	复合分割选项-头部(包含所有的头部元素，相关装饰除外)
Body	Boolean	否	分割选项-身体（包含脖子）
Hat	Boolean	否	分割选项-帽子
Headdress	Boolean	否	分割选项-头饰
Earrings	Boolean	否	分割选项-耳环
Necklace	Boolean	否	分割选项-项链
Belongings	Boolean	否	分割选项-随身物品（ 例如伞、包、手机等。 ）

Trace

人体轨迹信息

被如下接口引用：CreatePerson, CreateTrace, SearchTrace。

名称	类型	必选	描述
Images	Array of String	否	人体轨迹图片 Base64 数组。 数组长度最小为1最大为5。 单个图片 base64 编码后大小不可超过2M。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
Urls	Array of String	否	人体轨迹图片 Url 数组。 数组长度最小为1最大为5。 单个图片 base64 编码后大小不可超过2M。 Urls、Images必须提供一个，如果都提供，只使用 Urls。 图片存储于腾讯云的Url可保障更高下载速度和稳定性，建议图片存储于腾讯云。 非腾讯云存储的Url速度和稳定性可能受一定影响。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
BodyRects	Array of BodyRect	否	若输入的Images 和 Urls 是已经裁剪后的人体小图，则可以忽略本参数。 若否，或图片中包含多个人体，则需要通过本参数来指定图片中的人体框。 顺序对应 Images 或 Urls 中的顺序。 当不输入本参数时，我们将认为输入图片已是经过裁剪后的人体小图，不会进行人体检测而直接进行特征提取处理。

TraceInfo

人体轨迹信息。

被如下接口引用：GetPersonList。

名称	类型	描述
TraceId	String	人体轨迹ID。
BodyIds	Array of String	包含的人体轨迹图片Id列表。

UpperBodyCloth

上衣属性信息

被如下接口引用：DetectBody。

名称	类型	描述
Texture	UpperBodyClothTexture	上衣纹理信息。
Color	UpperBodyClothColor	上衣颜色信息。
Sleeve	UpperBodyClothSleeve	上衣衣袖信息。

UpperBodyClothColor

上衣颜色信息。

被如下接口引用：DetectBody。

名称	类型	描述

名称	类型	描述
Type	String	上衣颜色信息，返回值为以下集合中的一个 {红色系, 黄色系, 绿色系, 蓝色系, 黑色系, 灰白色系。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

UpperBodyClothSleeve

上衣衣袖信息。

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	上衣衣袖信息, 返回值为以下集合中的一个 {长袖, 短袖}。
Probability	Float	Type识别概率值，[0.0,1.0],代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

UpperBodyClothTexture

上衣纹理信息。

被如下接口引用：DetectBody。

名称	类型	描述
Type	String	上衣纹理信息，返回值为以下集合中的一个, {纯色, 格子, 大色块}。
Probability	Float	Type识别概率值，[0.0,1.0], 代表判断正确的概率。如0.8则代表有Type值有80%概率正确。

VideoBasicInformation

视频基础信息

被如下接口引用：DescribeSegmentationTask。

名称	类型	描述
FrameWidth	Integer	视频宽度
FrameHeight	Integer	视频高度
FramesPerSecond	Integer	视频帧速率(FPS)
Duration	Float	视频时长
TotalFrames	Integer	视频帧数

错误码

最近更新时间：2021-03-10 08:01:25

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。

错误码	说明
InvalidRequest	请求 body 的 multipart 格式错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
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。
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
FailedOperation.AudioDecodeFailed	音频解码失败。
FailedOperation.AudioEncodeFailed	音频编码失败。
FailedOperation.BodyFeatureFail	人体特征检测失败。
FailedOperation.BodyJointsFail	人体关键点检测失败。
FailedOperation.BodyQualityNotQualified	人体质量分过低。
FailedOperation.BodyRectIllegal	输入的人体框不合法。
FailedOperation.BodyRectNumIllegal	输入的人体框数量不合法。
FailedOperation.CreateTraceFailed	创建轨迹失败，请选择符合要求的人体图片。
FailedOperation.DownloadError	文件下载失败。
FailedOperation.GroupEmpty	搜索的人体库为空。
FailedOperation.ImageBodyDetectFailed	人体检测失败。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。

错误码	说明
FailedOperation.ImageFacedetectFailed	人脸检测失败。
FailedOperation.ImageNotSupported	不支持的图片文件。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionInsufficient	图片分辨率过小。
FailedOperation.ImageSizeExceed	base64编码后的图片数据过大。
FailedOperation.InnerError	服务内部错误，请重试。
FailedOperation.JobQueueFull	任务队列已满。
FailedOperation.NoBodyInPhoto	图片中没有人体。
FailedOperation.ProfileNumExceed	人像数过多。
FailedOperation.RequestEntityTooLarge	整个请求体太大（通常主要是图片）。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SegmentFailed	人像分割失败。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.TaskLimitExceeded	任务超过上限。
FailedOperation.TaskNotExist	任务不存在。
FailedOperation.TerminateTaskFailed	任务取消失败。
FailedOperation.TooLargeFileError	文件太大。
FailedOperation.UnKnowError	内部错误。
FailedOperation.VideoDecodeFailed	视频解码失败。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountTraceNumExceed	账号人体轨迹数量超出限制。
InvalidParameterValue.BodyModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.BodyRectsExceed	传入的人体框过多。
InvalidParameterValue.GroupIdAlreadyExist	人体库ID已经存在。人体库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人体库ID包含非法字符。人体库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人体库ID不存在。
InvalidParameterValue.GroupIdTooLong	人体库ID超出长度限制。
InvalidParameterValue.GroupNameAlreadyExist	人体库名称已经存在。人体库名称不可重复。
InvalidParameterValue.GroupNameTooLong	人体库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人体库数量超出限制。
InvalidParameterValue.GroupTagTooLong	人体库备注超出长度限制。
InvalidParameterValue.GroupTraceNumExceed	人体库人体轨迹数量超出限制。

错误码	说明
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量不在合法范围内。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.OffsetExceed	起始序号过大。请检查需要请求的数组长度。
InvalidParameterValue.PersonIdAlreadyExist	人员ID已经存在。人员ID不可重复。
InvalidParameterValue.PersonIdIllegal	人员ID包含非法字符。人员ID只支持英文、数字、-%@#&_。
InvalidParameterValue.PersonIdNotExist	人员ID不存在。
InvalidParameterValue.PersonIdTooLong	人员ID超出长度限制。
InvalidParameterValue.PersonNameIllegal	人员名称包含非法字符。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.PersonTraceNumExceed	人员人体轨迹数量超出限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.TraceBodyNumExceed	创建人体轨迹的人体图片数量超出限制。
InvalidParameterValue.TraceMatchThresholdIllegal	TraceMatchThreshold参数不合法。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.TooLargeFileError	文件太大。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
UnsupportedOperation.UnknowMethod	未知方法名。