人脸识别

API 文档



腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

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

【商标声明】

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

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

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

文档目录

API 文档
- 产品版本
- 更新历史
- 简介
- API 概览
- 调用方式
  - 请求结构
  - 公共参数
  - 签名方法 v3
  - 签名方法
  - 返回结果
  - 参数类型
- 人脸检测与分析相关接口
  - 人脸检测与分析
  - 人脸检测与属性分析
- 五官定位相关接口
  - 五官定位
  - 稠密关键点
- 人脸比对相关接口
  - 人脸比对
  - 防疫场景人脸比对
  - 人脸相似度检测
- 人员库管理相关接口
  - 创建人员库
  - 删除人员库
  - 获取人员库信息
  - 获取人员库列表
  - 修改人员库
  - 创建人员
  - 删除人员
  - 人员库删除人员
  - 获取人员归属信息
  - 获取人员基础信息
  - 获取指定人员列表
  - 获取人员列表长度
  - 修改人员描述信息
  - 修改人员基础信息
  - 复制人员
  - 增加人脸
  - 删除人脸
- 人脸搜索相关接口
  - 人脸搜索
  - 人脸搜索分库返回
  - 人员搜索
  - 人员搜索按库返回
- 人脸验证相关接口
  - 人员验证
  - 人脸验证

版权所有：腾讯云计算（北京）有限责任公司

人脸静态活体检测（高精度版）相关接口
人脸静态活体检测（高精度版）
数据结构
错误码
人脸识别 API 2018-03-01
更新历史
简介
API 概览
调用方式
请求结构
公共参数
签名方法 v3
签名方法
返回结果
参数类型
人脸检测与分析相关接口
人脸检测与分析
人脸检测与属性分析
五官定位相关接口
五官定位
稠密关键点
人脸比对相关接口
人脸比对
人员库管理相关接口
创建人员库
删除人员库
获取人员库信息
获取人员库列表
修改人员库
创建人员
删除人员
人员库删除人员
获取人员归属信息
获取人员基础信息
获取人员列表
获取人员列表长度
修改人员描述信息
修改人员基础信息
复制人员
增加人脸
删除人脸
人脸搜索相关接口
人脸搜索
人脸搜索分库返回
人员搜索
人员搜索按库返回
人脸验证相关接口
人脸验证
人员验证
人脸静态活体检测相关接口

版权所有：腾讯云计算（北京）有限责任公司

人脸静态活体检测
数据结构
错误码
人脸识别 API 2017
API 概览
错误码说明
鉴权签名
人脸检测
五官定位
人脸比对
个体信息管理
人脸验证
人脸检索
多脸检索
人脸静态活体检测

API 文档
产品版本

最近更新时间：2021-10-21 11:27:45

您正在浏览iai产品文档的版本: 2020-03-03

最新版本

- 2020-03-03 (当前版本)

以往版本

- 2018-03-01

更新历史

最近更新时间：2024-12-27 11:13:41

第 11 次发布

发布时间：2024-12-27 11:13:24

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- DetectFaceSimilarity

第 10 次发布

发布时间：2024-11-14 01:41:32

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- FaceRect
  - **修改成员：** X, Y, Width, Height

第 9 次发布

发布时间：2022-04-25 06:07:47

本次发布包含了以下内容：

改善已有的文档。

删除接口：

- CheckSimilarPerson
- EstimateCheckSimilarPersonCostTime
- GetCheckSimilarPersonJobIdList
- GetSimilarPersonResult

删除数据结构：

- JobIdInfo

第 8 次发布

发布时间：2020-11-30 08:19:25

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- CompareMaskFace

第 7 次发布

发布时间：2020-11-05 08:04:14

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- VerifyPerson
  - 新增出参：Score, IsMatch, FaceModelVersion

第 6 次发布

发布时间：2020-11-02 08:04:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- VerifyPerson
  - **修改入参：** PersonId
  - **删除出参：** Score, IsMatch, FaceModelVersion

第 5 次发布

发布时间：2020-09-18 08:03:48

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- DetectLiveFaceAccurate

第 4 次发布

发布时间：2020-08-13 08:12:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- AnalyzeDenseLandmarks
- DetectFaceAttributes

新增数据结构：

- AttributeItem
- DenseFaceShape
- Eye
- Eyebrow
- FaceDetailAttributesInfo
- FaceDetailInfo
- Hair
- Hat
- HeadPose

- Mouth

第 3 次发布

发布时间：2020-08-06 08:11:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- RevertGroupFaceModelVersion

第 2 次发布

发布时间：2020-08-05 08:12:13

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- GetUpgradeGroupFaceModelVersionJobList
- GetUpgradeGroupFaceModelVersionResult
- UpgradeGroupFaceModelVersion

新增数据结构：

- UpgradeJobInfo

第 1 次发布

发布时间：2020-05-29 14:24:56

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- AnalyzeFace
- CheckSimilarPerson
- CompareFace
- CopyPerson
- CreateFace
- CreateGroup
- CreatePerson
- DeleteFace
- DeleteGroup
- DeletePerson
- DeletePersonFromGroup
- DetectFace
- DetectLiveFace
- EstimateCheckSimilarPersonCostTime
- GetCheckSimilarPersonJobIdList
- GetGroupInfo

- GetGroupList
- GetPersonBaseInfo
- GetPersonGroupInfo
- GetPersonList
- GetPersonListNum
- GetSimilarPersonResult
- ModifyGroup
- ModifyPersonBaseInfo
- ModifyPersonGroupInfo
- SearchFaces
- SearchFacesReturnsByGroup
- SearchPersons
- SearchPersonsReturnsByGroup
- VerifyFace
- VerifyPerson

新增数据结构：

- Candidate
- FaceAttributesInfo
- FaceHairAttributesInfo
- FaceInfo
- FaceQualityCompleteness
- FaceQualityInfo
- FaceRect
- FaceShape
- GroupCandidate
- GroupExDescriptionInfo
- GroupInfo
- JobIdInfo
- PersonExDescriptionInfo
- PersonGroupInfo
- PersonInfo
- Point
- Result
- ResultsReturnsByGroup

简介

最近更新时间：2025-04-25 01:35:40

精准识别

腾讯云人脸识别服务在多个国际公开竞赛中刷新纪录，人脸比对在 2017 年的LFW 测评准确度高达 99.80%，人脸检索在 MegaFace 竞赛百万规模中首选识别率 83.29% 排名第一，识别准确率业界领先。

稳定性强

腾讯云人脸识别服务已经通过腾讯内部产品海量用户和复杂场景验证，运行稳定、鲁棒性强，服务可用性可达99.9%。

算法领先

基于第三代腾讯优图祖母模型，融合度量学习、迁移学习、多任务学习等多种训练手段来优化模型；针对不同场景业务的特点，定制化finetuning或distilling模型，满足业务性能与时延的双重需求。

简单易用

提供了丰富多样的API接口和离线SDK，快速植入应用和硬件设备，全面满足云和端的需求。

用途广泛

广泛应用于刷脸门禁考勤、人脸签到、人脸支付和人脸登录等众多场景。

实时响应

人脸识别具备高并发、高吞吐、低时延等特点，即使是百万规模人脸搜索，仍只需数百毫秒即可处理完毕，满足您的实时使用需求。

版权所有：腾讯云计算（北京）有限责任公司

API 概览

最近更新时间：2024-12-27 11:13:40

人脸检测与分析相关接口

接口名称	接口功能
DetectFace	人脸检测与分析
DetectFaceAttributes	人脸检测与属性分析

五官定位相关接口

接口名称	接口功能
AnalyzeDenseLandmarks	稠密关键点
AnalyzeFace	五官定位

人脸比对相关接口

接口名称	接口功能
CompareFace	人脸比对
CompareMaskFace	防疫场景人脸比对
DetectFaceSimilarity	人脸相似度检测

人员库管理相关接口

接口名称	接口功能
CreateGroup	创建人员库
DeleteGroup	删除人员库
GetGroupInfo	获取人员库信息
GetGroupList	获取人员库列表
ModifyGroup	修改人员库
CreatePerson	创建人员
DeletePerson	删除人员
DeletePersonFromGroup	人员库删除人员
GetPersonGroupInfo	获取人员归属信息
GetPersonBaseInfo	获取人员基础信息
GetPersonList	获取指定人员列表
GetPersonListNum	获取人员列表长度

版权所有：腾讯云计算（北京）有限责任公司

接口名称	接口功能
ModifyPersonGroupInfo	修改人员描述信息
ModifyPersonBaseInfo	修改人员基础信息
CopyPerson	复制人员
CreateFace	增加人脸
DeleteFace	删除人脸
GetUpgradeGroupFaceModelVersionJobList	获取人员库升级任务列表
GetUpgradeGroupFaceModelVersionResult	人员库升级结果查询
RevertGroupFaceModelVersion	人员库版本回滚
UpgradeGroupFaceModelVersion	人员库升级

人脸搜索相关接口

接口名称	接口功能
SearchFaces	人脸搜索
SearchFacesReturnsByGroup	人脸搜索分库返回
SearchPersons	人员搜索
SearchPersonsReturnsByGroup	人员搜索按库返回

人脸验证相关接口

接口名称	接口功能
VerifyFace	人脸验证
VerifyPerson	人员验证

人脸静态活体检测（高精度版）相关接口

接口名称	接口功能
DetectLiveFaceAccurate	人脸静态活体检测（高精度版）

人脸静态活体检测相关接口

接口名称	接口功能
DetectLiveFace	人脸静态活体检测

> ⚠ 注意：
> 以上给出的接口频率限制维度为 `API` + 接入地域 + 子账号，有关限频更多说明参考：API 频率限制说明

调用方式
请求结构

最近更新时间：2025-05-21 01:20:48

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

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

版权所有：腾讯云计算（北京）有限责任公司

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

2. 通信协议

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

3. 请求方法

支持的 HTTP 请求方法：

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

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

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

4. 字符编码

均使用 `UTF-8` 编码。

公共参数

最近更新时间：2024-11-15 01:39:35

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

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

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

HTTP GET 请求结构示例：

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

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

HTTP POST（application/json）请求结构示例：

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

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

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

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

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

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

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

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

10
--58731222010402--
```

使用签名方法 v1 的公共参数

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

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。**注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。**
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

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

HTTP GET 请求结构示例:

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

Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例:

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

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

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

地域列表

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

地域	取值
华北地区（北京）	ap-beijing
西南地区（成都）	ap-chengdu
西南地区（重庆）	ap-chongqing
华南地区（广州）	ap-guangzhou
华东地区（南京）	ap-nanjing
华东地区（上海）	ap-shanghai
亚太东南（新加坡）	ap-singapore

版权所有：腾讯云计算（北京）有限责任公司

签名方法 v3

最近更新时间：2024-12-25 01:38:54

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

推荐使用 API Explorer　　点击调试

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

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

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份
   签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 云 API 密钥 页面获取密钥相关信息。

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

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

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

申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

签名版本 v3 签名过程

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

下面以云服务器查询广州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。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC- 前缀）。

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

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

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST ）。此示例取值为 `POST`。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠（/）。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）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 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；

字段名称	解释
	2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 `content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n`。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号（;）分隔。 此示例为 content-type;host;x-tc-action
HashedRequestPayload	请求正文（payload，即 body，此示例为 `{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}`）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 `35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064`。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

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

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

字段名称	解释
Algorithm	签名算法，目前固定为 `TC3-HMAC-SHA256`。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 `1551113065`。
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的

字段名称	解释
	UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

注意：

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

根据以上规则，示例中得到的待签名字符串如下：

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

3. 计算签名

1）计算派生签名密钥，伪代码如下：

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

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

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

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

2）计算签名，伪代码如下：

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

此示例计算结果是 `10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f`。

4. 拼接 Authorization

按如下格式拼接 Authorization：

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

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

根据以上规则，示例中得到的值为：

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

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID********************************/2019-02-25/cvm/tc3_request,
SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b4
6093f6ab6c4f
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

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

> ⚠ **注意：**
>
> **请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过**

> 打印实际请求内容，网络抓包等方式对比排查。

签名演示

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

- Python
- Java
- PHP
- Go
- NodeJS
- .NET
- C++
- Ruby

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

- Signature Demo

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

Java

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

public class TencentCloudAPITC3Demo {
private final static Charset UTF8 = StandardCharsets.UTF_8;
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID********************************
private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
private final static String CT_JSON = "application/json; charset=utf-8";

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

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

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

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

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instan
ce-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" + hashedCanonicalReques
t;
System.out.println(stringToSign);

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

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

```
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

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

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

Python

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

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

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
```

```
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

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


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

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

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + host + '"'
```

```
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'")
```

Golang

```
package main

import (
"crypto/hmac"
"crypto/sha256"
"encoding/hex"
"fmt"
"os"
"strings"
"time"
)

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

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

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

// step 1: build canonical request string
httpRequestMethod := "POST"
canonicalURI := "/"
canonicalQueryString := ""
```

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

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

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

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

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

```
fmt.Println(curl)
}
```

PHP

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

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = implode("\n", [
"content-type:application/json; charset=utf-8",
"host:".$host,
"x-tc-action:".strtolower($action),
""
]);
$signedHeaders = implode(";", [
"content-type",
"host",
"x-tc-action",
]);
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

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

```
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

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

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

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

Ruby

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

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

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
```

```
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

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

puts canonical_request

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

# ************* 步骤 3：计算签名 *************
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

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

```
Signature=#{signature}"
puts authorization

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

DotNet

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

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

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

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

```
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZer
o) / 1000;
// ************* 步骤 1：拼接规范请求串 *************
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

// ************* 步骤 2：拼接待签名字符串 *************
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n"
+ requestTimestamp.ToString() + "\n"
+ credentialScope + "\n"
+ hashedCanonicalRequest;
Console.WriteLine(stringToSign);

// ************* 步骤 3：计算签名 *************
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);

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

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
```

```
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}
public static void Main(string[] args)
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID*******************************
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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

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

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

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

NodeJS

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

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

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

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

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

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

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

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

const canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload
```

```
console.log(canonicalRequest)

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

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

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

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

C++

```
#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
```

```
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

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

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

string sha256Hex(const string &str)
{
char buf[3];
unsigned char hash[SHA256_DIGEST_LENGTH];
SHA256_CTX sha256;
SHA256_Init(&sha256);
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
snprintf(buf, sizeof(buf), "%02x", hash[i]);
NewString = NewString + buf;
}
return NewString;
}

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

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

```
#else
h = HMAC_CTX_new();
#endif

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

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

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

return (ss.str());
}

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

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

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

string service = "cvm";
```

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

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

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

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

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

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

C

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

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

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

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
```

```
unsigned char* output, unsigned int* output_len)
{
HMAC_CTX *h;
#if OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

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


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

}

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

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

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

int main()
```

```
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID********************************
const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
const char* service = "cvm";
const char* host = "cvm.tencentcloudapi.com";
const char* region = "ap-guangzhou";
const char* action = "DescribeInstances";
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

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

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


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

```
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

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

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

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

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

其他语言

- Lua: GitHub
- Swift: GitHub
- Dart: GitHub
- Shell(Bash): GitHub

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

签名方法

最近更新时间：2024-12-25 01:38:55

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

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

推荐使用 API Explorer　　点击调试

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

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

1. 申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

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

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

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKID********************************
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886

参数名称	中文	参数值
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

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

2.1. 对参数排序

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

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

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

2.2. 拼接请求字符串

此步骤生成请求字符串。
将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 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&Secre
tId=AKID*******************************&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

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

4. 请求字符串: 即上一步生成的请求字符串。

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

示例的拼接结果为：

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

2.4. 生成签名串

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

具体代码如下，以 PHP 语言为例：

```
$secretKey = '********************************';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID********************************&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ= ，最终得到的签名串请求参数（Signature）为：7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 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

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

5. 签名演示

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

- Python
- Java
- PHP
- Go
- NodeJS
- .NET
- C++
- Ruby

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

- Signature Demo

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID********************************&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12` 。

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

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

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

Java

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

public class TencentCloudAPIDemo {
```

```
private final static String CHARSET = "UTF-8";

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

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

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

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID********************************
params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
params.put("Action", "DescribeInstances"); // 公共参数
params.put("Version", "2017-03-12"); // 公共参数
params.put("Region", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA
1")); // 公共参数
System.out.println(getUrl(params));
}
}
```

Python

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

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

import requests

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

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

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

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

Golang

```
package main

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

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

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

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

for i := range keys {
k := keys[i]
buf.WriteString(k)
buf.WriteString("=")
buf.WriteString(params[k])
buf.WriteString("&")
```

```
}
buf.Truncate(buf.Len() - 1)

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

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

PHP

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

ksort($param);

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

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

Ruby

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

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

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

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

DotNet

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

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

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

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

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

NodeJS

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

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

function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}
function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

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

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

```
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

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

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

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

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

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

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

返回结果

最近更新时间：2024-03-12 01:32:54

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

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

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

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

公共错误码

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

参数类型

最近更新时间：2022-08-10 06:33:47

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

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

人脸检测与分析相关接口
人脸检测与分析

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

检测给定图片中的人脸（Face）的位置、相应的面部属性和人脸质量信息，位置包括(x，y，w，h)，面部属性包括性别（gender）、年龄（age）、表情（expression）、魅力（beauty）、眼镜（glass）、发型（hair）、口罩（mask）和姿态(pitch，roll，yaw)，人脸质量信息包括整体质量分（score）、模糊分（sharpness）、光照分（brightness）和五官遮挡分（completeness）。

其中，人脸质量信息主要用于评价输入的人脸图片的质量。在使用人脸识别服务时，建议您对输入的人脸图片进行质量检测，提升后续业务处理的效果。该功能的应用场景包括：

1）人员库创建人员/增加人脸：保证人员人脸信息的质量，便于后续的业务处理。

2）人脸搜索：保证输入的图片质量，快速准确匹配到对应的人员。

3）人脸验证：保证人脸信息的质量，避免明明是本人却认证不通过的情况。

4）人脸融合：保证上传的人脸质量，人脸融合的效果更好。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：DetectFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
MaxFaceNum	否	Integer	最多处理的人脸数目。 - 默认值为1（仅检测图片中面积最大的那张人脸），最大值为120。 - 此参数用于控制处理待检测图片中的人脸个数，值越小，处理速度越快。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素，低于MinFaceSize值的人脸不会被检测。 - 只支持设置34和20，建议使用34。 示例值：34
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。

参数名称	必选	类型	描述
			- 所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 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。 示例值：https://test.image.myqcloud.com/testB.jpg
NeedFaceAttributes	否	Integer	是否需要返回人脸属性信息（FaceAttributesInfo）。 - 0 为不需要返回，1 为需要返回。 - 默认为 0。 - 非 1 值均视为不需要返回，此时 FaceAttributesInfo 不具备参考意义。 - 最多返回面积最大的 5 张人脸属性信息，超过 5 张人脸（第 6 张及以后的人脸）的 FaceAttributesInfo 不具备参考意义。 - 提取人脸属性信息较为耗时，如不需要人脸属性信息，建议关闭此项功能，加快人脸检测速度。 示例值：0
NeedQualityDetection	否	Integer	是否开启质量检测。 - 0 为关闭，1 为开启。 - 默认为 0。 - 非 1 值均视为不进行质量检测。 - 最多返回面积最大的 30 张人脸质量分信息，超过 30 张人脸（第 31 张及以后的人脸）的 FaceQualityInfo不具备参考意义。 - 建议：人脸入库操作建议开启此功能。 示例值：0
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 - 目前入参支持 “2.0”和“3.0“ 两个输入。 - 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 - 2020年11月26日后开通服务的账号仅支持输入“3.0”。 - 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：640

参数名称	类型	描述
ImageHeight	Integer	请求的图片高度。 示例值：440
FaceInfos	Array of FaceInfo	人脸信息列表。 - 包含人脸坐标信息、属性信息（若需要）、质量分信息（若需要）。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸检测与分析接口成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DetectFace
&MaxFaceNum=1
&MinFaceSize=40
&Url=http://test.image.myqcloud.com/testB.jpg
&NeedFaceAttributes=0
&NeedQualityDetection=0
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ImageWidth": 640,
"ImageHeight": 440,
"FaceInfos": [
{
"X": 156,
"Y": 129,
"Width": 196,
"Height": 196,
"FaceAttributesInfo": {
"Gender": 0,
"Age": 0,
"Expression": 0,
"Hat": false,
"Glass": false,
"EyeOpen": true,
"Mask": false,
"Hair": {
"Length": 0,
"Bang": 0,
"Color": 0
```

```
},
"Pitch": 0,
"Yaw": 0,
"Roll": 0,
"Beauty": 0
},
"FaceQualityInfo": {
"Score": 0,
"Sharpness": 0,
"Brightness": 0,
"Completeness": {
"Eyebrow": 0,
"Eye": 0,
"Nose": 0,
"Cheek": 0,
"Mouth": 0,
"Chin": 0
}
}
}
],
"FaceModelVersion": "3.0",
"RequestId": "a574102d-1b86-48a7-a08b-6a741a8fedb6"
}
}
```


示例2 人脸检测与分析接口开启质量检测成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DetectFace
&MaxFaceNum=1
&MinFaceSize=40
&Url=http://test.image.myqcloud.com/testA.jpg
&NeedFaceAttributes=1
&NeedQualityDetection=1
&<公共请求参数>
```

输出示例


```
{
"Response": {
"ImageWidth": 640,
"ImageHeight": 440,
"FaceInfos": [
{
"X": 156,
"Y": 129,
"Width": 196,
```

```
"Height": 196,
"FaceAttributesInfo": {
"Gender": 99,
"Age": 51,
"Expression": 99,
"Hat": false,
"Glass": false,
"EyeOpen": true,
"Mask": true,
"Hair": {
"Length": 1,
"Bang": 1,
"Color": 0
},
"Pitch": 17,
"Yaw": 5,
"Roll": -2,
"Beauty": 71
},
"FaceQualityInfo": {
"Score": 63,
"Sharpness": 73,
"Brightness": 47,
"Completeness": {
"Eyebrow": 99,
"Eye": 99,
"Nose": 99,
"Cheek": 52,
"Mouth": 99,
"Chin": 47
}
}
}
],
"FaceModelVersion": "3.0",
"RequestId": "bcde47b5-e6d8-446e-a538-bcecffbe306a"
}
}
```


示例3 人脸检测与分析接口失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DetectFace
&MaxFaceNum=1
&MinFaceSize=40
&Url=http://test.image.myqcloud.com/testA.jpg
&NeedFaceAttributes=1
```

```
&NeedQualityDetection=1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "54eedbca-8704-4388-9b1c-4bdf1a4308a4",
"Error": {
"Code": "InvalidParameterValue.NoFaceInPhoto",
"Message": "图片中没有人脸。"
}
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation	操作失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。

错误码	描述
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸检测与属性分析

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

检测给定图片中的人脸（Face）的位置、相应的面部属性和人脸质量信息，位置包括(x，y，w，h)，面部属性包括性别（gender）、年龄（age）、表情（expression）、魅力（beauty）、眼镜（glass）、发型（hair）、口罩（mask）和姿态(pitch，roll，yaw)。

其中，人脸质量信息主要用于评价输入的人脸图片的质量。在使用人脸识别服务时，建议您对输入的人脸图片进行质量检测，提升后续业务处理的效果。该功能的应用场景包括：

1）人员库创建人员/增加人脸：保证人员人脸信息的质量，便于后续的业务处理。

2）人脸搜索：保证输入的图片质量，快速准确匹配到对应的人员。

3）人脸验证：保证人脸信息的质量，避免明明是本人却认证不通过的情况。

4）人脸融合：保证上传的人脸质量，人脸融合的效果更好。

- 本接口是人脸检测与分析的升级，具体在于：
  1.本接口可以指定需要计算返回的人脸属性，避免无效计算，降低耗时；
  2.本接口支持更多属性细项数，也会持续增加更多功能。
  请您使用本接口完成相应的人脸检测与属性分析需求。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：DetectFaceAttributes。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
MaxFaceNum	否	Integer	最多处理的人脸数目。 - 默认值为1（仅检测图片中面积最大的那张人脸），最大值为120。 - 此参数用于控制处理待检测图片中的人脸个数，值越小，处理速度越快。 示例值：1
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。

参数名称	必选	类型	描述
			- 所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 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。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceAttributesType	否	String	是否返回年龄、性别、情绪等属性。 - 合法值为（大小写不敏感）：None、Age、Beauty、Emotion、Eye、Eyebrow、Gender、Hair、Hat、Headpose、Mask、Mouth、Moustache、Nose、Shape、Skin、Smile。 - None为不需要返回。 - 默认为 None。即FaceAttributesType属性为空时，各属性返回值为0。 - 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 张人脸属性信息，超过 5 张人脸（第 6 张及以后的人脸）的 AttributesInfo 不具备参考意义。 示例值：eye
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。本接口仅支持“3.0”输入 示例值：3.0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：550
ImageHeight	Integer	请求的图片高度。 示例值：366
FaceDetailInfos	Array of FaceDetailInfo	人脸信息列表。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸检测与属性分析成功示例

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"FaceAttributesType": "eye"
}
```

输出示例

```
{
"Response": {
"ImageWidth": 550,
"ImageHeight": 366,
"FaceModelVersion": "3.0",
"FaceDetailInfos": [
{
"FaceRect": {
"X": 375,
"Y": 37,
"Width": 63,
"Height": 82
},
"FaceDetailAttributesInfo": {
"Age": 0,
"Beauty": 0,
"Emotion": {
"Type": 0,
"Probability": 0
},
"Eye": {
"Glass": {
"Type": 0,
"Probability": 0.99936753511429
},
"EyeOpen": {
"Type": 0,
"Probability": 0.99949336051941
},
"EyelidType": {
"Type": 1,
"Probability": 0.75467920303345
},
```

```
"EyeSize": {
"Type": 2,
"Probability": 0.59152442216873
}
},
"Eyebrow": {
"EyebrowDensity": {
"Type": 0,
"Probability": 0
},
"EyebrowCurve": {
"Type": 0,
"Probability": 0
},
"EyebrowLength": {
"Type": 0,
"Probability": 0
}
},
"Gender": {
"Type": 0,
"Probability": 0
},
"Hair": {
"Length": {
"Type": 0,
"Probability": 0
},
"Bang": {
"Type": 0,
"Probability": 0
},
"Color": {
"Type": 0,
"Probability": 0
}
},
"Hat": {
"Style": {
"Type": 0,
"Probability": 0
},
"Color": {
"Type": 0,
"Probability": 0
}
},
"HeadPose": {
"Pitch": 0,
"Yaw": 0,
```

```
"Roll": 0
},
"Mask": {
"Type": 0,
"Probability": 0
},
"Mouth": {
"MouthOpen": {
"Type": 0,
"Probability": 0
}
},
"Moustache": {
"Type": 0,
"Probability": 0
},
"Nose": {
"Type": 2,
"Probability": 0.75233882665634
},
"Shape": {
"Type": 0,
"Probability": 0
},
"Skin": {
"Type": 0,
"Probability": 0
},
"Smile": 0
}
}
],
"RequestId": "b2c154b9-4620-4d37-8fd1-f6af3748f998"
}
}
```


示例2 人脸检测与属性分析失败示例

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testB.jpg",
```

```
"FaceAttributesType": "eye"
}
```

输出示例

```
{
"Response": {
"RequestId": "ab0ebc1d-7a0e-4327-808b-3d24322a97dd",
"Error": {
"Code": "InvalidParameterValue.NoFaceInPhoto",
"Message": "图片中没有人脸。"
}
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.ImageDecodeFailed	图片解码失败。

错误码	描述
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

五官定位相关接口
五官定位

最近更新时间：2025-04-25 01:35:40

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对请求图片进行五官定位（也称人脸关键点定位），计算构成人脸轮廓的 90 个点，包括眉毛（左右各 8 点）、眼睛（左右各 8 点）、鼻子（13 点）、嘴巴（22 点）、脸型轮廓（21 点）、眼珠[或瞳孔]（2点）。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：AnalyzeFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
Mode	否	Integer	检测模式。 - 取值范围： 0 为检测所有出现的人脸。 1 为检测面积最大的人脸。 - 默认为 0。 - 最多返回 10 张人脸的五官定位（人脸关键点）具体信息。 示例值：0
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 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。

参数名称	必选	类型	描述
			- 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 - 目前入参支持 “2.0”和“3.0“ 两个输入。 - 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 - 2020年11月26日后开通服务的账号仅支持输入“3.0”。 - 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：550
ImageHeight	Integer	请求的图片高度。 示例值：366
FaceShapeSet	Array of FaceShape	五官定位（人脸关键点）具体信息。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 五官定位接口成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=AnalyzeFace
&Mode=0
&Url=http://test.image.myqcloud.com/testA.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ImageWidth": 550,
"ImageHeight": 366,
"FaceShapeSet": [
{
"FaceProfile": [
{
"X": 63,
"Y": 335
},
{
"X": 63,
"Y": 374
},
{
"X": 66,
"Y": 412
},
{
"X": 74,
"Y": 450
},
{
"X": 85,
"Y": 487
},
{
"X": 100,
"Y": 522
},
{
"X": 121,
"Y": 554
},
{
"X": 147,
"Y": 582
},
{
"X": 176,
"Y": 608
},
{
"X": 208,
"Y": 627
},
{
"X": 245,
```

```
"Y": 634
},
{
"X": 282,
"Y": 627
},
{
"X": 315,
"Y": 607
},
{
"X": 344,
"Y": 582
},
{
"X": 370,
"Y": 554
},
{
"X": 391,
"Y": 522
},
{
"X": 405,
"Y": 487
},
{
"X": 416,
"Y": 449
},
{
"X": 423,
"Y": 411
},
{
"X": 427,
"Y": 372
},
{
"X": 426,
"Y": 334
}
],
"LeftEye": [
{
"X": 114,
"Y": 333
},
{
"X": 128,
```

```
"Y": 345
},
{
"X": 146,
"Y": 349
},
{
"X": 165,
"Y": 347
},
{
"X": 183,
"Y": 341
},
{
"X": 169,
"Y": 325
},
{
"X": 150,
"Y": 318
},
{
"X": 130,
"Y": 321
}
],
"RightEye": [
{
"X": 378,
"Y": 331
},
{
"X": 364,
"Y": 343
},
{
"X": 345,
"Y": 348
},
{
"X": 327,
"Y": 346
},
{
"X": 309,
"Y": 340
},
{
"X": 322,
```

```
"Y": 323
},
{
"X": 341,
"Y": 316
},
{
"X": 361,
"Y": 319
}
],
"LeftEyeBrow": [
{
"X": 79,
"Y": 280
},
{
"X": 108,
"Y": 275
},
{
"X": 138,
"Y": 274
},
{
"X": 168,
"Y": 277
},
{
"X": 198,
"Y": 279
},
{
"X": 173,
"Y": 256
},
{
"X": 139,
"Y": 251
},
{
"X": 105,
"Y": 256
}
],
"RightEyeBrow": [
{
"X": 410,
"Y": 277
},
```

```
{
"X": 380,
"Y": 273
},
{
"X": 350,
"Y": 272
},
{
"X": 320,
"Y": 275
},
{
"X": 290,
"Y": 277
},
{
"X": 315,
"Y": 255
},
{
"X": 349,
"Y": 249
},
{
"X": 384,
"Y": 254
}
],
"Mouth": [
{
"X": 173,
"Y": 522
},
{
"X": 193,
"Y": 541
},
{
"X": 217,
"Y": 554
},
{
"X": 244,
"Y": 558
},
{
"X": 272,
"Y": 554
},
```


版权所有：腾讯云计算（北京）有限责任公司

```
{
"X": 297,
"Y": 541
},
{
"X": 317,
"Y": 522
},
{
"X": 291,
"Y": 517
},
{
"X": 264,
"Y": 512
},
{
"X": 245,
"Y": 517
},
{
"X": 225,
"Y": 512
},
{
"X": 199,
"Y": 517
},
{
"X": 196,
"Y": 528
},
{
"X": 220,
"Y": 532
},
{
"X": 244,
"Y": 535
},
{
"X": 269,
"Y": 532
},
{
"X": 293,
"Y": 528
},
{
"X": 293,
```

```
"Y": 525
},
{
"X": 269,
"Y": 527
},
{
"X": 245,
"Y": 530
},
{
"X": 221,
"Y": 528
},
{
"X": 197,
"Y": 525
}
],
"Nose": [
{
"X": 244,
"Y": 428
},
{
"X": 245,
"Y": 341
},
{
"X": 231,
"Y": 367
},
{
"X": 217,
"Y": 392
},
{
"X": 203,
"Y": 418
},
{
"X": 187,
"Y": 448
},
{
"X": 217,
"Y": 464
},
{
"X": 245,
```

```
"Y": 468
},
{
"X": 272,
"Y": 464
},
{
"X": 302,
"Y": 448
},
{
"X": 287,
"Y": 417
},
{
"X": 273,
"Y": 392
},
{
"X": 259,
"Y": 366
}
],
"LeftPupil": [
{
"X": 188,
"Y": 303
}
],
"RightPupil": [
{
"X": 201,
"Y": 304
}
]
}
],
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```


示例2 五官定位接口失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=AnalyzeFace
&Mode=0
```

```
&Url=http://test.image.myqcloud.com/testB.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "c964ad35-5476-486e-a03f-311faa6652e6",
"Error": {
"Code": "InvalidParameterValue.NoFaceInPhoto",
"Message": "图片中没有人脸。"
}
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。

错误码	描述
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。

版权所有：腾讯云计算（北京）有限责任公司

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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

稠密关键点

最近更新时间：2025-04-25 01:35:40

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对请求图片进行五官定位（也称人脸关键点定位），获得人脸的精准信息，返回多达888点关键信息，对五官和脸部轮廓进行精确定位。

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：AnalyzeDenseLandmarks。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	否	String	公共参数，此参数为可选参数。
Mode	否	Integer	检测模式。 - 取值范围：0 为检测所有出现的人脸，1 为检测面积最大的人脸。 - 默认为 0。 - 最多返回 5 张人脸的五官定位（人脸关键点）具体信息。 示例值：0
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 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。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。本接口仅支持 “3.0 “ 输入。 示例值：3.0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参

参数名称	必选	类型	描述
			数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：550
ImageHeight	Integer	请求的图片高度。 示例值：336
DenseFaceShapeSet	Array of DenseFaceShape	稠密人脸关键点具体信息。
FaceModelVersion	String	人脸识别服务所用的算法模型版本。本接口仅支持 “3.0 “ 输入。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId ）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取稠密关键点成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=AnalyzeDenseLandmarks
&Mode=0
&Url=http://test.image.myqcloud.com/testA.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ImageWidth": 550,
"ImageHeight": 366,
"DenseFaceShapeSet": [
{
"Chin": [
{
"X": 376,
"Y": 66
}
],
"LeftEye": [
{
"X": 384,
```

```
"Y": 66
}
],
"RightEye": [
{
"X": 422,
"Y": 72
}
],
"LeftEyeBrow": [
{
"X": 382,
"Y": 61
}
],
"RightEyeBrow": [
{
"X": 428,
"Y": 68
}
],
"Nose": [
{
"X": 398,
"Y": 69
}
],
"MouthInside": [
{
"X": 386,
"Y": 98
}
],
"MouthOutside": [
{
"X": 386,
"Y": 98
}
],
"LeftPupil": [
{
"X": 390,
"Y": 65
}
],
"RightPupil": [
{
"X": 416,
"Y": 70
}
```


版权所有：腾讯云计算（北京）有限责任公司

```
],
"LeftEyeBags": [
{
"X": 395,
"Y": 69
}
],
"RightEyeBags": [
{
"X": 410,
"Y": 72
}
],
"Forehead": [
{
"X": 376,
"Y": 65
}
],
"CentralAxis": [
{
"X": 405,
"Y": 54
}
],
"X": 375,
"Y": 37,
"Width": 63,
"Height": 82
}
],
"FaceModelVersion": "3.0",
"RequestId": "cebb639a-ec53-4a23-a5a5-fbcc5f0ae2bc"
}
}
```


示例2 获取稠密关键点失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=AnalyzeDenseLandmarks
&Mode=0
&Url=http://test.image.myqcloud.com/testB.jpg
&<公共请求参数>
```

输出示例


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

```
"Error": {
"Code": "FailedOperation.ImageDecodeFailed",
"Message": "图片解码失败。"
},
"RequestId": "b0d281e1-1df9-461d-8ed4-5ce83b29530c"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded	超过配额限制。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸比对相关接口
人脸比对

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对两张图片中的人脸进行相似度比对，返回人脸相似度分数。

若您需要判断 “此人是否是某人”，即验证某张图片中的人是否是已知身份的某人，如常见的人脸登录场景，建议使用人脸验证或人员验证接口。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：CompareFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
ImageA	否	String	A 图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中置信度最高的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
ImageB	否	String	B 图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中置信度最高的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
UrlA	否	String	A 图片的 Url。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - A 图片的 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。

参数名称	必选	类型	描述
			- 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
UrlB	否	String	B 图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - B 图片的 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testB.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 - 目前入参支持 “2.0”和“3.0 “ 两个输入。 - 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 - 2020年11月26日后开通服务的账号仅支持输入“3.0”。 - 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制； 1: 较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多，在某一维度上存在轻微问题； 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	两张图片中人脸的相似度分数。 - 不同算法版本返回的相似度分数不同。 - 若需要验证两张图片中人脸是否为同一人，3.0版本误识率千分之一对应分数为40分，误识率万分之一对应分数为50分，误识率十万分之一对应分数为60分。 一般超过50分则可认定为同一人。 - 2.0版本误识率千分之一对应分数为70分，误识率万分之一对应分数为80分，误识率十万分之一对应分数为90分。 一般超过80分则可认定为同一人。

参数名称	类型	描述
		- 若需要验证两张图片中的人脸是否为同一人，建议使用人脸验证接口。 示例值：0.999
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸比对接口成功示例

输入示例

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Score": 0.999,
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

示例2 人脸比对接口异常示例

输入示例

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
```

```
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.FaceQualityNotQualified",
"Message": "人脸图片质量不符要求。"
},
"RequestId": "788270e8-91b7-4c59-bd00-1af257b1e482"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。

错误码	描述
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

防疫场景人脸比对

最近更新时间：2025-04-25 01:35:35

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对两张图片中的人脸进行相似度比对，返回人脸相似度分数。

防疫场景人脸比对接口可在人脸戴口罩情况下使用，口罩遮挡程度最高可以遮挡鼻尖。

如图片人脸不存在防疫场景下戴口罩的情况，建议使用人脸比对服务。

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：CompareMaskFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	否	String	公共参数，此参数为可选参数。
ImageA	否	String	A 图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
ImageB	否	String	B 图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
UrlA	否	String	A 图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - A 图片的 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg

参数名称	必选	类型	描述
UrlB	否	String	B 图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - B 图片的 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testB.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。该接口只支持"3.0" 示例值：3.0
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	两张图片中人脸的相似度分数。 - 不同算法版本返回的相似度分数不同。 - 若需要验证两张图片中人脸是否为同一人，3.0版本误识率千分之一对应分数为40分，误识率万分之一对应分数为50分，误识率十万分之一对应分数为60分。 一般超过50分则可认定为同一人。 - 2.0版本误识率千分之一对应分数为70分，误识率万分之一对应分数为80分，误识率十万分之一对应分数为90分。 一般超过80分则可认定为同一人。 - 若需要验证两张图片中的人脸是否为同一人，建议使用人脸验证接口。 示例值：99
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸比对接口成功示例

输入示例

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

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Score": 0.999,
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

示例2 人脸比对接口异常示例

输入示例

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.FaceQualityNotQualified",
"Message": "人脸图片质量不符要求。"
},
"RequestId": "788270e8-91b7-4c59-bd00-1af257b1e481"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。

错误码	描述
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸相似度检测

最近更新时间：2025-04-25 01:35:35

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对两张图片中的人脸进行相似度比对，返回人脸相似度分数。

若您需要判断 “此人是否是某人”，即验证某张图片中的人是否是已知身份的某人，如常见的人脸登录场景，建议使用人脸验证或人员验证接口。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：DetectFaceSimilarity。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	否	String	公共参数，此参数为可选参数。
ImageA	否	String	A 图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中置信度最高的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
ImageB	否	String	B 图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中置信度最高的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
UrlA	否	String	A 图片的 Url。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - A 图片的 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。

参数名称	必选	类型	描述
			- 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
UrlB	否	String	B 图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - B 图片的 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testB.jpg
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制； 1: 较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多，在某一维度上存在轻微问题； 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	取值范围 [0.00, 100.00]。 推荐相似度大于等于70时可判断为同一人，可根据具体场景自行调整阈值（阈值70的误通过率为千分之一，阈值80的误通过率是万分之一）。 示例值：0.999
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸相似度检测接口成功示例

输入示例

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
```

```
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Score": 0.999,
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

示例2 人脸相似度检测接口异常示例

输入示例

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.FaceQualityNotQualified",
"Message": "人脸图片质量不符要求。"
},
"RequestId": "788270e8-91b7-4c59-bd00-1af257b1e482"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.CompareFail	比对失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.LifePhotoDetectFaces	检测到多张人脸。
FailedOperation.LifePhotoDetectFake	实人比对没通过。
FailedOperation.LifePhotoDetectNoFaces	未能检测到完整人脸。
FailedOperation.LifePhotoSizeError	传入图片过大或过小。
FailedOperation.RequestTimeout	后端服务超时。

错误码	描述
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员库管理相关接口
创建人员库

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于创建一个空的人员库，如果人员库已存在返回错误。
可根据需要创建自定义描述字段，用于辅助描述该人员库下的人员信息。

1个APPID下最多创建10万个人员库（Group）、最多包含5000万张人脸（Face）。

不同算法模型版本（FaceModelVersion）的人员库（Group）最多可包含人脸（Face）数不同。算法模型版本为2.0的人员库最多包含100万张人脸，算法模型版本为3.0的人员库最多可包含300万张人脸。

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：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupName	是	String	人员库名称，[1,60]个字符，可修改，不可重复。 示例值：腾讯深圳员工库
GroupId	是	String	人员库 ID，不可修改，不可重复。支持英文、数字、-%@#&_，长度限制64B。 示例值：TencentShenZhenEmployee
GroupExDescriptions.N	否	Array of String	人员库自定义描述字段，用于描述人员库中人员属性，该人员库下所有人员将拥有此描述字段。 - 最多可以创建5个。 - 每个自定义描述字段支持[1,30]个字符。 - 在同一人员库中自定义描述字段不可重复。 - 例： 设置某人员库“自定义描述字段”为["学号","工号","手机号"]， 则该人员库下所有人员将拥有名为“学号”、“工号”、“手机号”的描述字段， 可在对应人员描述字段中填写内容，登记该人员的学号、工号、手机号等信息。 示例值：["学号","工号","手机号"]
Tag	否	String	人员库信息备注，[0，40]个字符。 示例值：不含实习生
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 - 目前入参支持 “2.0”和“3.0“ 两个输入。 - 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 - 2020年11月26日后开通服务的账号仅支持输入“3.0”。

参数名称	必选	类型	描述
			- 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0

3. 输出参数

参数名称	类型	描述
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建人员库成功示例

创建人员库。

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=腾讯深圳员工库
&FaceModelVersion=3.0
&GroupId=TencentShenZhenEmployee
&Tag=不含实习生
&GroupExDescriptions.0=事业群
&GroupExDescriptions.1=部门名
&GroupExDescriptions.2=组名
&<公共请求参数>
```

输出示例

```
{
"Response": {
"FaceModelVersion": "3.0",
"RequestId": "e53ee4ec-9099-4b35-a129-21dd4820ff85"
}
}
```

示例2 创建人员库错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=腾讯深圳员工库
&GroupId=腾讯深圳员工库
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdIllegal",
"Message": "人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。"
},
"RequestId": "8125dda4-2905-4e02-88bd-79a93a660ad2"
}
}
```

示例3 创建人员库ID重复错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=腾讯深圳员工库
&GroupId=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdAlreadyExist",
"Message": "人员库ID已经存在。人员库ID不可重复。"
},
"RequestId": "76ec6e41-37d6-4ab9-abef-48ef0c6ab175"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。

错误码	描述
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。

错误码	描述
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

删除人员库

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

删除该人员库及包含的所有的人员。同时，人员对应的所有人脸信息将被删除。若某人员同时存在多个人员库中，该人员不会被删除，但属于该人员库中的自定义描述字段信息会被删除，属于其他人员库的自定义描述字段信息不受影响。

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：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人员库ID，取值为创建人员库接口中的GroupId。 示例值：21

3. 输出参数

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

4. 示例

示例1 删除人员库成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteGroup
&GroupId=ZhuYuanDormitoryNo1
&<公共请求参数>
```

输出示例

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

```
"RequestId": "2ff7181c-ef20-4f02-b4ed-13924b381368"
}
}
```

示例2 删除人员库失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteGroup
&GroupId=ZhuYuanDormitory
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "878b225f-d3a4-4aa0-a849-3df415ac61a9"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。

错误码	描述
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员库信息

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取人员库信息。

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：GetGroupInfo。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人员库 ID，取值为创建人员库接口中的GroupId。 示例值：32

3. 输出参数

参数名称	类型	描述
GroupName	String	人员库名称。 示例值：正式人员名单
GroupId	String	人员库ID。 示例值：31
GroupExDescriptions	Array of String	人员库自定义描述字段。 示例值：["年龄"]
Tag	String	人员库信息备注。 示例值：无实习生
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
CreationTimestamp	Integer	Group的创建时间和日期。CreationTimestamp 的值是自 Unix 纪元时间到Group创建时间的毫秒数。 示例值：1694139227378
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId ）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取人员库信息成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupInfo
&GroupId=31
&<公共请求参数>
```

输出示例

```
{
"Response": {
"GroupName": "人员库",
"GroupId": "31",
"GroupExDescriptions": [
"年龄"
],
"Tag": "不含实习生",
"FaceModelVersion": "3.0",
"CreationTimestamp": 1559530931613,
"RequestId": "6faa05d0-287b-464e-8755-cf3e8d0c703d"
}
}
```

示例2 获取人员库信息失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupInfo
&GroupId=200
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "0e77ad29-ad65-4901-9efc-b49a6e0a357b"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
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。
ResourceUnavailable.StopUsing	账号已停服。
ResourcesSoldOut.ChargeStatusException	计费状态异常。

版权所有：腾讯云计算（北京）有限责任公司

获取人员库列表

最近更新时间：2025-04-25 01:35:38

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取人员库列表。

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：2020-03-03。
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 示例值：0
Limit	否	Integer	返回数量，默认值为10，最大值为1000 示例值：10

3. 输出参数

参数名称	类型	描述
GroupInfos	Array of GroupInfo	返回的人员库信息
GroupNum	Integer	人员库总数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例-2

返回数量不能大于1000

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupList
&Offset=0
&Limit=1001
&<公共请求参数>
```

输出示例


```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.LimitExceed",
"Message": "返回数量不在合法范围内。"
},
"RequestId": "a6bb57de-773b-45e4-9e1c-c30ef48eba85"
}
}
```


示例2 获取人员库列表接口

获取人员库列表

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupList
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例


```
{
"Response": {
"GroupInfos": [
{
"GroupName": "腾讯深圳员工",
"GroupId": "TencentShenZhenEmployee",
"GroupExDescriptions": [
"事业群",
"部门名",
"组名"
],
"Tag": "不含实习生",
"FaceModelVersion": "3.0",
"CreationTimestamp": 1592210245686
},
{
"GroupName": "某某大学竹园宿舍楼1号楼",
"GroupId": "ZhuYuanDormitoryNo1",
"GroupExDescriptions": [
```

```
"学院名",
"专业",
"年级",
"学号"
],
"Tag": "全是女生哦",
"FaceModelVersion": "3.0",
"CreationTimestamp": 1594552117118
}
],
"GroupNum": 2,
"RequestId": "72102087-a18d-4225-9ae9-87ef49e9f63e"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。

错误码	描述
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。

错误码	描述
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

修改人员库

最近更新时间：2025-04-25 01:35:37

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

修改人员库名称、备注、自定义描述字段名称。

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：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人员库ID，取值为创建人员库接口中的GroupId。 示例值：32
GroupName	否	String	人员库名称。 示例值：人员库名单
GroupExDescriptionInfos.N	否	Array of GroupExDescriptionInfo	需要修改的人员库自定义描述字段，key-value。
Tag	否	String	人员库信息备注。 示例值：无实习生

3. 输出参数

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

4. 示例

示例1 修改人员库成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=ZhuYuanDormitoryNo1
&GroupName=某某大学竹园1号宿舍楼
&Tag=研究生入住
```

```
&GroupExDescriptionInfos.0.GroupExDescriptionIndex=0
&GroupExDescriptionInfos.0.GroupExDescription=学院
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "da051f16-0400-49d4-9832-2cfd3b51866f"
}
}
```

示例2 修改人员库名称、备注等信息的成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=TencentShenZhenEmployee
&GroupName=腾讯深圳在职人员库
&Tag=包含实习生
&GroupExDescriptionInfos.0.GroupExDescriptionIndex=1
&GroupExDescriptionInfos.0.GroupExDescription=部门
&GroupExDescriptionInfos.1.GroupExDescriptionIndex=2
&GroupExDescriptionInfos.1.GroupExDescription=项目组
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "98381738-509f-4749-8a66-dbf54247333d"
}
}
```

示例3 修改人员库字段重复错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=TencentShenZhenEmployee
&GroupExDescriptionInfos.0.GroupExDescriptionIndex=0
&GroupExDescriptionInfos.0.GroupExDescription=部门
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.DuplicatedGroupDescription",
"Message": "同一人员库中自定义描述字段不可重复。"
},
"RequestId": "4b483679-3f08-4a7c-b827-4a6332dc030f"
}
}
```

示例4 修改人员库名称重复错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=TencentShenZhenEmployee
&GroupName=某某大学竹园1号宿舍楼
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupNameAlreadyExist",
"Message": "人员库名称已经存在。人员库名称不可重复。"
},
"RequestId": "2dcce170-30cf-4078-9b3c-309362fbad21"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee

- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

创建人员

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

创建人员，添加人脸、姓名、性别及其他相关信息。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：2020-03-03。
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，取值为创建人员库接口中的GroupId。 示例值：78
PersonName	是	String	人员名称。[1，60]个字符，可修改，可重复。 示例值：韦小宝
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。 示例值：2001
Gender	否	Integer	性别。 - 取值范围：0代表未填写，1代表男性，2代表女性。 示例值：1
PersonExDescriptionInfos.N	否	Array of PersonExDescriptionInfo	人员描述字段内容，key-value。[0，60]个字符，可修改，可重复。
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。

参数名称	必选	类型	描述
			- jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 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。 示例值：http://test.image.myqcloud.com/testB.jpg
UniquePersonControl	否	Integer	此参数用于控制判断 Image 或 Url 中图片包含的人脸，是否在人员库中已有疑似的同一人。 - 如果判断为已有相同人在人员库中，则不会创建新的人员，返回疑似同一人的人员信息。 - 如果判断没有，则完成创建人员。 - 取值范围： 0: 不进行判断，无论是否有疑似同一人在库中均完成入库； 1:较低的同一人判断要求（百一误识别率）； 2: 一般的同一人判断要求（千一误识别率）； 3: 较高的同一人判断要求（万一误识别率）； 4: 很高的同一人判断要求（十万一误识别率）。 - 默认 0。 - 注： 要求越高，则疑似同一人的概率越小。不同要求对应的误识别率仅为参考值，您可以根据实际情况调整。 示例值：0
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
FaceId	String	人脸图片唯一标识。

参数名称	类型	描述
		示例值：3454816969590585885
FaceRect	FaceRect	检测出的人脸框的位置。 注意：此字段可能返回 null，表示取不到有效值。
SimilarPersonId	String	疑似同一人的PersonId。 - 当 UniquePersonControl 参数不为0且人员库中有疑似的同一人，此参数才有意义。 示例值：18
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建人员成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=ZhuYuanDormitoryNo1
&PersonName=Junly
&PersonId=2001
&Gender=1
&Url=http://test.image.myqcloud.com/testB.jpg
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=0
&PersonExDescriptionInfos.0.PersonExDescription=计算机学院
&PersonExDescriptionInfos.1.PersonExDescriptionIndex=1
&PersonExDescriptionInfos.1.PersonExDescription=软件工程
&PersonExDescriptionInfos.2.PersonExDescriptionIndex=2
&PersonExDescriptionInfos.2.PersonExDescription=15级
&PersonExDescriptionInfos.3.PersonExDescriptionIndex=3
&PersonExDescriptionInfos.3.PersonExDescription=3150808
&<公共请求参数>
```

输出示例

```
{
"Response": {
"FaceId": "3454816969590585885",
"SimilarPersonId": "",
"FaceRect": {
"X": 172,
"Y": 122,
"Width": 178,
"Height": 228
},
"FaceModelVersion": "3.0",
"RequestId": "738a2fb0-bd4b-4cf6-9c91-18d50978d76b"
```

```
}
}
```

示例2 创建人员时ID重复错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=ZhuYuanDormitoryNo1
&PersonName=Cheng
&PersonId=1001
&Gender=2
&Url=http://test.image.myqcloud.com/testD.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonIdAlreadyExist",
"Message": "人员ID已经存在。人员ID不可重复。"
},
"RequestId": "76dedef7-af27-4a20-9064-c5ef3133926d"
}
}
```

示例3 创建人员时库ID不存在错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=ShenZhenCitizen
&PersonName=Siccy
&PersonId=3001
&Gender=2
&Url=http://test.image.myqcloud.com/testC.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "dfa512fc-fd07-4bf1-a292-cb497b620857"
```

```
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.ImageFacedetectFailed	人脸检测失败。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UniquePersonControlIllegal	UniquePersonControl参数不合法。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

删除人员

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

删除该人员信息，此操作会导致所有人员库均删除此人员。同时，该人员的所有人脸信息将被删除。

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：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：3001

3. 输出参数

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

4. 示例

示例1 删除人员信息成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePerson
&PersonId=2001
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "4b609b7c-7a9a-4ea0-be8b-1fb638c2ceca"
```

```
}
}
```

示例2 删除人员信息失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePerson
&PersonId=3001
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "ResourceNotFound.ErrorPersonNotExisted",
"Message": "个体不存在。"
},
"RequestId": "4b9c1bdc-86eb-413f-8c74-15bcf4466cd5"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。

版权所有：腾讯云计算（北京）有限责任公司

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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员库删除人员

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

从某人员库中删除人员，此操作仅影响该人员库。若该人员仅存在于指定的人员库中，该人员将被删除，其所有的人脸信息也将被删除。

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：DeletePersonFromGroup。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：3001
GroupId	是	String	人员库ID，取值为创建人员库接口中的GroupId。 示例值：ZhuYuanDormitoryNo1

3. 输出参数

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

4. 示例

示例1 人员库删除人员成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePersonFromGroup
&PersonId=1001
&GroupId=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "f9b22bbc-c94d-4287-a2ab-fa68f7bbbaf3"
}
}
```

示例2 人员库删除人员失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePersonFromGroup
&PersonId=3001
&GroupId=ZhuYuanDormitoryNo1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.GroupPersonMapNotExist",
"Message": "组中不包含对应的人员Id。"
},
"RequestId": "b9d9641d-3de2-4a6f-a728-36a3b66e7bd0"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

版权所有：腾讯云计算（北京）有限责任公司

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.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。

错误码	描述
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员归属信息

最近更新时间：2025-04-25 01:35:38

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员的信息，包括加入的人员库、描述内容等。

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：GetPersonGroupInfo。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：1002
Offset	否	Integer	起始序号，默认值为0。 示例值：0
Limit	否	Integer	返回数量，默认值为10，最大值为100。 示例值：10

3. 输出参数

参数名称	类型	描述
PersonGroupInfos	Array of PersonGroupInfo	包含此人员的人员库及描述字段内容列表。
GroupNum	Integer	人员库总数量。 注意：此字段可能返回 null，表示取不到有效值。 示例值：30
FaceModelVersion	String	人脸识别服务所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取人员归属信息成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonGroupInfo
&PersonId=2001
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
"Response": {
"GroupNum": "1",
"PersonGroupInfos": [
{
"GroupId": "ZhuYuanDormitoryNo1",
"PersonExDescriptions": [
"计算机学院",
"软件工程",
"15级",
"3150108080"
]
}
],
"FaceModelVersion": "3.0",
"RequestId": "671f0a1d-2b35-47c4-b9d1-b18053f71a04"
}
}
```

示例2 获取人员归属信息失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonGroupInfo
&PersonId=1002
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonIdNotExist",
"Message": "人员ID不存在。"
},
```

```
"RequestId": "98330f25-eb2e-432a-a30c-3830774210c1"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。

错误码	描述
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员基础信息

最近更新时间：2025-04-25 01:35:38

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员的信息，包括姓名、性别、人脸等。

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：GetPersonBaseInfo。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，创建人员接口中的PersonId。 示例值：1002

3. 输出参数

参数名称	类型	描述
PersonName	String	人员名称。 示例值：韦小宝
Gender	Integer	人员性别。 - 取值范围：0代表未填写，1代表男性，2代表女性。 示例值：2
FaceIds	Array of String	包含的人脸 ID 列表。 示例值：["287364080202264488"]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取人员基础信息成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonBaseInfo
&PersonId=1001
```

```
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonName": "EvanLiao",
"Gender": 1,
"FaceIds": [
"2873640802022644880",
"2875186538564559728"
],
"RequestId": "9568a077-0710-40d2-9d6a-b9483d3f2051"
}
}
```

示例2 获取人员基础信息失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonBaseInfo
&PersonId=1002
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonIdNotExist",
"Message": "人员ID不存在。"
},
"RequestId": "98b4a0bc-802b-4764-9701-bc0c6c544395"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。

错误码	描述
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。

错误码	描述
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取指定人员列表

最近更新时间：2025-04-25 01:35:38

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员库中的人员列表。

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：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人员库ID，取值为创建人员库接口中的GroupId。 示例值：ZhuYuanDormitory
Offset	否	Integer	起始序号，默认值为0。 示例值：0
Limit	否	Integer	返回数量，默认值为10，最大值为1000。 示例值：10

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	该人员库的人员数量。 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
FaceNum	Integer	该人员库的人脸数量。 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
FaceModelVersion	String	人脸识别所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取人员列表成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonList
&GroupId=ZhuYuanDormitoryNo1
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonInfos": [
{
"PersonName": "Junly",
"PersonId": "2001",
"Gender": 1,
"PersonExDescriptions": [
"计算机学院",
"软件工程",
"15级",
"3150108080"
],
"FaceIds": [
"2877244861637985315"
],
"CreationTimestamp": 1594642823572
}
],
"PersonNum": 1,
"FaceNum": 1,
"FaceModelVersion": "3.0",
"RequestId": "337d1bb5-3f88-4681-9291-5194f478f0d1"
}
}
```

示例2 获取人员列表失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonList
&GroupId=ZhuYuanDormitory
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "b7c0cd81-d621-465f-8fd6-86a6b49e67be"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。

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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员列表长度

最近更新时间：2025-04-25 01:35:38

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员库中人员数量。

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：GetPersonListNum。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人员库ID，取值为创建人员库接口中的GroupId。 示例值：TencentShenZhenEmployee

3. 输出参数

参数名称	类型	描述
PersonNum	Integer	人员数量。 示例值：10
FaceNum	Integer	人脸数量。 示例值：10
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取人员库中人员数量成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonListNum
&GroupId=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonNum": 2,
"FaceNum": 2,
"RequestId": "18fb1e31-0d33-4371-9f41-b1fd59e0c781"
}
}
```

示例2 获取人员库中人员数量失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonListNum
&GroupId=Tencent
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "2865a458-4db1-46bd-ae52-b7507c9f4e62"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。

错误码	描述
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

修改人员描述信息

最近更新时间：2025-04-25 01:35:37

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

修改指定人员库人员描述内容。

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：ModifyPersonGroupInfo。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupId	是	String	人员库ID，取值为创建人员库接口中的GroupId。 示例值：TencentShenZhenEmployee
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：1001
PersonExDescriptionInfos.N	是	Array of PersonExDescriptionInfo	需要修改的人员描述字段内容，key-value。

3. 输出参数

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

4. 示例

示例1 修改人员描述信息成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonGroupInfo
&GroupId=ZhuYuanDormitoryNo1
&PersonId=2001
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=3
```

```
&PersonExDescriptionInfos.0.PersonExDescription=3150108080
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "77e11a2d-0fc0-4fba-ab25-016e998221c3"
}
}
```

示例2 修改人员描述信息失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonGroupInfo
&GroupId=TencentShenZhenEmployee
&PersonId=1001
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=1
&PersonExDescriptionInfos.0.PersonExDescription=人工智能实验室
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "f944657d-6260-4a2e-a4d8-9c1845f39e56"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee

- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。

错误码	描述
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

修改人员基础信息

最近更新时间：2025-04-25 01:35:37

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

修改人员信息，包括名称、性别等。人员名称和性别修改会同步到包含该人员的所有人员库。

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：ModifyPersonBaseInfo。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：2001
PersonName	否	String	需要修改的人员名称。 示例值：韦小宝
Gender	否	Integer	需要修改的人员性别，1代表男性，2代表女性。 示例值：1

3. 输出参数

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

4. 示例

示例1 修改人员基础信息成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonBaseInfo
&PersonId=2001
&PersonName=JunlyWang
&Gender=1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "6e2a0fb7-7c9e-45c7-abf1-338457d1def7"
}
}
```

示例2 修改人员基础信息失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonBaseInfo
&PersonId=9001
&PersonName=EvanLiao
&Gender=1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.GroupPersonMapExist",
"Message": "组中已包含对应的人员Id。"
},
"RequestId": "f944657d-6260-4a2e-a4d8-9c1845f39e56"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

复制人员

最近更新时间：2025-04-25 01:35:40

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

将已存在于某人员库的人员复制到其他人员库，该人员的描述信息不会被复制。单个人员最多只能同时存在100个人员库中。

- 注：若该人员创建时算法模型版本为2.0，复制到非2.0算法模型版本的Group中时，复制操作将会失败。

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：CopyPerson。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：1001
GroupIds.N	是	Array of String	待加入的人员库列表，数组元素取值为创建人员库接口中的GroupId。 示例值：["TencentShenZhenEmployee"]

3. 输出参数

参数名称	类型	描述
SucGroupNum	Integer	成功加入的人员库数量。 示例值：10
SucGroupIds	Array of String	成功加入的人员库列表。 示例值：["TencentShenZhenEmployee"]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 复制人员成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CopyPerson
&PersonId=2001
&GroupIds.0=TencentShenZhenEmployee
&GroupIds.1=ShenZhenCitizen
&<公共请求参数>
```

输出示例

```
{
"Response": {
"SucGroupNum": 2,
"SucGroupIds": [
"ShenZhenCitizen",
"TencentShenZhenEmployee"
],
"RequestId": "9841c008-e597-4356-a8d5-c47a246abd8d"
}
}
```

示例2 复制人员失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CopyPerson
&PersonId=1001
&GroupIds.0=TencentShenZhenEmployee
&GroupIds.1=ZhuYuanDormitoryNo1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.GroupPersonMapExist",
"Message": "组中已包含对应的人员Id。"
},
"RequestId": "8ecfa566-68a9-4eb5-a6da-95ba9be26947"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。

错误码	描述
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。

版权所有：腾讯云计算（北京）有限责任公司

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.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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

增加人脸

最近更新时间：2025-04-25 01:35:40

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

将一组人脸图片添加到一个人员中。一个人员最多允许包含 5 张图片。若该人员存在多个人员库中，所有人员库中该人员图片均会增加。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：CreateFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：1001
Images.N	否	Array of String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 人员人脸总数量不可超过5张。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Urls.N	否	Array of String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。所有格式的图片短边像素不小于64。 - 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。 - 人员人脸总数量不可超过5张。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 示例值：["http://test.image.myqcloud.com/testA.jpg"]
FaceMatchThreshold	否	Float	只有和该人员已有的人脸相似度超过FaceMatchThreshold值的人脸，才能增加人脸成功。 - 默认值60分。

参数名称	必选	类型	描述
			- 取值范围：[0,100] 。 示例值：60
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
SucFaceNum	Integer	加入成功的人脸数量。 示例值：1
SucFaceIds	Array of String	加入成功的人脸ID列表。 示例值：["2875186538564559728"]
RetCode	Array of Integer	每张人脸图片添加结果。 - 其他非 0 值代表算法服务异常，取值范围： -1101 代表未检测到人脸。 -1102 代表图片解码失败。 -1109 代表图片尺寸过大或者过小。 -1601代表不符合图片质量控制要求。 -1604 代表人脸相似度没有超过FaceMatchThreshold。 - RetCode的顺序和入参中 Images 或 Urls 的顺序一致。 示例值：[0]
SucIndexes	Array of Integer	加入成功的人脸索引。 - 索引顺序和入参中 Images 或 Urls 的顺序一致。 - 例如， Urls 中 有 3 个 url，第二个 url 失败，则 SucIndexes 值为 [0,2] 。 示例值：[0]
SucFaceRects	Array of FaceRect	加入成功的人脸框位置。顺序和入参中 Images 或 Urls 的顺序一致。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId ）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 增加人脸成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateFace
&PersonId=1001
&Urls.0=http://test.image.myqcloud.com/testA.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"SucFaceNum": 1,
"SucFaceIds": [
"2875186538564559728"
],
"RetCode": [
0
],
"SucIndexes": [
0
],
"SucFaceRects": [
{
"X": 135,
"Y": 42,
"Width": 98,
"Height": 136
}
],
"FaceModelVersion": "3.0",
"RequestId": "07d63403-a199-4bbe-b9e0-692356ac738d"
}
}
```

示例2 增加人脸失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateFace
&PersonId=1001
&Urls.0=http://test.image.myqcloud.com/testB.jpg
&Urls.1=http://test.image.myqcloud.com/testC.jpg
&Urls.2=http://test.image.myqcloud.com/testD.jpg
&Urls.3=http://test.image.myqcloud.com/testE.jpg
```

版权所有：腾讯云计算（北京）有限责任公司

```
&Urls.4=http://test.image.myqcloud.com/testF.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.UploadFaceNumExceed",
"Message": "一次最多上传四张人脸。"
},
"RequestId": "44506f79-2191-49bc-997b-748a566d781c"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。

错误码	描述
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

删除人脸

最近更新时间：2025-04-25 01:35:39

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

删除一个人员下的人脸图片。如果该人员只有一张人脸图片，则返回错误。

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：DeleteFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	人员ID，取值为创建人员接口中的PersonId。 示例值：5001
FaceIds.N	是	Array of String	待删除的人脸ID列表，数组元素取值为增加人脸接口返回的FaceId。 示例值：["2875186538564559728"]

3. 输出参数

参数名称	类型	描述
SucDeletedNum	Integer	删除成功的人脸数量. 示例值：1
SucFaceIds	Array of String	删除成功的人脸ID列表。 示例值：["2875186538564559728"]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除人脸成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteFace
&PersonId=1001
```

```
&FaceIds.0=2875186538564559728
&<公共请求参数>
```

输出示例

```
{
"Response": {
"SucDeletedNum": 1,
"SucFaceIds": [
"2875186538564559728"
],
"RequestId": "38f96439-d5ea-48ea-899b-44a284708f6b"
}
}
```

示例2 删除人脸失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteFace
&PersonId=5001
&FaceIds.0=2875186538564559728
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "ResourceNotFound.ErrorPersonNotExisted",
"Message": "个体不存在。"
},
"RequestId": "e2b603e5-51d0-4a8a-8319-2df026ae7518"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸搜索相关接口
人脸搜索

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，识别结果按照相似度从大到小排序。

支持一次性识别图片中的最多 10 张人脸，支持一次性跨 100 个人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

与人员搜索及人员搜索按库返回接口不同的是，本接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行验证，而人员搜索及人员搜索按库返回接口 会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个Person下有4张 Face，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使搜索更加准确。

本接口需与人员库管理相关接口结合使用。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

- 不可同时搜索不同算法模型版本（FaceModelVersion）的人员库。

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：SearchFaces。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限100个。 - 数组元素取值为创建人员库接口中的GroupId。 - 不可同时搜索不同算法模型版本（FaceModelVersion）的人员库。 示例值：["TencentShenZhenEmployee"]
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。

参数名称	必选	类型	描述
			- 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 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。 示例值：http://test.image.myqcloud.com/testA.jpg
MaxFaceNum	否	Integer	最多识别的人脸数目。 - 默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 - MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。 - 例如：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。 - 默认为34。 - 低于34的人脸图片无法被识别。 - 建议设置为80。 示例值：40
MaxPersonNum	否	Integer	单张被识别的人脸返回的最相似人员数量。 - 默认值为5，最大值为100。 - 例如，设MaxFaceNum为1，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。 示例值：5
NeedPersonInfo	否	Integer	是否返回人员具体信息。 - 取值范围：0 为关闭，1 为开启。 - 默认为 0。 - 其他非0非1值默认为0。 示例值：0
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制。 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况。 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况。 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况。 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题。 - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有超过FaceMatchThreshold值的结果才会返回。默认为0。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。

参数名称	必选	类型	描述
			- 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Results	Array of Result	识别结果。
FaceNum	Integer	搜索的人员库中包含的人脸数。 示例值：1
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸搜索成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchFaces
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=1
&MinFaceSize=40
&MaxPersonNum=5
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Results": [
{
"RetCode": 0,
"Candidates": [
{
"PersonId": "1001",
"FaceId": "2875093635484912302",
"PersonName": "Jerry",
"Gender": 0,
"Score": 100,
"PersonGroupInfos": [
```

```
{
"GroupId": "abc",
"PersonExDescriptions": [
"description1",
"description2"
]
}
]
}
],
"FaceRect": {
"X": 370,
"Y": 46,
"Width": 75,
"Height": 75
}
}
],
"FaceNum": 1,
"FaceModelVersion": "3.0",
"RequestId": "57b42b73-b978-45b9-8095-8f50e9642d35"
}
}
```


示例2 人脸搜索失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchFaces
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=11
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例


```
{
"Response": {
"Error": {
"Code": "FailedOperation.SearchFacesExceed",
"Message": "检索人脸个数超过限制。"
},
"RequestId": "f04ba2f3-532c-4207-9caa-6e818ded7fb9"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。

错误码	描述
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸搜索分库返回

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，按照**人员库的维度**以人员相似度从大到小顺序排列。

支持一次性识别图片中的最多 10 张人脸，支持跨人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

与人员搜索及人员搜索按库返回接口不同的是，本接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行验证，而人员搜索及人员搜索按库返回接口 会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个Person下有4张 Face，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使搜索更加准确。

本接口需与人员库管理相关接口结合使用。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

- 不可同时搜索不同算法模型版本（FaceModelVersion）的人员库。

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：SearchFacesReturnsByGroup。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限60个。 - 数组元素取值为创建人员库接口中的GroupId。 - 不可同时搜索不同算法模型版本（FaceModelVersion）的人员库。 示例值：["TencentShenZhenEmployee\n"]
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==

参数名称	必选	类型	描述
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 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。 示例值：http://test.image.myqcloud.com/testA.jpg
MaxFaceNum	否	Integer	最多识别的人脸数目。 - 默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 - MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。 - 例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。 - 默认为34。 - 低于34将影响搜索精度。 - 建议设置为80。 示例值：40
MaxPersonNumPerGroup	否	Integer	被检测到的人脸，对应最多返回的最相似人员数目。 - 默认值为5，最大值为10。 - 例，设MaxFaceNum为3，MaxPersonNumPerGroup为5，GroupIds长度为3，则最多可能返回3*5=45个人员。 示例值：5
NeedPersonInfo	否	Integer	是否返回人员具体信息。 - 取值范围：0 为关闭，1 为开启。 - 默认为 0。 - 其他非0非1值默认为0 示例值：0
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制。 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况。 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况。 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况。 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题。 - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有大于等于FaceMatchThreshold值的结果才会返回。 - 默认为0。 - 取值范围：[0.0,100.0) 。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋

参数名称	必选	类型	描述
			转识别支持则无法正确检测、识别图片中的人脸。 – 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
FaceNum	Integer	搜索的人员库中包含的人脸数。 示例值：1
ResultsReturnsByGroup	Array of ResultsReturnsByGroup	识别结果。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸搜索分库返回成功示例

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"MaxPersonNumPerGroup": "5",
"GroupIds": [
"TencentShenZhenEmployee"
],
"MaxFaceNum": "1",
"MinFaceSize": "40"
}
```

输出示例

```
{
"Response": {
"FaceNum": 1,
"ResultsReturnsByGroup": [
{
```

```
"GroupCandidates": [
{
"Candidates": [
{
"PersonGroupInfos": [
{
"GroupId": "1002",
"PersonExDescriptions": [
"年龄",
"性别"
]
}
],
"PersonId": "1002",
"Gender": 0,
"PersonName": "韦小宝",
"Score": 100,
"FaceId": "2875093635484912302"
}
],
"GroupId": "Group1"
}
],
"FaceRect": {
"Y": 46,
"X": 370,
"Height": 1,
"Width": 1
},
"RetCode": 0
}
],
"RequestId": "57b42b73-b978-45b9-8095-8f50e9642d35",
"FaceModelVersion": "3.0"
}
}
```


示例2 人脸搜索分库返回失败示例

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testB.jpg",
"MaxPersonNumPerGroup": "5",
```

```
"GroupIds": [
"TencentShenZhenEmployee"
],
"MaxFaceNum": "1",
"MinFaceSize": "40"
}
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDecodeFailed",
"Message": "图片解码失败。"
},
"RequestId": "0e77ad29-ad65-4901-9efc-b49a6e0a357b"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。

错误码	描述
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员搜索

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，按照相似度从大到小排列。

支持一次性识别图片中的最多 10 张人脸，支持一次性跨 100 个人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

本接口会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个 Person 下有4张 Face ，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员搜索（确定待识别的人脸图片是某人）更加准确。而人脸搜索及人脸搜索按库返回接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行搜索。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。
- 仅支持算法模型版本（FaceModelVersion）为3.0的人员库。

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：SearchPersons。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限100个。数组元素取值为创建人员库接口中的GroupId。 示例值：["TencentShenZhenEmployee"]
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 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。

参数名称	必选	类型	描述
			- 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。 示例值：http://test.image.myqcloud.com/testA.jpg
MaxFaceNum	否	Integer	最多识别的人脸数目。 - 默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 - MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。 - 例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。 - 默认为34。 - 低于34将影响搜索精度。 - 建议设置为80。 示例值：40
MaxPersonNum	否	Integer	单张被识别的人脸返回的最相似人员数量。 - 默认值为5，最大值为100。 - 例，设MaxFaceNum为1，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。 示例值：3
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制。 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况。 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况。 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况。 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题。 - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有大于等于FaceMatchThreshold值的结果才会返回。 - 默认为0。 - 取值范围：[0.0,100.0) 。 示例值：0
NeedPersonInfo	否	Integer	是否返回人员具体信息。 - 取值范围：0 为关闭，1 为开启。 - 默认为 0。 - 其他非0非1值默认为0 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Results	Array of Result	识别结果。
PersonNum	Integer	搜索的人员库中包含的人员数。若输入图片中所有人脸均不符合质量要求，则返回0。 示例值：5
FaceModelVersion	String	人脸识别所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人员搜索成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersons
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=2
&MinFaceSize=40
&MaxPersonNum=3
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Results": [
{
"RetCode": 0,
"Candidates": [
{
"PersonId": "person1",
"FaceId": "",
"Score": 100,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
"PersonId": "person2",
"FaceId": "",
"Score": 60,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
```

```
},
{
"PersonId": "person3",
"FaceId": "",
"Score": 50,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
}
],
"FaceRect": {
"X": 139,
"Y": 59,
"Width": 124,
"Height": 162
}
},
{
"RetCode": 0,
"Candidates": [
{
"PersonId": "person2",
"FaceId": "",
"Score": 100,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
"PersonId": "person1",
"FaceId": "",
"Score": 60,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
"PersonId": "person3",
"FaceId": "",
"Score": 20,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
}
],
"FaceRect": {
"X": 328,
"Y": 70,
"Width": 119,
"Height": 162
```

```
}
}
],
"PersonNum": 5,
"FaceModelVersion": "3.0",
"RequestId": "c4608852-ff60-4a01-8dc3-367f6046baaf"
}
}
```

示例2 人员搜索失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersons
&Url=http://test.image.myqcloud.com/testA
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDownloadError",
"Message": "图片下载错误。"
},
"RequestId": "527ecffe-4c6a-47c9-8217-4dd2e3f018da"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.ImageEmptyError	图片为空。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。

错误码	描述
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UrlIllegal	URL格式不合法。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.GetAuthInfoError	获取认证信息失败。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员搜索按库返回

最近更新时间：2025-04-25 01:35:36

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，按照**人员库的维度**以人员相似度从大到小顺序排列。

支持一次性识别图片中的最多 10 张人脸，支持跨人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

本接口会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个 Person 下有4张 Face ，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员搜索（确定待识别的人脸图片是某人）更加准确。而人脸搜索及人脸搜索按库返回接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行搜索。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。
- 仅支持算法模型版本（FaceModelVersion）为3.0的人员库。

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：SearchPersonsReturnsByGroup。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限60个。数组元素取值为创建人员库接口中的GroupId。 示例值：["TencentShenZhenEmployee"]
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 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。

参数名称	必选	类型	描述
			- 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testB.jpg
MaxFaceNum	否	Integer	最多识别的人脸数目。 - 默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 - MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。 - 例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。 - 默认为34。 - 低于34将影响搜索精度。 - 建议设置为80。 示例值：80
MaxPersonNumPerGroup	否	Integer	被检测到的人脸，对应最多返回的最相似人员数目。 - 默认值为5，最大值为10。 - 例，设MaxFaceNum为3，MaxPersonNumPerGroup为5，GroupIds长度为3，则最多可能返回3*5*3=45个人员。 示例值：5
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制。 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况。 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况。 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况。 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题。 - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有超过FaceMatchThreshold值的结果才会返回。默认为0。 示例值：0
NeedPersonInfo	否	Integer	是否返回人员具体信息。 - 取值范围：0 为关闭，1 为开启。 - 默认为 0。 - 其他非0非1值默认为0。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
PersonNum	Integer	搜索的人员库中包含的人员数。若输入图片中所有人脸均不符合质量要求，则返回0。 示例值：10
ResultsReturnsByGroup	Array of ResultsReturnsByGroup	识别结果。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人员搜索按库返回成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersonsReturnsByGroup
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=1
&MinFaceSize=40
&MaxPersonNumPerGroup=5
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例


```
{
"Response": {
"ResultsReturnsByGroup": [
{
"RetCode": 0,
"FaceRect": {
"X": 375,
"Y": 37,
"Width": 63,
"Height": 82
},
"GroupCandidates": [
{
"GroupId": "TencentShenZhenEmployee",
"Candidates": [
{
"PersonId": "testtest3",
"FaceId": "",
"Score": 100,
"PersonName": null,
```

```
"Gender": null,
"PersonGroupInfos": null
}
]
}
]
}
],
"PersonNum": 1,
"FaceModelVersion": "3.0",
"RequestId": "c8a06ec5-ecb4-4dd7-a8c8-ce5e675495c2"
}
}
```

示例2 人员搜索按库返回失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersonsReturnsByGroup
&Url=http://test.image.myqcloud.com/testB.jpg
&MaxFaceNum=1
&MinFaceSize=40
&MaxPersonNumPerGroup=5
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDownloadError",
"Message": "图片下载错误。"
},
"RequestId": "527ecffe-4c6a-47c9-8217-4dd2e3f018da"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdNotExists	对应的人员库ID在库中不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UrlIllegal	URL格式不合法。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸验证相关接口
人员验证

最近更新时间：2025-04-25 01:35:35

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

给定一张人脸图片和一个 PersonId，判断图片中的人和 PersonId 对应的人是否为同一人。PersonId 请参考人员库管理相关接口。
本接口会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个Person下有4张 Face，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员验证（确定待识别的人脸图片是某人员）更加准确。

和人脸比对相关接口不同的是，人脸验证相关接口用于判断 “此人是否是此人”，“此人”的信息已存于人员库中，“此人”可能存在多张人脸图片；而人脸比对相关接口用于判断两张人脸的相似度。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。
- 仅支持算法模型版本（FaceModelVersion）为3.0的人员库。

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：VerifyPerson。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	待验证的人员ID。人员ID具体信息请参考人员库管理相关接口。 示例值：2001
Image	否	String	图片 base64 数据。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 图片的 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。

参数名称	必选	类型	描述
			- 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制。 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况。 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况。 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况。 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题。 - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	给定的人脸照片与 PersonId 对应的相似度。若 PersonId 下有多张人脸（Face），会融合多张人脸信息进行验证。 示例值：100
IsMatch	Boolean	是否为同一人的判断。 示例值：false
FaceModelVersion	String	人脸识别所用的算法模型版本，是该 Person 所在的人员库的算法模型版本。 - 在创建人员库时设置，详情可参考算法模型版本 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸验证成功示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=VerifyPerson
&Url=http://test.image.myqcloud.com/testA.jpg
&PersonId=1001
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Score": 100,
"IsMatch": true,
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

示例2 人脸验证失败示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=VerifyPerson
&Url=http://test.image.myqcloud.com/testB.jpg
&PersonId=1001
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDecodeFailed",
"Message": "图片解码失败。"
},
"RequestId": "0e77ad29-ad65-4901-9efc-b49a6e0a357b"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee

- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
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.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded	超过配额限制。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.Freeze	账号已被冻结。

错误码	描述
ResourceUnavailable.GetAuthInfoError	获取认证信息失败。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸验证

最近更新时间：2025-04-25 01:35:35

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

给定一张人脸图片和一个 PersonId，判断图片中的人和 PersonId 对应的人是否为同一人。PersonId 请参考人员库管理相关接口。

与人脸比对接口不同的是，人脸验证用于判断 “此人是否是此人”，“此人”的信息已存于人员库中，“此人”可能存在多张人脸图片；而人脸比对用于判断两张人脸的相似度。

与人员验证接口不同的是，人脸验证将该人员（Person）下的每个人脸（Face）都作为单独个体进行验证，而人员验证会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个 Person下有4张 Face，人员验证接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员验证（确定待识别的人脸图片是某人员）更加准确。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：VerifyFace。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
PersonId	是	String	待验证的人员ID。人员ID具体信息请参考人员库管理相关接口。 示例值：1001
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 所有格式的图片短边像素不小于64。 - 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。 - 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。

参数名称	必选	类型	描述
			- 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
QualityControl	否	Integer	图片质量控制。 - 取值范围： 0: 不进行控制。 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况。 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况。 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况。 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题。 - 默认 0。 - 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	给定的人脸图片与 PersonId 对应人脸的相似度。 - 若 PersonId 下有多张人脸（Face），返回相似度最大的分数。 - 不同算法版本返回的相似度分数不同。 - 若需要验证两张图片中人脸是否为同一人，3.0版本误识率千分之一对应分数为40分，误识率万分之一对应分数为50分，误识率十万分之一对应分数为60分。一般超过50分则可认定为同一人。 - 2.0版本误识率千分之一对应分数为70分，误识率万分之一对应分数为80分，误识率十万分之一对应分数为90分。一般超过80分则可认定为同一人。 示例值：100
IsMatch	Boolean	是否为同一人判断，固定阈值分数为60分，若想更灵活地调整阈值可取Score参数返回进行判断。 示例值：1
FaceModelVersion	String	人脸识别所用的算法模型版本，是该 Person 所在的人员库的算法模型版本。 - 在创建人员库时设置，详情可参考算法模型版本 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸验证成功示例

输入示例

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

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"PersonId": "1001"
}
```

输出示例

```
{
"Response": {
"Score": 100,
"IsMatch": true,
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

示例2 人脸验证失败示例

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testB.jpg",
"PersonId": "1001"
}
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDecodeFailed",
"Message": "图片解码失败。"
},
"RequestId": "0e77ad29-ad65-4901-9efc-b49a6e0a357b"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。

错误码	描述
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸静态活体检测（高精度版）相关接口
人脸静态活体检测（高精度版）

最近更新时间：2025-04-25 01:35:35

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

人脸静态活体检测（高精度版）可用于对用户上传的静态图片进行防翻拍活体检测，以判断是否是翻拍图片。

相比现有静态活体检测服务，高精度版在维持高真人通过率的前提下，增强了对高清屏幕、裁剪纸片、3D面具等攻击的防御能力，攻击拦截率约为业内同类型产品形态4-5倍。同时支持多场景人脸核验，满足移动端、PC端各类型场景的图片活体检验需求，适用于各个行业不同的活体检验应用。

人脸静态活体检测（高精度版）接口于2022年8月1日 00:00起正式开始计费，采取后付费按量计费模式，详见计费概述。

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：DetectLiveFaceAccurate。
Version	是	String	公共参数，本接口取值：2020-03-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 图片的宽高比请接近 3:4，手机拍摄比例最佳。 - 人脸尺寸大于100X100像素。 - 图片格式支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - Url、Image必须提供一个，如果都提供，只使用 Url。 - 图片的宽高比请接近 3:4，手机拍摄比例最佳。 - 人脸尺寸大于100X100像素。 - 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。 - 图片格式支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。目前入参支持“3.0“。 示例值：3.0

3. 输出参数

参数名称	类型	描述
Score	Float	活体打分。 - 取值范围 [0,100]。 - 根据活体分数对应的阈值区间来判断是否为翻拍。 - 目前阈值可分为[5,10,40,70,90]，其中推荐阈值为40。 示例值：99
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸静态活体检测（高精度版）成功示例

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg"
}
```

输出示例

```
{
"Response": {
"Score": 99,
"FaceModelVersion": "3.0",
"RequestId": "048a5eb3-f7a9-4c1f-a982-00e90b6b892c"
}
}
```

示例2 人脸静态活体检测（高精度版）失败示例

输入示例

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

{
```

```
"Url": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDecodeFailed",
"Message": "图片解码失败。"
},
"RequestId": "0e77ad29-ad65-4901-9efc-b49a6e0a357b"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。

错误码	描述
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

数据结构

最近更新时间：2024-11-14 01:41:44

AttributeItem

人脸属性信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Type	Integer	属性值。 示例值：1
Probability	Float	Type识别概率值。 - 取值范围：[0,1]，代表判断正确的概率。 示例值：90

Candidate

识别出的最相似候选人

被如下接口引用：SearchFaces, SearchFacesReturnsByGroup, SearchPersons, SearchPersonsReturnsByGroup。

名称	类型	描述
PersonId	String	人员ID。 示例值：person1
FaceId	String	人脸ID。 - 仅在SearchFaces/SearchFacesReturnsByGroup接口返回时有效。 - 人员搜索类接口采用融合特征方式检索，该字段无意义。 示例值：3820314501007076807
Score	Float	候选者的匹配得分。 - 1万大小人脸底库下，误识率百分之一对应分数为70分，误识率千分之一对应分数为80分，误识率万分之一对应分数为90分。 - 10万大小人脸底库下，误识率百分之一对应分数为80分，误识率千分之一对应分数为90分，误识率万分之一对应分数为100分。 - 30万大小人脸底库下，误识率百分之一对应分数为85分，误识率千分之一对应分数为95分。 - 一般80分左右可适用大部分场景，建议分数不要超过90分。您可以根据实际情况选择合适的分数。 示例值：50
PersonName	String	人员名称。 注意：此字段可能返回 null，表示取不到有效值。 示例值：韦小宝
Gender	Integer	人员性别。 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
PersonGroupInfos	Array of PersonGroupInfo	包含此人员的人员库及描述字段内容列表。 注意：此字段可能返回 null，表示取不到有效值。

DenseFaceShape

稠密关键点详细信息

被如下接口引用：AnalyzeDenseLandmarks。

名称	类型	描述
X	Integer	人脸框左上角横坐标。 示例值：198
Y	Integer	人脸框左上角纵坐标。 示例值：254
Width	Integer	人脸框宽度。 示例值：924
Height	Integer	人脸框高度。 示例值：809
LeftEye	Array of Point	描述左侧眼睛轮廓的 XX 点。
RightEye	Array of Point	描述右侧眼睛轮廓的 XX 点。
LeftEyeBrow	Array of Point	描述左侧眉毛轮廓的 XX 点。
RightEyeBrow	Array of Point	描述右侧眉毛轮廓的 XX 点。
MouthOutside	Array of Point	描述外嘴巴轮廓的 XX 点， 从左侧开始逆时针返回。
MouthInside	Array of Point	描述内嘴巴轮廓的 XX 点，从左侧开始逆时针返回。
Nose	Array of Point	描述鼻子轮廓的 XX 点。
LeftPupil	Array of Point	左瞳孔轮廓的 XX 个点。
RightPupil	Array of Point	右瞳孔轮廓的 XX 个点。
CentralAxis	Array of Point	中轴线轮廓的 XX 个点。
Chin	Array of Point	下巴轮廓的 XX 个点。
LeftEyeBags	Array of Point	左眼袋的 XX 个点。
RightEyeBags	Array of Point	右眼袋的 XX 个点。
Forehead	Array of Point	额头的 XX 个点。

Eye

眼睛信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Glass	AttributeItem	识别是否佩戴眼镜。 - AttributeItem对应的Type为 —— 0：无眼镜，1：普通眼镜，2：墨镜
EyeOpen	AttributeItem	识别眼睛的睁开、闭合状态。 - AttributeItem对应的Type为 —— 0：睁开，1：闭眼
EyelidType	AttributeItem	识别是否双眼皮。 - AttributeItem对应的Type为 —— 0：无，1：有。

名称	类型	描述
EyeSize	AttributeItem	眼睛大小。 - AttributeItem对应的Type为 —— 0：小眼睛，1：普通眼睛，2：大眼睛。

Eyebrow

眉毛信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
EyebrowDensity	AttributeItem	眉毛浓密。 - AttributeItem对应的Type为 —— 0：淡眉，1：浓眉。
EyebrowCurve	AttributeItem	眉毛弯曲。 - AttributeItem对应的Type为 —— 0：不弯，1：弯眉。
EyebrowLength	AttributeItem	眉毛长短。 - AttributeItem对应的Type为 —— 0：短眉毛，1：长眉毛。

FaceAttributesInfo

人脸属性信息，包含性别(gender)、年龄(age)、表情(expression)、
魅力(beauty)、眼镜(glass)、口罩（ mask ）、头发（ hair ）和姿态 (pitch，roll，yaw)。只有当 NeedFaceAttributes 设为 1 时才返回有效信息，最多返回面积最大的 5 张人脸属性信息，超过 5 张人脸（第 6 张及以后的人脸）的 FaceAttributesInfo 不具备参考意义。

被如下接口引用：DetectFace。

名称	类型	描述
Gender	Integer	性别。 - 取值说明：[0~49]为女性，[50，100]为男性，越接近0和100表示置信度越高。 - NeedFaceAttributes 不为 1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：40
Age	Integer	年龄 。 - 取值范围：[0~100]。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：39
Expression	Integer	微笑。 - 取值说明：[0(normal，正常)~100(laugh，大笑)]。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：13
Glass	Boolean	是否有眼镜。 - 取值范围： [true,false]。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：true
Pitch	Integer	上下偏移。 - 取值范围：[-30,30]，单位角度。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 - 建议：人脸入库选择[-10,10]的图片。 示例值：13
Yaw	Integer	左右偏移。 - 取值范围：[-30,30]，单位角度。

名称	类型	描述
		- NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 - 建议：人脸入库选择[-10,10]的图片。 示例值：21
Roll	Integer	平面旋转。 - 取值范围：[-180,180]，单位角度。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 - 建议：人脸入库选择[-20,20]的图片。 示例值：54
Beauty	Integer	魅力。 - 取值范围：[0~100]。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：50
Hat	Boolean	是否有帽子。 - 取值范围： [true,false]。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。 示例值：true
Mask	Boolean	是否有口罩。 - 取值范围： [true,false]。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。 示例值：false
Hair	FaceHairAttributesInfo	头发信息，包含头发长度（length）、有无刘海（bang）、头发颜色（color）。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。
EyeOpen	Boolean	双眼是否睁开。 - 取值范围： [true,false]。 - 只要有超过一只眼睛闭眼，就返回false。 - NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。 示例值：true

FaceDetailAttributesInfo

人脸属性信息，根据 FaceAttributesType 输入的类型，返回年龄（Age）、颜值（Beauty）
情绪（Emotion）、眼睛信息（Eye）、眉毛（Eyebrow）、性别（Gender）
头发（Hair）、帽子（Hat）、姿态（Headpose）、口罩（Mask）、嘴巴（Mouth）、胡子（Moustache）
鼻子（Nose）、脸型（Shape）、肤色（Skin）、微笑（Smile）等人脸属性信息。
若 FaceAttributesType 没有输入相关类型，则FaceDetaiAttributesInfo返回的细项不具备参考意义。

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Age	Integer	年龄。 - 取值范围： [0,65]，其中65代表“65岁及以上”。 - FaceAttributesType 不含Age 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：30
Beauty	Integer	美丑打分。 - 取值范围：[0,100]。 - FaceAttributesType 不含 Beauty 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	描述
Emotion	AttributeItem	情绪，可识别自然、高兴、惊讶、生气、悲伤、厌恶、害怕。 - AttributeItem对应的Type为 —— 0：自然，1：高兴，2：惊讶，3：生气，4：悲伤，5：厌恶，6：害怕。 - FaceAttributesType 不含Emotion 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Eye	Eye	眼睛相关信息，可识别是否戴眼镜、是否闭眼、是否双眼皮和眼睛大小。 - FaceAttributesType 不含Eye 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Eyebrow	Eyebrow	眉毛相关信息，可识别眉毛浓密、弯曲、长短信息。 FaceAttributesType 不含Eyebrow 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Gender	AttributeItem	性别信息。 - AttributeItem对应的Type为 —— 0：男性，1：女性。 - FaceAttributesType 不含Gender 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Hair	Hair	头发信息，包含头发长度、有无刘海、头发颜色。 - FaceAttributesType 不含Hair 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Hat	Hat	帽子信息，可识别是否佩戴帽子、帽子款式、帽子颜色。 - FaceAttributesType 不含Hat 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
HeadPose	HeadPose	姿态信息，包含人脸的上下偏移、左右偏移、平面旋转信息。 - FaceAttributesType 不含Headpose 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Mask	AttributeItem	口罩佩戴信息。 - AttributeItem对应的Type为 —— 0: 无口罩， 1: 有口罩不遮脸，2: 有口罩遮下巴，3: 有口罩遮嘴，4: 正确佩戴口罩。 - FaceAttributesType 不含Mask 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Mouth	Mouth	嘴巴信息，可识别是否张嘴、嘴唇厚度。 - FaceAttributesType 不含 Mouth 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Moustache	AttributeItem	胡子信息。 - AttributeItem对应的Type为 —— 0：无胡子，1：有胡子。 - FaceAttributesType 不含 Moustache 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Nose	AttributeItem	鼻子信息。 - AttributeItem对应的Type为 —— 0：朝天鼻，1：鹰钩鼻，2：普通，3：圆鼻头 - FaceAttributesType 不含 Nose 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Shape	AttributeItem	脸型信息。 - AttributeItem对应的Type为 —— 0：方脸，1：三角脸，2：鹅蛋脸，3：心形脸，4：圆脸。 - FaceAttributesType 不含 Shape 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Skin	AttributeItem	肤色信息。 - AttributeItem对应的Type为 —— 0：黄色皮肤，1：棕色皮肤，2：黑色皮肤，3：白色皮肤。 - FaceAttributesType 不含 Skin 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Smile	Integer	微笑程度 - 取值范围：[0,100]。 - FaceAttributesType 不含 Smile 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 示例值：20

FaceDetailInfo

人脸信息列表。

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
FaceRect	FaceRect	检测出的人脸框位置。
FaceDetailAttributesInfo	FaceDetailAttributesInfo	人脸属性信息。 - 根据 FaceAttributesType 输入的类型，返回年龄（Age）、颜值（Beauty）、情绪（Emotion）、眼睛信息（Eye）、眉毛（Eyebrow）、性别（Gender）、头发（Hair）、帽子（Hat）、姿态（Headpose）、口罩（Mask）、嘴巴（Mouth）、胡子（Moustache）、鼻子（Nose）、脸型（Shape）、肤色（Skin）、微笑（Smile）等人脸属性信息。 - 若 FaceAttributesType 没有输入相关类型，则 FaceDetaiAttributesInfo返回的细项不具备参考意义。

FaceHairAttributesInfo

人脸属性中的发型信息。

被如下接口引用：DetectFace。

名称	类型	描述
Length	Integer	取值范围：0：光头，1：短发，2：中发，3：长发，4：绑发。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Bang	Integer	取值范围：0：有刘海，1：无刘海。 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Color	Integer	取值范围：0：黑色，1：金色，2：棕色，3：灰白色。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

FaceInfo

人脸信息列表。

被如下接口引用：DetectFace。

名称	类型	描述
X	Integer	人脸框左上角横坐标。 - 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 - 若需截取完整人脸，可以在完整分completeness满足需求的情况下，将负值坐标取0。 示例值：35
Y	Integer	人脸框左上角纵坐标。 - 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 - 若需截取完整人脸，可以在完整分completeness满足需求的情况下，将负值坐标取0。 示例值：61
Width	Integer	人脸框宽度。 示例值：10
Height	Integer	人脸框高度。 示例值：8

名称	类型	描述
FaceAttributesInfo	FaceAttributesInfo	人脸属性信息。 - 包含性别(gender)、年龄(age)、表情(expression)、魅力(beauty)、眼镜(glass)、口罩（mask）、头发（hair）和姿态 (pitch，roll，yaw)。 - 只有当 NeedFaceAttributes 设为 1 时才返回有效信息。
FaceQualityInfo	FaceQualityInfo	人脸质量信息。 - 包含质量分（score）、模糊分（sharpness）、光照分（brightness）、遮挡分（completeness）。 - 只有当NeedFaceDetection设为1时才返回有效信息。 注意：此字段可能返回 null，表示取不到有效值。

FaceQualityCompleteness

五官遮挡分，评价眉毛（Eyebrow）、眼睛（Eye）、鼻子（Nose）、脸颊（Cheek）、嘴巴（Mouth）、下巴（Chin）的被遮挡程度。

被如下接口引用：DetectFace。

名称	类型	描述
Eyebrow	Integer	眉毛的遮挡分数。 - 取值范围：[0,100]，分数越高遮挡越少。 - 参考范围：[0,80]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。 示例值：70
Eye	Integer	眼睛的遮挡分数。 - 取值范围：[0,100],分数越高遮挡越少。 - 参考范围：[0,80]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。 示例值：60
Nose	Integer	鼻子的遮挡分数。 - 取值范围：[0,100],分数越高遮挡越少。 - 参考范围：[0,60]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。 示例值：30
Cheek	Integer	脸颊的遮挡分数。 - 取值范围：[0,100],分数越高遮挡越少。 - 参考范围：[0,70]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
Mouth	Integer	嘴巴的遮挡分数。 - 取值范围：[0,100],分数越高遮挡越少。 - 参考范围：[0,50]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
Chin	Integer	下巴的遮挡分数。 - 取值范围：[0,100],分数越高遮挡越少。 - 参考范围：[0,70]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。 示例值：80

FaceQualityInfo

版权所有：腾讯云计算（北京）有限责任公司

人脸质量信息，包含质量分（score）、模糊分（sharpness）、光照分（brightness）、遮挡分（completeness）。只有当NeedFaceDetection设为1时才返回有效信息。

被如下接口引用：DetectFace。

名称	类型	描述
Score	Integer	质量分。 - 取值范围：[0,100]，综合评价图像质量是否适合人脸识别，分数越高质量越好。 - 正常情况，只需要使用Score作为质量分总体的判断标准即可。Sharpness、Brightness、Completeness等细项分仅供参考。 - 参考范围：[0,40]较差，[40,60] 一般，[60,80]较好，[80,100]很好。 - 建议：人脸入库选取70以上的图片。 注意：此字段可能返回 null，表示取不到有效值。 示例值：80
Sharpness	Integer	清晰分。 - 取值范围：[0,100]，评价图片清晰程度，分数越高越清晰。 - 参考范围：[0,40]特别模糊，[40,60]模糊，[60,80]一般，[80,100]清晰。 - 建议：人脸入库选取80以上的图片。 注意：此字段可能返回 null，表示取不到有效值。 示例值：100
Brightness	Integer	光照分。 - 取值范围：[0,100]，评价图片光照程度，分数越高越亮。 - 参考范围： [0,30]偏暗，[30,70]光照正常，[70,100]偏亮。 - 建议：人脸入库选取[30,70]的图片。 注意：此字段可能返回 null，表示取不到有效值。 示例值：30
Completeness	FaceQualityCompleteness	五官遮挡分，评价眉毛（Eyebrow）、眼睛（Eye）、鼻子（Nose）、脸颊（Cheek）、嘴巴（Mouth）、下巴（Chin）的被遮挡程度。 注意：此字段可能返回 null，表示取不到有效值。

FaceRect

检测出的人脸框的位置

被如下接口引用：CreateFace, CreatePerson, DetectFaceAttributes, SearchFaces, SearchFacesReturnsByGroup, SearchPersons, SearchPersonsReturnsByGroup。

名称	类型	必选	描述
X	Integer	否	人脸框左上角横坐标。 - 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 - 若需截取完整人脸，可以在完整分completess满足需求的情况下，将负值坐标取0。 示例值：253
Y	Integer	否	人脸框左上角纵坐标。 - 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 - 若需截取完整人脸，可以在完整分completess满足需求的情况下，将负值坐标取0。 示例值：414
Width	Integer	否	人脸宽度。 示例值：180
Height	Integer	否	人脸高度。 示例值：90

FaceShape

五官定位（人脸关键点）具体信息。

被如下接口引用：AnalyzeFace。

名称	类型	描述
FaceProfile	Array of Point	描述脸型轮廓的 21 点。
LeftEye	Array of Point	描述左侧眼睛轮廓的 8 点。
RightEye	Array of Point	描述右侧眼睛轮廓的 8 点。
LeftEyeBrow	Array of Point	描述左侧眉毛轮廓的 8 点。
RightEyeBrow	Array of Point	描述右侧眉毛轮廓的 8 点。
Mouth	Array of Point	描述嘴巴轮廓的 22 点。
Nose	Array of Point	描述鼻子轮廓的 13 点。
LeftPupil	Array of Point	左瞳孔轮廓的 1 个点。
RightPupil	Array of Point	右瞳孔轮廓的 1 个点。

GroupCandidate

分组识别结果Item

被如下接口引用：SearchFacesReturnsByGroup, SearchPersonsReturnsByGroup。

名称	类型	描述
GroupId	String	人员库ID 。 示例值：1001
Candidates	Array of Candidate	识别出的最相似候选人。

GroupExDescriptionInfo

需要修改的人员库自定义描述字段key-value

被如下接口引用：ModifyGroup。

名称	类型	必选	描述
GroupExDescriptionIndex	Integer	是	人员库自定义描述字段Index，从0开始。
GroupExDescription	String	是	需要更新的人员库自定义描述字段内容。 示例值：年龄

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	人员库名称。 示例值：人员库

名称	类型	描述
GroupId	String	人员库ID。 示例值：13
GroupExDescriptions	Array of String	人员库自定义描述字段。 注意：此字段可能返回 null，表示取不到有效值。 示例值：["年龄"]
Tag	String	人员库信息备注。 注意：此字段可能返回 null，表示取不到有效值。 示例值：无实习生
FaceModelVersion	String	人脸识别所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。 示例值：3.0
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) 00:00:00。 - 有关更多信息，请参阅 Unix 时间。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1694139227378

Hair

头发信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Length	AttributeItem	头发长度信息。 AttributeItem对应的Type为 —— 0：光头，1：短发，2：中发，3：长发，4：绑发。
Bang	AttributeItem	刘海信息。 AttributeItem对应的Type为 —— 0：无刘海，1：有刘海。
Color	AttributeItem	头发颜色信息。 AttributeItem对应的Type为 —— 0：黑色，1：金色，2：棕色，3：灰白色。

Hat

帽子信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Style	AttributeItem	帽子佩戴状态信息。 AttributeItem对应的Type为 —— 0：不戴帽子，1：普通帽子，2：头盔，3：保安帽。
Color	AttributeItem	帽子颜色。 AttributeItem对应的Type为 —— 0：不戴帽子，1：红色系，2：黄色系，3：蓝色系，4：黑色系，5：灰白色系，6：混色系。

HeadPose

姿态信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Pitch	Integer	上下偏移[-30,30]。 示例值：28
Yaw	Integer	左右偏移[-30,30]。 示例值：10
Roll	Integer	平面旋转[-180,180]。 示例值：23

Mouth

嘴巴信息。

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
MouthOpen	AttributeItem	是否张嘴信息。 AttributeItem对应的Type为 —— 0：不张嘴，1：张嘴。

PersonExDescriptionInfo

需要修改的人员描述字段内容，key-value

被如下接口引用：CreatePerson, ModifyPersonGroupInfo。

名称	类型	必选	描述
PersonExDescriptionIndex	Integer	是	人员描述字段Index，从0开始。 示例值：0
PersonExDescription	String	是	需要更新的人员描述字段内容。 示例值：年龄

PersonGroupInfo

包含此人员的人员库及描述字段内容列表

被如下接口引用：GetPersonGroupInfo, SearchFaces, SearchFacesReturnsByGroup, SearchPersons, SearchPersonsReturnsByGroup。

名称	类型	描述
GroupId	String	包含此人员的人员库ID。 示例值：1001
PersonExDescriptions	Array of String	人员描述字段内容。 示例值：["年龄"]

PersonInfo

返回的人员信息

被如下接口引用：GetPersonList。

名称	类型	描述
PersonName	String	人员名称。 示例值：韦小宝
PersonId	String	人员Id。 示例值：1001
Gender	Integer	人员性别。 示例值：0
PersonExDescriptions	Array of String	人员描述字段内容。 示例值：["年龄"]
FaceIds	Array of String	包含的人脸照片列表。 示例值：["2877244861637985315"]
CreationTimestamp	Integer	人员的创建时间和日期 CreationTimestamp。 - CreationTimestamp 的值是自 Unix 纪元时间到Person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) 00:00:00。 - 有关更多信息，请参阅 Unix 时间。 示例值：1594642823572

Point

坐标

被如下接口引用：AnalyzeDenseLandmarks, AnalyzeFace。

名称	类型	描述
X	Integer	x坐标。 示例值：294
Y	Integer	Y坐标。 示例值：137

Result

人脸的识别结果

被如下接口引用：SearchFaces, SearchPersons。

名称	类型	描述
Candidates	Array of Candidate	识别出的最相似候选人。
FaceRect	FaceRect	检测出的人脸框位置。
RetCode	Integer	检测出的人脸图片状态返回码。 - 0 表示正常。 - -1601代表不符合图片质量控制要求，此时Candidate内容为空。 示例值：0

ResultsReturnsByGroup

识别结果。

被如下接口引用：SearchFacesReturnsByGroup, SearchPersonsReturnsByGroup。

名称	类型	描述
FaceRect	FaceRect	检测出的人脸框位置。
GroupCandidates	Array of GroupCandidate	识别结果。
RetCode	Integer	检测出的人脸图片状态返回码。 – 0 表示正常。 – –1601代表不符合图片质量控制要求，此时Candidate内容为空。

UpgradeJobInfo

人员库升级任务信息

被如下接口引用：GetUpgradeGroupFaceModelVersionJobList。

名称	类型	描述
JobId	String	人员库升级任务ID，用于查询、获取升级的进度和结果。
GroupId	String	人员库ID。
FromFaceModelVersion	String	当前算法模型版本。
ToFaceModelVersion	String	目标算法模型版本。
StartTime	Integer	升级起始时间。 StartTime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) 00:00:00。 有关更多信息，请参阅 Unix 时间。
Status	Integer	0表示升级中，1表示升级完毕，2表示回滚完毕，3表示升级失败。

错误码

最近更新时间：2024-12-27 11:13:32

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

业务错误码

在接口调用过程中，成功调用量和返回 InvalidParameterValue.NoFaceInPhoto（图片不包含人脸）错误码的失败调用量，会计入计费。

错误码	说明
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.CompareFail	比对失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。

错误码	说明
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupCannotUpgrade	指定人员库的升级操作无法执行。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupLostFaces	人员库存在人脸图片丢失。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.JobCannnotRollback	任务无法回滚。
FailedOperation.LifePhotoDetectFaces	检测到多张人脸。
FailedOperation.LifePhotoDetectFake	实人比对没通过。
FailedOperation.LifePhotoDetectNoFaces	未能检测到完整人脸。
FailedOperation.LifePhotoSizeError	传入图片过大或过小。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
FailedOperation.UpgradeJobIdNotExist	人员库升级任务不存在。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段

错误码	说明
	名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdNotExists	对应的人员库ID在库中不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.ImageEmptyError	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
InvalidParameterValue.PersonIdAlreadyExist	人员ID已经存在。人员ID不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UniquePersonControlIllegal	UniquePersonControl参数不合法。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.GetAuthInfoError	获取认证信息失败。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸识别 API 2018-03-01 更新历史

最近更新时间：2024-11-22 11:48:47

第 21 次发布

发布时间：2024-11-22 11:48:39

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- FaceRect
  - 修改成员：X, Y, Width, Height

第 20 次发布

发布时间：2022-04-22 18:43:48

本次发布包含了以下内容：

改善已有的文档。

删除接口：

- CheckSimilarPerson
- EstimateCheckSimilarPersonCostTime
- GetCheckSimilarPersonJobIdList
- GetSimilarPersonResult

删除数据结构：

- JobIdInfo

第 19 次发布

发布时间：2020-11-05 08:04:14

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- VerifyPerson
  - 新增出参：Score, IsMatch, FaceModelVersion

第 18 次发布

发布时间：2020-11-02 08:04:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- VerifyPerson
  - **修改入参：** PersonId
  - **删除出参：** Score, IsMatch, FaceModelVersion

第 17 次发布

发布时间：2020-08-13 08:12:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- AnalyzeDenseLandmarks
- DetectFaceAttributes

新增数据结构：

- AttributeItem
- DenseFaceShape
- Eye
- Eyebrow
- FaceDetailAttributesInfo
- FaceDetailInfo
- Hair
- Hat
- HeadPose
- Mouth

第 16 次发布

发布时间：2020-08-06 08:11:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- GetUpgradeGroupFaceModelVersionJobList
- GetUpgradeGroupFaceModelVersionResult
- RevertGroupFaceModelVersion
- UpgradeGroupFaceModelVersion

新增数据结构：

- UpgradeJobInfo

第 15 次发布

发布时间：2020-04-21 08:10:37

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- AnalyzeFace
  - 新增入参：NeedRotateDetection
- CompareFace
  - 新增入参：NeedRotateDetection
- CreateFace
  - 新增入参：NeedRotateDetection
- CreatePerson
  - 新增入参：NeedRotateDetection
- DetectFace
  - 新增入参：NeedRotateDetection
- SearchFaces
  - 新增入参：NeedRotateDetection
- SearchFacesReturnsByGroup
  - 新增入参：NeedRotateDetection
- SearchPersons
  - 新增入参：NeedRotateDetection
- SearchPersonsReturnsByGroup
  - 新增入参：NeedRotateDetection
- VerifyFace
  - 新增入参：NeedRotateDetection
- VerifyPerson
  - 新增入参：NeedRotateDetection

第 14 次发布

发布时间：2020-01-09 17:29:23

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- GetCheckSimilarPersonJobIdList
- GetGroupInfo

新增数据结构：

- JobIdInfo

第 13 次发布

发布时间：2019-12-05 17:47:15

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- CheckSimilarPerson
- EstimateCheckSimilarPersonCostTime
- GetSimilarPersonResult

修改接口：

- SearchPersons
  - 新增入参：NeedPersonInfo
- SearchPersonsReturnsByGroup
  - 新增入参：NeedPersonInfo

第 12 次发布

发布时间：2019-11-14 18:07:33

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- DetectLiveFace
  - 新增入参：FaceModelVersion
  - 新增出参：FaceModelVersion, IsLiveness

第 11 次发布

发布时间：2019-11-01 10:51:30

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- SearchFacesReturnsByGroup
- SearchPersons
- SearchPersonsReturnsByGroup
- VerifyPerson

修改接口：

- CompareFace
  - 新增入参：QualityControl
- CreateFace
  - 新增入参：FaceMatchThreshold, QualityControl
  - 新增出参：SucIndexes, SucFaceRects, FaceModelVersion
- CreateGroup
  - 新增入参：FaceModelVersion
  - 新增出参：FaceModelVersion
- CreatePerson
  - 新增入参：UniquePersonControl, QualityControl
  - 新增出参：FaceRect, SimilarPersonId, FaceModelVersion
- GetPersonGroupInfo
  - 新增出参：FaceModelVersion
- GetPersonList
  - 新增出参：FaceModelVersion
- SearchFaces
  - 新增入参：QualityControl, FaceMatchThreshold
  - 新增出参：FaceModelVersion
- VerifyFace
  - 新增入参：QualityControl

版权所有：腾讯云计算（北京）有限责任公司

- 新增出参：FaceModelVersion

新增数据结构：

- GroupCandidate
- ResultsReturnsByGroup

修改数据结构：

- PersonInfo
  - 新增成员：CreationTimestamp
- Result
  - 新增成员：RetCode

第 10 次发布

发布时间：2019-10-11 14:13:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- GroupInfo
  - **修改成员：** CreationTimestamp

第 9 次发布

发布时间：2019-09-19 19:46:00

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- CompareFace
  - 新增入参：FaceModelVersion
  - 新增出参：FaceModelVersion

第 8 次发布

发布时间：2019-08-22 19:37:04

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- AnalyzeFace
  - 新增出参：FaceModelVersion
- DetectFace
  - 新增入参：FaceModelVersion
  - 新增出参：FaceModelVersion

第 7 次发布

发布时间：2019-08-01 17:27:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- AnalyzeFace
  - 新增入参：FaceModelVersion
- SearchFaces
  - 新增入参：NeedPersonInfo

修改数据结构：

- Candidate
  - 新增成员：PersonName, Gender, PersonGroupInfos

第 6 次发布

发布时间：2019-07-04 16:08:02

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- GroupInfo
  - 新增成员：FaceModelVersion, CreationTimestamp

第 5 次发布

发布时间：2019-06-27 17:24:44

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- GroupInfo
  - **删除成员：** FaceModelVersion, CreationTimestamp

第 4 次发布

发布时间：2019-06-06 20:17:29

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- GroupInfo
  - 新增成员：FaceModelVersion, CreationTimestamp

第 3 次发布

发布时间：2019-04-12 10:44:26

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- FaceAttributesInfo
  - 新增成员：EyeOpen

第 2 次发布

发布时间：2019-02-22 11:28:36

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- FaceRect
  - **修改成员：** X, Y

第 1 次发布

发布时间：2019-01-23 16:53:11

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- AnalyzeFace
- CompareFace
- CopyPerson
- CreateFace
- CreateGroup
- CreatePerson
- DeleteFace
- DeleteGroup
- DeletePerson
- DeletePersonFromGroup
- DetectFace
- DetectLiveFace
- GetGroupList
- GetPersonBaseInfo
- GetPersonGroupInfo
- GetPersonList
- GetPersonListNum
- ModifyGroup
- ModifyPersonBaseInfo
- ModifyPersonGroupInfo
- SearchFaces
- VerifyFace

新增数据结构：

- Candidate
- FaceAttributesInfo

- FaceHairAttributesInfo
- FaceInfo
- FaceQualityCompleteness
- FaceQualityInfo
- FaceRect
- FaceShape
- GroupExDescriptionInfo
- GroupInfo
- PersonExDescriptionInfo
- PersonGroupInfo
- PersonInfo
- Point
- Result

版权所有：腾讯云计算（北京）有限责任公司

简介

最近更新时间：2025-04-25 01:35:25

精准识别

腾讯云人脸识别服务在多个国际公开竞赛中刷新纪录，人脸比对在 2017 年的LFW 测评准确度高达 99.80%，人脸检索在 MegaFace 竞赛百万规模中首选识别率 83.29% 排名第一，识别准确率业界领先。

稳定性强

腾讯云人脸识别服务已经通过腾讯内部产品海量用户和复杂场景验证，运行稳定、鲁棒性强，服务可用性可达99.9%。

算法领先

基于第三代腾讯优图祖母模型，融合度量学习、迁移学习、多任务学习等多种训练手段来优化模型；针对不同场景业务的特点，定制化finetuning或distilling模型，满足业务性能与时延的双重需求。

简单易用

提供了丰富多样的API接口和离线SDK，快速植入应用和硬件设备，全面满足云和端的需求。

用途广泛

广泛应用于刷脸门禁考勤、人脸签到、人脸支付和人脸登录等众多场景。

实时响应

人脸识别具备高并发、高吞吐、低时延等特点，即使是百万规模人脸搜索，仍只需数百毫秒即可处理完毕，满足您的实时使用需求。

版权所有：腾讯云计算（北京）有限责任公司

API 概览

最近更新时间：2024-08-02 01:40:25

人脸检测与分析相关接口

接口名称	接口功能
DetectFace	人脸检测与分析
DetectFaceAttributes	人脸检测与属性分析

五官定位相关接口

接口名称	接口功能
AnalyzeDenseLandmarks	稠密关键点
AnalyzeFace	五官定位

人脸比对相关接口

接口名称	接口功能
CompareFace	人脸比对

人员库管理相关接口

接口名称	接口功能
CreateGroup	创建人员库
DeleteGroup	删除人员库
GetGroupInfo	获取人员库信息
GetGroupList	获取人员库列表
ModifyGroup	修改人员库
CreatePerson	创建人员
DeletePerson	删除人员
DeletePersonFromGroup	人员库删除人员
GetPersonGroupInfo	获取人员归属信息
GetPersonBaseInfo	获取人员基础信息
GetPersonList	获取人员列表
GetPersonListNum	获取人员列表长度
ModifyPersonGroupInfo	修改人员描述信息
ModifyPersonBaseInfo	修改人员基础信息

版权所有：腾讯云计算（北京）有限责任公司

接口名称	接口功能
CopyPerson	复制人员
CreateFace	增加人脸
DeleteFace	删除人脸
GetUpgradeGroupFaceModelVersionJobList	获取人员库升级任务列表
GetUpgradeGroupFaceModelVersionResult	人员库升级结果查询
RevertGroupFaceModelVersion	人员库版本回滚
UpgradeGroupFaceModelVersion	人员库升级

人脸搜索相关接口

接口名称	接口功能
SearchFaces	人脸搜索
SearchFacesReturnsByGroup	人脸搜索分库返回
SearchPersons	人员搜索
SearchPersonsReturnsByGroup	人员搜索按库返回

人脸验证相关接口

接口名称	接口功能
VerifyFace	人脸验证
VerifyPerson	人员验证

人脸静态活体检测相关接口

接口名称	接口功能
DetectLiveFace	人脸静态活体检测

> ⚠ 注意：
> 以上给出的接口频率限制维度为 `API` + 接入地域 + 子账号，有关限频更多说明参考：API 频率限制说明

调用方式
请求结构

最近更新时间：2025-05-21 01:20:42

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

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

版权所有：腾讯云计算（北京）有限责任公司

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

2. 通信协议

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

3. 请求方法

支持的 HTTP 请求方法：

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

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

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

4. 字符编码

均使用 `UTF-8` 编码。

公共参数

最近更新时间：2024-11-15 01:39:19

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

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

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

HTTP GET 请求结构示例：

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

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

HTTP POST（application/json）请求结构示例：

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

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

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

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

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

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

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

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

10
--58731222010402--
```

使用签名方法 v1 的公共参数

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

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。**注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。**
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

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

HTTP GET 请求结构示例:

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

Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例:

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

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

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

地域列表

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

地域	取值
华北地区（北京）	ap-beijing
西南地区（成都）	ap-chengdu
西南地区（重庆）	ap-chongqing
华南地区（广州）	ap-guangzhou
华东地区（南京）	ap-nanjing
华东地区（上海）	ap-shanghai
亚太东南（新加坡）	ap-singapore

版权所有：腾讯云计算（北京）有限责任公司

签名方法 v3

最近更新时间：2024-12-25 01:38:39

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

推荐使用 API Explorer　　点击调试

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

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

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份
   签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 云 API 密钥 页面获取密钥相关信息。

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

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

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

申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

签名版本 v3 签名过程

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

下面以云服务器查询广州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。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC- 前缀）。

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

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

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST ）。此示例取值为 `POST`。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠（/）。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）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 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；

字段名称	解释
	2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 `content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n`。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号（;）分隔。 此示例为 content-type;host;x-tc-action
HashedRequestPayload	请求正文（payload，即 body，此示例为 `{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}`）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 `35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064`。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

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

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

字段名称	解释
Algorithm	签名算法，目前固定为 `TC3-HMAC-SHA256`。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 `1551113065`。
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的

字段名称	解释
	UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

注意：

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

根据以上规则，示例中得到的待签名字符串如下：

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

3. 计算签名

1）计算派生签名密钥，伪代码如下：

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

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

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

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

2）计算签名，伪代码如下：

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

此示例计算结果是 `10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f`。

4. 拼接 Authorization

按如下格式拼接 Authorization：

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

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

根据以上规则，示例中得到的值为：

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

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID********************************/2019-02-25/cvm/tc3_request,
SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b4
6093f6ab6c4f
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

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

> ⚠ **注意：**
>
> **请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过**

> 打印实际请求内容，网络抓包等方式对比排查。

签名演示

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

- Python
- Java
- PHP
- Go
- NodeJS
- .NET
- C++
- Ruby

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

- Signature Demo

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

Java

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

public class TencentCloudAPITC3Demo {
private final static Charset UTF8 = StandardCharsets.UTF_8;
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID********************************
private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
private final static String CT_JSON = "application/json; charset=utf-8";

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

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

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

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

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instan
ce-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" + hashedCanonicalReques
t;
System.out.println(stringToSign);

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

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

```
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

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

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

Python

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

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

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
```

```
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

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


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

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

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + host + '"'
```

```
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'")
```

Golang

```
package main

import (
"crypto/hmac"
"crypto/sha256"
"encoding/hex"
"fmt"
"os"
"strings"
"time"
)

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

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

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

// step 1: build canonical request string
httpRequestMethod := "POST"
canonicalURI := "/"
canonicalQueryString := ""
```

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

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

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

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

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

```
fmt.Println(curl)
}
```

PHP

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

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = implode("\n", [
"content-type:application/json; charset=utf-8",
"host:".$host,
"x-tc-action:".strtolower($action),
""
]);
$signedHeaders = implode(";", [
"content-type",
"host",
"x-tc-action",
]);
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

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

```
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

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

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

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

Ruby

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

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

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
```

```
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

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

puts canonical_request

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

# ************* 步骤 3：计算签名 *************
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

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

```
Signature=#{signature}"
puts authorization

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

DotNet

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

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

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

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

```
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZer
o) / 1000;
// ************* 步骤 1：拼接规范请求串 *************
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

// ************* 步骤 2：拼接待签名字符串 *************
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n"
+ requestTimestamp.ToString() + "\n"
+ credentialScope + "\n"
+ hashedCanonicalRequest;
Console.WriteLine(stringToSign);

// ************* 步骤 3：计算签名 *************
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);

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

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
```

```
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}
public static void Main(string[] args)
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID*******************************
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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

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

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

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

NodeJS

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

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

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

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

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

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

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

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

const canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload
```

```
console.log(canonicalRequest)

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

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

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

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

C++

```
#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
```

```
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

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

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

string sha256Hex(const string &str)
{
char buf[3];
unsigned char hash[SHA256_DIGEST_LENGTH];
SHA256_CTX sha256;
SHA256_Init(&sha256);
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
snprintf(buf, sizeof(buf), "%02x", hash[i]);
NewString = NewString + buf;
}
return NewString;
}

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

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

```
#else
h = HMAC_CTX_new();
#endif

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

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

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

return (ss.str());
}

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

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

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

string service = "cvm";
```

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

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

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

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

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

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

C

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

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

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

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
```

```
unsigned char* output, unsigned int* output_len)
{
HMAC_CTX *h;
#if OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

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


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

}

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

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

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

int main()
```

```
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID********************************
const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
const char* service = "cvm";
const char* host = "cvm.tencentcloudapi.com";
const char* region = "ap-guangzhou";
const char* action = "DescribeInstances";
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

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

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


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

```
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

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

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

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

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

其他语言

- Lua: GitHub
- Swift: GitHub
- Dart: GitHub
- Shell(Bash): GitHub

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

签名方法

最近更新时间：2024-12-25 01:38:39

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

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

推荐使用 API Explorer　点击调试

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

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

1. 申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

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

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

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKID********************************
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886

参数名称	中文	参数值
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

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

2.1. 对参数排序

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

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

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

2.2. 拼接请求字符串

此步骤生成请求字符串。
将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 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&Secre
tId=AKID******************************&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

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

4. 请求字符串: 即上一步生成的请求字符串。

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

示例的拼接结果为：

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

2.4. 生成签名串

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

具体代码如下，以 PHP 语言为例：

```
$secretKey = '********************************';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID********************************&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ= ，最终得到的签名串请求参数（Signature）为：7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 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

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

5. 签名演示

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

- Python
- Java
- PHP
- Go
- NodeJS
- .NET
- C++
- Ruby

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

- Signature Demo

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID********************************&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12` 。

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

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

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

Java

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

public class TencentCloudAPIDemo {
```

```
private final static String CHARSET = "UTF-8";

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

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

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

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID********************************
params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
params.put("Action", "DescribeInstances"); // 公共参数
params.put("Version", "2017-03-12"); // 公共参数
params.put("Region", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA
1")); // 公共参数
System.out.println(getUrl(params));
}
}
```

Python

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

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

import requests

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

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

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

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

Golang

```
package main

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

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

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

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

for i := range keys {
k := keys[i]
buf.WriteString(k)
buf.WriteString("=")
buf.WriteString(params[k])
buf.WriteString("&")
```

```
}
buf.Truncate(buf.Len() - 1)

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

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

PHP

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

ksort($param);

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

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

Ruby

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

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

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

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

DotNet

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

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

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

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

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

NodeJS

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

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

function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}
function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

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

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

```
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 ********************************
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

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

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

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

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

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

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

返回结果

最近更新时间：2024-03-12 19:51:44

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

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

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

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

公共错误码

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

参数类型

最近更新时间：2022-08-10 06:12:18

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

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

人脸检测与分析相关接口
人脸检测与分析

最近更新时间：2025-04-25 01:35:20

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

检测给定图片中的人脸（Face）的位置、相应的面部属性和人脸质量信息，位置包括(x，y，w，h)，面部属性包括性别（gender）、年龄（age）、表情（expression）、魅力（beauty）、眼镜（glass）、发型（hair）、口罩（mask）和姿态(pitch，roll，yaw)，人脸质量信息包括整体质量分（score）、模糊分（sharpness）、光照分（brightness）和五官遮挡分（completeness）。

其中，人脸质量信息主要用于评价输入的人脸图片的质量。在使用人脸识别服务时，建议您对输入的人脸图片进行质量检测，提升后续业务处理的效果。该功能的应用场景包括：

1）人员库创建人员/增加人脸：保证人员人脸信息的质量，便于后续的业务处理。

2）人脸搜索：保证输入的图片质量，快速准确匹配到对应的人员。

3）人脸验证：保证人脸信息的质量，避免明明是本人却认证不通过的情况。

4）人脸融合：保证上传的人脸质量，人脸融合的效果更好。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：DetectFace。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
MaxFaceNum	否	Integer	最多处理的人脸数目。默认值为1（仅检测图片中面积最大的那张人脸），最大值为120。 此参数用于控制处理待检测图片中的人脸个数，值越小，处理速度越快。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。 默认为34。建议不低于34。 低于MinFaceSize值的人脸不会被检测。 示例值：40
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。

参数名称	必选	类型	描述
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。 示例值：http://test.image.myqcloud.com/testB.jpg
NeedFaceAttributes	否	Integer	是否需要返回人脸属性信息（FaceAttributesInfo）。0 为不需要返回，1 为需要返回。默认为 0。 非 1 值均视为不需要返回，此时 FaceAttributesInfo 不具备参考意义。 最多返回面积最大的 5 张人脸属性信息，超过 5 张人脸（第 6 张及以后的人脸）的 FaceAttributesInfo 不具备参考意义。 提取人脸属性信息较为耗时，如不需要人脸属性信息，建议关闭此项功能，加快人脸检测速度。 示例值：0
NeedQualityDetection	否	Integer	是否开启质量检测。0 为关闭，1 为开启。默认为 0。 非 1 值均视为不进行质量检测。 最多返回面积最大的 30 张人脸质量分信息，超过 30 张人脸（第 31 张及以后的人脸）的 FaceQualityInfo不具备参考意义。 建议：人脸入库操作建议开启此功能。 示例值：0
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。目前入参支持 “2.0”和“3.0“ 两个输入。 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：640
ImageHeight	Integer	请求的图片高度。 示例值：440
FaceInfos	Array of FaceInfo	人脸信息列表。包含人脸坐标信息、属性信息（若需要）、质量分信息（若需要）。
FaceModelVersion	String	人脸识别服务所用的算法模型版本。 目前入参支持 “2.0”和“3.0“ 两个输入。 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 2020年11月26日后开通服务的账号仅支持输入“3.0”。 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0

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

4. 示例

示例1 人脸检测与分析接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=DetectFace
&MaxFaceNum=1
&MinFaceSize=40
&Url=http://test.image.myqcloud.com/testB.jpg
&NeedFaceAttributes=0
&NeedQualityDetection=0
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ImageWidth": 640,
"ImageHeight": 440,
"FaceInfos": [
{
"X": 156,
"Y": 129,
"Width": 196,
"Height": 196,
"FaceAttributesInfo": {
"Gender": 0,
"Age": 0,
"Expression": 0,
"Hat": false,
"Glass": false,
"EyeOpen": true,
"Mask": false,
"Hair": {
"Length": 0,
"Bang": 0,
"Color": 0
},
"Pitch": 0,
"Yaw": 0,
"Roll": 0,
"Beauty": 0
},
"FaceQualityInfo": {
```

```
"Score": 0,
"Sharpness": 0,
"Brightness": 0,
"Completeness": {
"Eyebrow": 0,
"Eye": 0,
"Nose": 0,
"Cheek": 0,
"Mouth": 0,
"Chin": 0
}
}
}
],
"FaceModelVersion": "3.0",
"RequestId": "a574102d-1b86-48a7-a08b-6a741a8fedb6"
}
}
```


示例2 人脸检测与分析接口-2

输入示例

```
https://iai.tencentcloudapi.com/?Action=DetectFace
&MaxFaceNum=1
&MinFaceSize=40
&Url=http://test.image.myqcloud.com/testA.jpg
&NeedFaceAttributes=1
&NeedQualityDetection=1
&<公共请求参数>
```

输出示例


```
{
"Response": {
"ImageWidth": 640,
"ImageHeight": 440,
"FaceInfos": [
{
"X": 156,
"Y": 129,
"Width": 196,
"Height": 196,
"FaceAttributesInfo": {
"Gender": 99,
"Age": 51,
"Expression": 99,
"Hat": false,
"Glass": false,
```

```
"EyeOpen": true,
"Mask": true,
"Hair": {
"Length": 1,
"Bang": 1,
"Color": 0
},
"Pitch": 17,
"Yaw": 5,
"Roll": -2,
"Beauty": 71
},
"FaceQualityInfo": {
"Score": 63,
"Sharpness": 73,
"Brightness": 47,
"Completeness": {
"Eyebrow": 99,
"Eye": 99,
"Nose": 99,
"Cheek": 52,
"Mouth": 99,
"Chin": 47
}
}
}
],
"FaceModelVersion": "3.0",
"RequestId": "bcde47b5-e6d8-446e-a538-bcecffbe306a"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee

- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation	操作失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。

错误码	描述
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonIdTooLong	人员ID超出长度限制。
InvalidParameterValue.PersonNameIllegal	人员名称包含非法字符。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded	超过配额限制。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸检测与属性分析

最近更新时间：2025-04-25 01:35:20

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

检测给定图片中的人脸（Face）的位置、相应的面部属性和人脸质量信息，位置包括(x，y，w，h)，面部属性包括性别（gender）、年龄（age）、表情（expression）、魅力（beauty）、眼镜（glass）、发型（hair）、口罩（mask）和姿态(pitch，roll，yaw)，人脸质量信息包括整体质量分（score）、模糊分（sharpness）、光照分（brightness）和五官遮挡分（completeness）。

其中，人脸质量信息主要用于评价输入的人脸图片的质量。在使用人脸识别服务时，建议您对输入的人脸图片进行质量检测，提升后续业务处理的效果。该功能的应用场景包括：

1）人员库创建人员/增加人脸：保证人员人脸信息的质量，便于后续的业务处理。

2）人脸搜索：保证输入的图片质量，快速准确匹配到对应的人员。

3）人脸验证：保证人脸信息的质量，避免明明是本人却认证不通过的情况。

4）人脸融合：保证上传的人脸质量，人脸融合的效果更好。

- 本接口是人脸检测与分析的升级，具体在于：

1.本接口可以指定需要计算返回的人脸属性，避免无效计算，降低耗时；

2.本接口支持更多属性细项数，也会持续增加更多功能。

请您使用本接口完成相应的人脸检测与属性分析需求。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：DetectFaceAttributes。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	是	String	公共参数，详见产品支持的 地域列表。
MaxFaceNum	否	Integer	最多处理的人脸数目。 默认值为1（仅检测图片中面积最大的那张人脸），最大值为120。 此参数用于控制处理待检测图片中的人脸个数，值越小，处理速度越快。 示例值：1
Image	否	String	图片 base64 数据，base64 编码后大小不可超过5M。 jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。

参数名称	必选	类型	描述
			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。 jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 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。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceAttributesType	否	String	是否返回年龄、性别、情绪等属性。 合法值为（大小写不敏感）：None、Age、Beauty、Emotion、Eye、Eyebrow、Gender、Hair、Hat、Headpose、Mask、Mouth、Moustache、Nose、Shape、Skin、Smile。 None为不需要返回。默认为 None。即FaceAttributesType属性为空时，各属性返回值为0。 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 张人脸属性信息，超过 5 张人脸（第 6 张及以后的人脸）的 AttributesInfo 不具备参考意义。 示例值：eye
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。本接口仅支持“3.0”输入 示例值：3.0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：550
ImageHeight	Integer	请求的图片高度。 示例值：366
FaceDetailInfos	Array of FaceDetailInfo	人脸信息列表。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸检测与属性分析

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"FaceAttributesType": "eye"
}
```

输出示例

```
{
"Response": {
"ImageWidth": 550,
"ImageHeight": 366,
"FaceModelVersion": "3.0",
"FaceDetailInfos": [
{
"FaceRect": {
"X": 375,
"Y": 37,
"Width": 63,
"Height": 82
},
"FaceDetailAttributesInfo": {
"Age": 0,
"Beauty": 0,
"Emotion": {
"Type": 0,
"Probability": 0
},
"Eye": {
"Glass": {
"Type": 0,
"Probability": 0.99936753511429
},
"EyeOpen": {
"Type": 0,
"Probability": 0.99949336051941
},
"EyelidType": {
"Type": 1,
"Probability": 0.75467920303345
},
"EyeSize": {
"Type": 2,
"Probability": 0.59152442216873
```

```
}
},
"Eyebrow": {
"EyebrowDensity": {
"Type": 0,
"Probability": 0
},
"EyebrowCurve": {
"Type": 0,
"Probability": 0
},
"EyebrowLength": {
"Type": 0,
"Probability": 0
}
},
"Gender": {
"Type": 0,
"Probability": 0
},
"Hair": {
"Length": {
"Type": 0,
"Probability": 0
},
"Bang": {
"Type": 0,
"Probability": 0
},
"Color": {
"Type": 0,
"Probability": 0
}
},
"Hat": {
"Style": {
"Type": 0,
"Probability": 0
},
"Color": {
"Type": 0,
"Probability": 0
}
},
"HeadPose": {
"Pitch": 0,
"Yaw": 0,
"Roll": 0
},
"Mask": {
```


版权所有：腾讯云计算（北京）有限责任公司

```
"Type": 0,
"Probability": 0
},
"Mouth": {
"MouthOpen": {
"Type": 0,
"Probability": 0
}
},
"Moustache": {
"Type": 0,
"Probability": 0
},
"Nose": {
"Type": 2,
"Probability": 0.75233882665634
},
"Shape": {
"Type": 0,
"Probability": 0
},
"Skin": {
"Type": 0,
"Probability": 0
},
"Smile": 0
}
}
],
"RequestId": "b2c154b9-4620-4d37-8fd1-f6af3748f998"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee

- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

五官定位相关接口
五官定位

最近更新时间：2025-05-13 01:37:08

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对请求图片进行五官定位（也称人脸关键点定位），计算构成人脸轮廓的 90 个点，包括眉毛（左右各 8 点）、眼睛（左右各 8 点）、鼻子（13 点）、嘴巴（22 点）、脸型轮廓（21 点）、眼珠[或瞳孔]（2点）。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：AnalyzeFace。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
Mode	否	Integer	检测模式。 - 取值范围：0 为检测所有出现的人脸， 1 为检测面积最大的人脸。 - 默认为 0。 - 最多返回 10 张人脸的五官定位（人脸关键点）具体信息。 示例值：0
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。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
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。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 - 目前入参支持 “2.0”和“3.0“ 两个输入。 - 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。

参数名称	必选	类型	描述
			- 2020年11月26日后开通服务的账号仅支持输入“3.0”。 - 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：550
ImageHeight	Integer	请求的图片高度。 示例值：366
FaceShapeSet	Array of FaceShape	五官定位（人脸关键点）具体信息。
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 五官定位接口成功示例

对请求图片进行五官定位。

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"Mode": "0"
}
```

输出示例

```
{
"Response": {
"ImageWidth": 550,
"ImageHeight": 366,
"FaceShapeSet": [
{
"FaceProfile": [
{
"X": 63,
"Y": 335
},
{
"X": 63,
"Y": 374
},
{
"X": 66,
"Y": 412
},
{
"X": 74,
"Y": 450
},
{
"X": 85,
"Y": 487
},
{
"X": 100,
"Y": 522
},
{
"X": 121,
"Y": 554
},
{
"X": 147,
"Y": 582
},
{
"X": 176,
"Y": 608
},
{
"X": 208,
"Y": 627
},
{
"X": 245,
```

```
"Y": 634
},
{
"X": 282,
"Y": 627
},
{
"X": 315,
"Y": 607
},
{
"X": 344,
"Y": 582
},
{
"X": 370,
"Y": 554
},
{
"X": 391,
"Y": 522
},
{
"X": 405,
"Y": 487
},
{
"X": 416,
"Y": 449
},
{
"X": 423,
"Y": 411
},
{
"X": 427,
"Y": 372
},
{
"X": 426,
"Y": 334
}
],
"LeftEye": [
{
"X": 114,
"Y": 333
},
{
"X": 128,
```

```
"Y": 345
},
{
"X": 146,
"Y": 349
},
{
"X": 165,
"Y": 347
},
{
"X": 183,
"Y": 341
},
{
"X": 169,
"Y": 325
},
{
"X": 150,
"Y": 318
},
{
"X": 130,
"Y": 321
}
],
"RightEye": [
{
"X": 378,
"Y": 331
},
{
"X": 364,
"Y": 343
},
{
"X": 345,
"Y": 348
},
{
"X": 327,
"Y": 346
},
{
"X": 309,
"Y": 340
},
{
"X": 322,
```

```
"Y": 323
},
{
"X": 341,
"Y": 316
},
{
"X": 361,
"Y": 319
}
],
"LeftEyeBrow": [
{
"X": 79,
"Y": 280
},
{
"X": 108,
"Y": 275
},
{
"X": 138,
"Y": 274
},
{
"X": 168,
"Y": 277
},
{
"X": 198,
"Y": 279
},
{
"X": 173,
"Y": 256
},
{
"X": 139,
"Y": 251
},
{
"X": 105,
"Y": 256
}
],
"RightEyeBrow": [
{
"X": 410,
"Y": 277
},
```

```
{
"X": 380,
"Y": 273
},
{
"X": 350,
"Y": 272
},
{
"X": 320,
"Y": 275
},
{
"X": 290,
"Y": 277
},
{
"X": 315,
"Y": 255
},
{
"X": 349,
"Y": 249
},
{
"X": 384,
"Y": 254
}
],
"Mouth": [
{
"X": 173,
"Y": 522
},
{
"X": 193,
"Y": 541
},
{
"X": 217,
"Y": 554
},
{
"X": 244,
"Y": 558
},
{
"X": 272,
"Y": 554
},
```

```
{
"X": 297,
"Y": 541
},
{
"X": 317,
"Y": 522
},
{
"X": 291,
"Y": 517
},
{
"X": 264,
"Y": 512
},
{
"X": 245,
"Y": 517
},
{
"X": 225,
"Y": 512
},
{
"X": 199,
"Y": 517
},
{
"X": 196,
"Y": 528
},
{
"X": 220,
"Y": 532
},
{
"X": 244,
"Y": 535
},
{
"X": 269,
"Y": 532
},
{
"X": 293,
"Y": 528
},
{
"X": 293,
```

```
"Y": 525
},
{
"X": 269,
"Y": 527
},
{
"X": 245,
"Y": 530
},
{
"X": 221,
"Y": 528
},
{
"X": 197,
"Y": 525
}
],
"Nose": [
{
"X": 244,
"Y": 428
},
{
"X": 245,
"Y": 341
},
{
"X": 231,
"Y": 367
},
{
"X": 217,
"Y": 392
},
{
"X": 203,
"Y": 418
},
{
"X": 187,
"Y": 448
},
{
"X": 217,
"Y": 464
},
{
"X": 245,
```

```
"Y": 468
},
{
"X": 272,
"Y": 464
},
{
"X": 302,
"Y": 448
},
{
"X": 287,
"Y": 417
},
{
"X": 273,
"Y": 392
},
{
"X": 259,
"Y": 366
}
],
"LeftPupil": [
{
"X": 188,
"Y": 303
}
],
"RightPupil": [
{
"X": 201,
"Y": 304
}
]
}
],
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```


示例2 五官定位接口异常示例

对请求图片进行五官定位，图片中无人脸。

输入示例

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

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

{
"Url": "http://test.image.myqcloud.com/testB.jpg",
"Mode": "0"
}
```

输出示例

```
{
"Response": {
"RequestId": "c964ad35-5476-486e-a03f-311faa6652e6",
"Error": {
"Code": "InvalidParameterValue.NoFaceInPhoto",
"Message": "图片中没有人脸。"
}
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

稠密关键点

最近更新时间：2025-04-25 01:35:25

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对请求图片进行五官定位（也称人脸关键点定位），获得人脸的精准信息，返回多达888点关键信息，对五官和脸部轮廓进行精确定位。

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：AnalyzeDenseLandmarks。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
Mode	否	Integer	检测模式。 - 取值范围：0 为检测所有出现的人脸，1 为检测面积最大的人脸。 - 默认为 0。 - 最多返回 5 张人脸的五官定位（人脸关键点）具体信息。 示例值：1
Image	否	String	图片 base64 数据。 - base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
Url	否	String	图片的 Url 。 - 对应图片 base64 编码后大小不可超过5M。 - jpg格式长边像素不可超过4000，其他格式图片长边像素不可超2000。 - 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。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。本接口仅支持 “3.0 “ 输入。 示例值：3.0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。 - 取值范围：0为不开启，1为开启。 - 默认为0。 - 本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。 - 若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参

参数名称	必选	类型	描述
			数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
ImageWidth	Integer	请求的图片宽度。 示例值：98
ImageHeight	Integer	请求的图片高度。 示例值：189
DenseFaceShapeSet	Array of DenseFaceShape	稠密人脸关键点具体信息。
FaceModelVersion	String	人脸识别服务所用的算法模型版本。本接口仅支持 “3.0 “ 输入。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId ）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 稠密关键点接口

对请求图片进行五官定位，获得人脸的精准信息。

输入示例

```
https://iai.tencentcloudapi.com/?Action=AnalyzeDenseLandmarks
&Mode=0
&Url=http://test.image.myqcloud.com/testA.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ImageWidth": 940,
"ImageHeight": 930,
"DenseFaceShapeSet": [
{
"X": 450,
"Y": 230,
"Width": 340,
"Height": 540,
"LeftEye": [
{
"X": 20,
"Y": 40
```

```
}
],
"RightEye": [
{
"X": 80,
"Y": 30
}
],
"LeftEyeBrow": [
{
"X": 94,
"Y": 90
}
]
}
],
"FaceModelVersion": "3.0",
"RequestId": "1219c74e7-698e-4ac2-8bc4-4c996cbdda8b"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸比对相关接口
人脸比对

最近更新时间：2025-05-13 01:37:07

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

对两张图片中的人脸进行相似度比对，返回人脸相似度分数。

若您需要判断 “此人是否是某人”，即验证某张图片中的人是否是已知身份的某人，如常见的人脸登录场景，建议使用人脸验证或人员验证接口。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：CompareFace。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
ImageA	否	String	A 图片 base64 数据，base64 编码后大小不可超过5M。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
ImageB	否	String	B 图片 base64 数据，base64 编码后大小不可超过5M。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：/9j/4AAQSkZJRg.....s97n//2Q==
UrlA	否	String	A 图片的 Url ，对应图片 base64 编码后大小不可超过5M。 A 图片的 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。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
UrlB	否	String	B 图片的 Url ，对应图片 base64 编码后大小不可超过5M。 B 图片的 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。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。

版权所有：腾讯云计算（北京）有限责任公司

参数名称	必选	类型	描述
			支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testB.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 目前入参支持 “2.0”和“3.0“ 两个输入。 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 2020年11月26日后开通服务的账号仅支持输入“3.0”。 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1: 较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多，在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	两张图片中人脸的相似度分数。 不同算法版本返回的相似度分数不同。 若需要验证两张图片中人脸是否为同一人，3.0版本误识率千分之一对应分数为40分，误识率万分之一对应分数为50分，误识率十万分之一对应分数为60分。 一般超过50分则可认定为同一人。 2.0版本误识率千分之一对应分数为70分，误识率万分之一对应分数为80分，误识率十万分之一对应分数为90分。 一般超过80分则可认定为同一人。 若需要验证两张图片中的人脸是否为同一人，建议使用人脸验证接口。 示例值：0.999
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸比对接口

输入示例

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

{
"UrlA": "http://test.image.myqcloud.com/testA.jpg",
"UrlB": "http://test.image.myqcloud.com/testB.jpg"
}
```

输出示例

```
{
"Response": {
"Score": 0.999,
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c",
"FaceModelVersion": "3.0"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员库管理相关接口
创建人员库

最近更新时间：2025-04-25 01:35:24

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于创建一个空的人员库，如果人员库已存在返回错误。
可根据需要创建自定义描述字段，用于辅助描述该人员库下的人员信息。

1个APPID下最多创建10万个人员库（Group）、最多包含5000万张人脸（Face）。

不同算法模型版本（FaceModelVersion）的人员库（Group）最多可包含人脸（Face）数不同。算法模型版本为2.0的人员库最多包含100万张人脸，算法模型版本为3.0的人员库最多可包含300万张人脸。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupName	是	String	人员库名称，[1,60]个字符，可修改，不可重复。 示例值：腾讯深圳员工库
GroupId	是	String	人员库 ID，不可修改，不可重复。支持英文、数字、-%@#&_，长度限制64B。 示例值：TencentShenZhenEmployee
GroupExDescriptions.N	否	Array of String	人员库自定义描述字段，用于描述人员库中人员属性，该人员库下所有人员将拥有此描述字段。 最多可以创建5个。 每个自定义描述字段支持[1,30]个字符。 在同一人员库中自定义描述字段不可重复。 例： 设置某人员库“自定义描述字段”为["学号","工号","手机号"]， 则该人员库下所有人员将拥有名为“学号”、“工号”、“手机号”的描述字段， 可在对应人员描述字段中填写内容，登记该人员的学号、工号、手机号等信息。 示例值：["\u4e8b\u4e1a\u7fa4","\u90e8\u95e8\u540d","\u7ec4\u540d"]
Tag	否	String	人员库信息备注，[0，40]个字符。 示例值：不含实习生
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 目前入参支持 “2.0”和“3.0“ 两个输入。 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 2020年11月26日后开通服务的账号仅支持输入“3.0”。

参数名称	必选	类型	描述
			不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0

3. 输出参数

参数名称	类型	描述
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员库ID不支持中文

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=腾讯深圳员工库
&GroupId=腾讯深圳员工库
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdIllegal",
"Message": "人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。"
},
"RequestId": "8125dda4-2905-4e02-88bd-79a93a660ad2"
}
}
```

示例2 错误示例-2

人员库ID不可重复

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=腾讯深圳员工库
&GroupId=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdAlreadyExist",
"Message": "人员库ID已经存在。人员库ID不可重复。"
},
"RequestId": "76ec6e41-37d6-4ab9-abef-48ef0c6ab175"
}
}
```

示例3 创建人员库接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=某某大学竹园宿舍楼1号楼
&GroupId=ZhuYuanDormitoryNo1
&FaceModelVersion=3.0
&Tag=全是女生哦
&GroupExDescriptions.0=学院名
&GroupExDescriptions.1=专业
&GroupExDescriptions.2=年级
&GroupExDescriptions.3=学号
&<公共请求参数>
```

输出示例

```
{
"Response": {
"FaceModelVersion": "3.0",
"RequestId": "1695f3dd-b668-479e-8b87-f37de371a8ec"
}
}
```

示例4 创建人员库接口-2

创建人员库

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateGroup
&GroupName=腾讯深圳员工库
&FaceModelVersion=3.0
&GroupId=TencentShenZhenEmployee
&Tag=不含实习生
&GroupExDescriptions.0=事业群
&GroupExDescriptions.1=部门名
```

```
&GroupExDescriptions.2=组名
&<公共请求参数>
```

输出示例

```
{
"Response": {
"FaceModelVersion": "3.0",
"RequestId": "e53ee4ec-9099-4b35-a129-21dd4820ff85"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。

错误码	描述
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

删除人员库

最近更新时间：2025-04-25 01:35:24

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

删除该人员库及包含的所有的人员。同时，人员对应的所有人脸信息将被删除。若某人员同时存在多个人员库中，该人员不会被删除，但属于该人员库中的自定义描述字段信息会被删除，属于其他人员库的自定义描述字段信息不受影响。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupId	是	String	人员库ID。

3. 输出参数

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

4. 示例

示例1 错误示例

人员库ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteGroup
&GroupId=ZhuYuanDormitory
&<公共请求参数>
```

输出示例

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

```
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "878b225f-d3a4-4aa0-a849-3df415ac61a9"
}
}
```

示例2 删除人员库接口

删除指定人员库及包含的所有的人员

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteGroup
&GroupId=ZhuYuanDormitoryNo1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "2ff7181c-ef20-4f02-b4ed-13924b381368"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

版权所有：腾讯云计算（北京）有限责任公司

- 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.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。

错误码	描述
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。

版权所有：腾讯云计算（北京）有限责任公司

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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员库信息

最近更新时间：2025-04-25 01:35:23

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取人员库信息。

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：GetGroupInfo。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupId	是	String	人员库 ID。

3. 输出参数

参数名称	类型	描述
GroupName	String	人员库名称
GroupId	String	人员库ID
GroupExDescriptions	Array of String	人员库自定义描述字段
Tag	String	人员库信息备注
FaceModelVersion	String	人脸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。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取人员库信息

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupInfo
&GroupId=aaa
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"GroupName": "aaa",
"GroupId": "aaa",
"GroupExDescriptions": [
"24"
],
"Tag": "fdsaf",
"FaceModelVersion": "3.0",
"CreationTimestamp": 1559530931613,
"RequestId": "6faa05d0-287b-464e-8755-cf3e8d0c703d"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
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。
ResourceUnavailable.StopUsing	账号已停服。
ResourcesSoldOut.ChargeStatusException	计费状态异常。

版权所有：腾讯云计算（北京）有限责任公司

获取人员库列表

最近更新时间：2025-04-25 01:35:23

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取人员库列表。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
Offset	否	Integer	起始序号，默认值为0 示例值：0
Limit	否	Integer	返回数量，默认值为10，最大值为1000 示例值：10

3. 输出参数

参数名称	类型	描述
GroupInfos	Array of GroupInfo	返回的人员库信息
GroupNum	Integer	人员库总数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例-2

返回数量不能大于1000

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupList
&Offset=0
&Limit=1001
&<公共请求参数>
```

输出示例


```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.LimitExceed",
"Message": "返回数量不在合法范围内。"
},
"RequestId": "a6bb57de-773b-45e4-9e1c-c30ef48eba85"
}
}
```


示例2 获取人员库列表接口

获取人员库列表

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetGroupList
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例


```
{
"Response": {
"GroupInfos": [
{
"GroupName": "腾讯深圳员工",
"GroupId": "TencentShenZhenEmployee",
"GroupExDescriptions": [
"事业群",
"部门名",
"组名"
],
"Tag": "不含实习生",
"FaceModelVersion": "3.0",
"CreationTimestamp": 1592210245686
},
{
"GroupName": "某某大学竹园宿舍楼1号楼",
"GroupId": "ZhuYuanDormitoryNo1",
"GroupExDescriptions": [
```

```
"学院名",
"专业",
"年级",
"学号"
],
"Tag": "全是女生哦",
"FaceModelVersion": "3.0",
"CreationTimestamp": 1594552117118
}
],
"GroupNum": 2,
"RequestId": "72102087-a18d-4225-9ae9-87ef49e9f63e"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。

错误码	描述
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

修改人员库

最近更新时间：2025-04-25 01:35:22

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

修改人员库名称、备注、自定义描述字段名称。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupId	是	String	人员库ID
GroupName	否	String	人员库名称
GroupExDescriptionInfos.N	否	Array of GroupExDescriptionInfo	需要修改的人员库自定义描述字段，key-value
Tag	否	String	人员库信息备注

3. 输出参数

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

4. 示例

示例1 错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=TencentShenZhenEmployee
&GroupExDescriptionInfos.0.GroupExDescriptionIndex=0
&GroupExDescriptionInfos.0.GroupExDescription=部门
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.DuplicatedGroupDescription",
"Message": "同一人员库中自定义描述字段不可重复。"
},
"RequestId": "4b483679-3f08-4a7c-b827-4a6332dc030f"
}
}
```

示例2 错误示例-2

人员库名称不可重复

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=TencentShenZhenEmployee
&GroupName=某某大学竹园1号宿舍楼
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupNameAlreadyExist",
"Message": "人员库名称已经存在。人员库名称不可重复。"
},
"RequestId": "2dcce170-30cf-4078-9b3c-309362fbad21"
}
}
```

示例3 修改人员库接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=ZhuYuanDormitoryNo1
&GroupName=某某大学竹园1号宿舍楼
&Tag=研究生入住
&GroupExDescriptionInfos.0.GroupExDescriptionIndex=0
&GroupExDescriptionInfos.0.GroupExDescription=学院
&Version=2018-03-01
&<公共请求参数>
```

版权所有：腾讯云计算（北京）有限责任公司

输出示例

```
{
"Response": {
"RequestId": "da051f16-0400-49d4-9832-2cfd3b51866f"
}
}
```

示例4 修改人员库接口-2

修改人员库名称、备注、自定义描述字段名称

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyGroup
&GroupId=TencentShenZhenEmployee
&GroupName=腾讯深圳在职人员库
&Tag=包含实习生
&GroupExDescriptionInfos.0.GroupExDescriptionIndex=1
&GroupExDescriptionInfos.0.GroupExDescription=部门
&GroupExDescriptionInfos.1.GroupExDescriptionIndex=2
&GroupExDescriptionInfos.1.GroupExDescription=项目组
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "98381738-509f-4749-8a66-dbf54247333d"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

创建人员

最近更新时间：2025-04-25 01:35:24

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

创建人员，添加人脸、姓名、性别及其他相关信息。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
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。
Gender	否	Integer	0代表未填写，1代表男性，2代表女性。
PersonExDescriptionInfos.N	否	Array of PersonExDescriptionInfo	人员描述字段内容，key-value。[0，60]个字符，可修改，可重复。
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。
UniquePersonControl	否	Integer	此参数用于控制判断 Image 或 Url 中图片包含的人脸，是否在人员库中已有疑似的同一人。 如果判断为已有相同人在人员库中，则不会创建新的人员，返回疑似同一人的人员信息。 如果判断没有，则完成创建人员。

参数名称	必选	类型	描述
			0: 不进行判断，无论是否有疑似同一人在库中均完成入库； 1:较低的同一人判断要求（百一误识别率）； 2: 一般的同一人判断要求（千一误识别率）； 3: 较高的同一人判断要求（万一误识别率）； 4: 很高的同一人判断要求（十万一误识别率）。 默认 0。 注： 要求越高，则疑似同一人的概率越小。不同要求对应的误识别率仅为参考值，您可以根据实际情况调整。
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。

3. 输出参数

参数名称	类型	描述
FaceId	String	人脸图片唯一标识。
FaceRect	FaceRect	检测出的人脸框的位置。 注意：此字段可能返回 null，表示取不到有效值。
SimilarPersonId	String	疑似同一人的PersonId。 当 UniquePersonControl 参数不为0且人员库中有疑似的同一人，此参数才有意义。
FaceModelVersion	String	人脸识别所用的算法模型版本。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员ID不可重复

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=ZhuYuanDormitoryNo1
```

```
&PersonName=Cheng
&PersonId=1001
&Gender=2
&Url=http://test.image.myqcloud.com/testD.jpg
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonIdAlreadyExist",
"Message": "人员ID已经存在。人员ID不可重复。"
},
"RequestId": "76dedef7-af27-4a20-9064-c5ef3133926d"
}
}
```

示例2 错误示例-2

人员库ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=ShenZhenCitizen
&PersonName=Siccy
&PersonId=3001
&Gender=2
&Url=http://test.image.myqcloud.com/testC.jpg
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "dfa512fc-fd07-4bf1-a292-cb497b620857"
}
}
```

示例3 创建人员接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=ZhuYuanDormitoryNo1
&PersonName=Junly
&PersonId=2001
&Gender=1
&Url=http://test.image.myqcloud.com/testB.jpg
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=0
&PersonExDescriptionInfos.0.PersonExDescription=计算机学院
&PersonExDescriptionInfos.1.PersonExDescriptionIndex=1
&PersonExDescriptionInfos.1.PersonExDescription=软件工程
&PersonExDescriptionInfos.2.PersonExDescriptionIndex=2
&PersonExDescriptionInfos.2.PersonExDescription=15级
&PersonExDescriptionInfos.3.PersonExDescriptionIndex=3
&PersonExDescriptionInfos.3.PersonExDescription=3150808
&Version=2018-03-01
&<公共请求参数>
```

输出示例


```
{
"Response": {
"FaceId": "3454816969590585885",
"SimilarPersonId": "",
"FaceRect": {
"X": 172,
"Y": 122,
"Width": 178,
"Height": 228
},
"FaceModelVersion": "3.0",
"RequestId": "738a2fb0-bd4b-4cf6-9c91-18d50978d76b"
}
}
```


示例4 创建人员接口-2

创建人员

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreatePerson
&GroupId=TencentShenZhenEmployee
&PersonName=evanliao
&PersonId=1001
&Gender=1
&Url=http://test.image.myqcloud.com/testA.jpg
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=0
&PersonExDescriptionInfos.0.PersonExDescription=云与智慧产业事业群
&PersonExDescriptionInfos.1.PersonExDescriptionIndex=1
```

```
&PersonExDescriptionInfos.1.PersonExDescription=人工智能产品中心
&PersonExDescriptionInfos.2.PersonExDescriptionIndex=2
&PersonExDescriptionInfos.2.PersonExDescription=人脸识别产品组
&Version=2018-03-01
&<公共请求参数>
```

输出示例


```
{
"Response": {
"FaceId": "3454816969590585881",
"SimilarPersonId": "",
"FaceRect": {
"X": 172,
"Y": 122,
"Width": 178,
"Height": 228
},
"FaceModelVersion": "3.0",
"RequestId": "269a6bc0-0016-47e8-a63d-517ce4052b0e"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UniquePersonControlIllegal	UniquePersonControl参数不合法。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded	超过配额限制。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

删除人员

最近更新时间：2025-04-25 01:35:23

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

删除该人员信息，此操作会导致所有人员库均删除此人员。同时，该人员的所有人脸信息将被删除。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
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，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePerson
&PersonId=3001
&Version=2018-03-01
&<公共请求参数>
```

输出示例

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

```
"Error": {
"Code": "ResourceNotFound.ErrorPersonNotExisted",
"Message": "个体不存在。"
},
"RequestId": "4b9c1bdc-86eb-413f-8c74-15bcf4466cd5"
}
}
```

示例2 删除人员接口

删除指定人员信息

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePerson
&PersonId=2001
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "4b609b7c-7a9a-4ea0-be8b-1fb638c2ceca"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员库删除人员

最近更新时间：2025-04-25 01:35:23

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

从某人员库中删除人员，此操作仅影响该人员库。若该人员仅存在于指定的人员库中，该人员将被删除，其所有的人脸信息也将被删除。

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：DeletePersonFromGroup。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员ID
GroupId	是	String	人员库ID

3. 输出参数

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

4. 示例

示例1 错误示例

人员库中不包含指定人员

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePersonFromGroup
&PersonId=3001
&GroupId=ZhuYuanDormitoryNo1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.GroupPersonMapNotExist",
"Message": "组中不包含对应的人员Id。"
},
"RequestId": "b9d9641d-3de2-4a6f-a728-36a3b66e7bd0"
}
}
```

示例2 人员库删除人员接口

从某人员库中删除人员，此操作仅影响该人员库

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeletePersonFromGroup
&PersonId=1001
&GroupId=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "f9b22bbc-c94d-4287-a2ab-fa68f7bbbaf3"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。

错误码	描述
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。

错误码	描述
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员归属信息

最近更新时间：2025-04-25 01:35:23

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员的信息，包括加入的人员库、描述内容等。

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：GetPersonGroupInfo。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员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100

3. 输出参数

参数名称	类型	描述
PersonGroupInfos	Array of PersonGroupInfo	包含此人员的人员库及描述字段内容列表
GroupNum	Integer	人员库总数量 注意：此字段可能返回 null，表示取不到有效值。
FaceModelVersion	String	人脸识别服务所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonGroupInfo
&PersonId=1002
&Offset=0
&Limit=10
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonIdNotExist",
"Message": "人员ID不存在。"
},
"RequestId": "98330f25-eb2e-432a-a30c-3830774210c1"
}
}
```

示例2 获取人员归属信息接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonGroupInfo
&PersonId=2001
&Offset=0
&Limit=10
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonGroupInfos": [
{
"GroupId": "ZhuYuanDormitoryNo1",
"PersonExDescriptions": [
"计算机学院",
"软件工程",
"15级",
"3150108080"
]
}
],
"RequestId": "671f0a1d-2b35-47c4-b9d1-b18053f71a04"
```

```
}
}
```

示例3 获取人员归属信息接口-2

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonGroupInfo
&PersonId=1001
&Offset=0
&Limit=10
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonGroupInfos": [
{
"GroupId": "TencentShenZhenEmployee",
"PersonExDescriptions": [
"云与智慧产业事业群",
"大数据及人工智能产品中心",
"人脸识别产品组"
]
}
],
"RequestId": "b7a505ad-4a85-42da-97b3-886c7999fb76"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee

- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee
- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员基础信息

最近更新时间：2025-04-25 01:35:23

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员的信息，包括姓名、性别、人脸等。

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：GetPersonBaseInfo。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员ID

3. 输出参数

参数名称	类型	描述
PersonName	String	人员名称
Gender	Integer	人员性别
FaceIds	Array of String	包含的人脸 ID 列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonBaseInfo
&PersonId=1002
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonIdNotExist",
"Message": "人员ID不存在。"
},
"RequestId": "98b4a0bc-802b-4764-9701-bc0c6c544395"
}
}
```

示例2 获取人员基础信息接口

获取指定人员的信息，包括姓名、性别、人脸等

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonBaseInfo
&PersonId=1001
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonName": "EvanLiao",
"Gender": 1,
"FaceIds": [
"2873640802022644880",
"2875186538564559728"
],
"RequestId": "9568a077-0710-40d2-9d6a-b9483d3f2051"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。

错误码	描述
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员列表

最近更新时间：2025-04-25 01:35:22

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员库中的人员列表。

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：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupId	是	String	人员库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	该人员库的人员数量 注意：此字段可能返回 null，表示取不到有效值。
FaceNum	Integer	该人员库的人脸数量 注意：此字段可能返回 null，表示取不到有效值。
FaceModelVersion	String	人脸识别所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员库ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonList
&GroupId=ZhuYuanDormitory
&Offset=0
&Limit=10
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "b7c0cd81-d621-465f-8fd6-86a6b49e67be"
}
}
```

示例2 获取人员列表接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonList
&GroupId=ZhuYuanDormitoryNo1
&Offset=0
&Limit=10
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonInfos": [
{
"PersonName": "Junly",
"PersonId": "2001",
"Gender": 1,
"PersonExDescriptions": [
"计算机学院",
"软件工程",
"15级",
"3150108080"
],
"FaceIds": [
```

```
"2877244861637985315"
]
}
],
"PersonNum": 1,
"FaceNum": 1,
"RequestId": "337d1bb5-3f88-4681-9291-5194f478f0d1"
}
}
```

示例3 获取人员列表接口-2

获取指定人员库中的人员列表

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonList
&GroupId=TencentShenZhenEmployee
&Offset=0
&Limit=10
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonInfos": [
{
"PersonName": "evanliao",
"PersonId": "1001",
"Gender": 1,
"PersonExDescriptions": [
"云与智慧产业事业群",
"大数据及人工智能产品中心",
"人脸识别产品组"
],
"FaceIds": [
"2877242150180891493"
]
}
],
"PersonNum": 1,
"FaceNum": 1,
"RequestId": "aa292f16-27d9-423b-9048-cdd43f6e4156"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。

错误码	描述
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
InvalidParameterValue.PersonIdAlreadyExist	人员ID已经存在。人员ID不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

获取人员列表长度

最近更新时间：2025-04-25 01:35:22

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

获取指定人员库中人员数量。

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：GetPersonListNum。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupId	是	String	人员库ID

3. 输出参数

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

4. 示例

示例1 错误示例

人员库ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonListNum
&GroupId=Tencent
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.GroupIdNotExist",
"Message": "人员库ID不存在。"
},
"RequestId": "2865a458-4db1-46bd-ae52-b7507c9f4e62"
}
}
```

示例2 获取人员列表长度接口

获取指定人员库中人员数量

输入示例

```
https://iai.tencentcloudapi.com/?Action=GetPersonListNum
&GroupId=TencentShenZhenEmployee
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"PersonNum": 2,
"FaceNum": 2,
"RequestId": "18fb1e31-0d33-4371-9f41-b1fd59e0c781"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Java: GitHub, Gitee
- Tencent Cloud SDK 3.0 for PHP: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Go: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Node.js: GitHub, Gitee

版权所有：腾讯云计算（北京）有限责任公司

- Tencent Cloud SDK 3.0 for .NET: GitHub, Gitee
- Tencent Cloud SDK 3.0 for C++: GitHub, Gitee
- Tencent Cloud SDK 3.0 for Ruby: GitHub, Gitee

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

错误码	描述
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

修改人员描述信息

最近更新时间：2025-04-25 01:35:22

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

修改指定人员库人员描述内容。

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：ModifyPersonGroupInfo。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupId	是	String	人员库ID
PersonId	是	String	人员ID
PersonExDescriptionInfos.N	是	Array of PersonExDescriptionInfo	需要修改的人员描述字段内容，key-value

3. 输出参数

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

4. 示例

示例1 错误示例

下标不合法

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonGroupInfo
&GroupId=TencentShenZhenEmployee
&PersonId=1001
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=-1
&PersonExDescriptionInfos.0.PersonExDescription=人工智能实验室
```

```
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameter",
"Message": "The value type of parameter `PersonExDescriptionInfos.0.PersonExDescriptionIndex` is not vali
d."
},
"RequestId": "f944657d-6260-4a2e-a4d8-9c1845f39e56"
}
}
```

示例2 修改人员描述信息接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonGroupInfo
&GroupId=ZhuYuanDormitoryNo1
&PersonId=2001
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=3
&PersonExDescriptionInfos.0.PersonExDescription=3150108080
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "77e11a2d-0fc0-4fba-ab25-016e998221c3"
}
}
```

示例3 修改人员描述信息接口-2

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonGroupInfo
&GroupId=TencentShenZhenEmployee
&PersonId=1001
&PersonExDescriptionInfos.0.PersonExDescriptionIndex=1
&PersonExDescriptionInfos.0.PersonExDescription=大数据及人工智能产品中心
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "77e11a2d-0fc0-4fba-ab25-016e998221c3"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageFacedetectFailed	人脸检测失败。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

修改人员基础信息

最近更新时间：2025-04-25 01:35:22

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

修改人员信息，包括名称、性别等。人员名称和性别修改会同步到包含该人员的所有人员库。

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：ModifyPersonBaseInfo。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员ID
PersonName	否	String	需要修改的人员名称
Gender	否	Integer	需要修改的人员性别

3. 输出参数

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

4. 示例

示例1 错误示例

人员性别设置错误

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonBaseInfo
&PersonId=1001
&Gender=-1
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.PersonGenderIllegal",
"Message": "人员性别设置出错。0代表未填写，1代表男性，2代表女性。"
},
"RequestId": "f40bf659-0283-4773-abe8-945f643d5015"
}
}
```

示例2 修改人员基础信息接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonBaseInfo
&PersonId=2001
&PersonName=JunlyWang
&Gender=1
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "6e2a0fb7-7c9e-45c7-abf1-338457d1def7"
}
}
```

示例3 修改人员基础信息接口-2

输入示例

```
https://iai.tencentcloudapi.com/?Action=ModifyPersonBaseInfo
&PersonId=1001
&PersonName=EvanLiao
&Gender=1
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "6e2a0fb7-7c9e-45c7-abf1-338457d1def7"
```

```
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

复制人员

最近更新时间：2025-04-25 01:35:24

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

将已存在于某人员库的人员复制到其他人员库，该人员的描述信息不会被复制。单个人员最多只能同时存在100个人员库中。

- 注：若该人员创建时算法模型版本为2.0，复制到非2.0算法模型版本的Group中时，复制操作将会失败。

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：CopyPerson。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员ID
GroupIds.N	是	Array of String	待加入的人员库列表

3. 输出参数

参数名称	类型	描述
SucGroupNum	Integer	成功加入的人员库数量
SucGroupIds	Array of String	成功加入的人员库列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=CopyPerson
&PersonId=1001
```

```
&GroupIds.0=TencentShenZhenEmployee
&GroupIds.1=ZhuYuanDormitoryNo1
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.GroupPersonMapExist",
"Message": "组中已包含对应的人员Id。"
},
"RequestId": "8ecfa566-68a9-4eb5-a6da-95ba9be26947"
}
}
```

示例2 复制人员接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=CopyPerson
&PersonId=2001
&GroupIds.0=TencentShenZhenEmployee
&GroupIds.1=ShenZhenCitizen
&<公共请求参数>
```

输出示例

```
{
"Response": {
"SucGroupNum": 2,
"SucGroupIds": [
"ShenZhenCitizen",
"TencentShenZhenEmployee"
],
"RequestId": "9841c008-e597-4356-a8d5-c47a246abd8d"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。

错误码	描述
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonNameIllegal	人员名称包含非法字符。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

增加人脸

最近更新时间：2025-04-25 01:35:24

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

将一组人脸图片添加到一个人员中。一个人员最多允许包含 5 张图片。若该人员存在多个人员库中，所有人员库中该人员图片均会增加。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：CreateFace。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员ID。 示例值：1001
Images.N	否	Array of String	图片 base64 数据，base64 编码后大小不可超过5M。 人员人脸总数量不可超过5张。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
Urls.N	否	Array of 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。 人员人脸总数量不可超过5张。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 示例值：["http://test.image.myqcloud.com/testA.jpg"]
FaceMatchThreshold	否	Float	只有和该人员已有的人脸相似度超过FaceMatchThreshold值的人脸，才能增加人脸成功。 默认值60分。取值范围[0,100] 。 示例值：60
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况；

参数名称	必选	类型	描述
			3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
SucFaceNum	Integer	加入成功的人脸数量 示例值：1
SucFaceIds	Array of String	加入成功的人脸ID列表 示例值：["2875186538564559728"]
RetCode	Array of Integer	每张人脸图片添加结果，-1101 代表未检测到人脸，-1102 代表图片解码失败，-1109 代表图片尺寸过大或者过小， -1601代表不符合图片质量控制要求, -1604 代表人脸相似度没有超过 FaceMatchThreshold。 其他非 0 值代表算法服务异常。 RetCode的顺序和入参中 Images 或 Urls 的顺序一致。 示例值：[0]
SucIndexes	Array of Integer	加入成功的人脸索引。索引顺序和入参中 Images 或 Urls 的顺序一致。 例， Urls 中 有 3 个 url，第二个 url 失败，则 SucIndexes 值为 [0,2] 。
SucFaceRects	Array of FaceRect	加入成功的人脸框位置。顺序和入参中 Images 或 Urls 的顺序一致。
FaceModelVersion	String	人脸识别所用的算法模型版本。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId ）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

已有人员最多再新增4张人脸图片

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateFace
&PersonId=1001
&Urls.0=http://test.image.myqcloud.com/testB.jpg
&Urls.1=http://test.image.myqcloud.com/testC.jpg
&Urls.2=http://test.image.myqcloud.com/testD.jpg
&Urls.3=http://test.image.myqcloud.com/testE.jpg
```

```
&Urls.4=http://test.image.myqcloud.com/testF.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "InvalidParameterValue.UploadFaceNumExceed",
"Message": "一次最多上传四张人脸。"
},
"RequestId": "44506f79-2191-49bc-997b-748a566d781c"
}
}
```

示例2 增加人脸接口

将一组人脸图片添加到一个人员中

输入示例

```
https://iai.tencentcloudapi.com/?Action=CreateFace
&PersonId=1001
&Urls.0=http://test.image.myqcloud.com/testA.jpg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"SucFaceNum": 1,
"SucFaceIds": [
"2875186538564559728"
],
"RetCode": [
0
],
"SucIndexes": [
0
],
"SucFaceRects": [
{
"X": 135,
"Y": 42,
"Width": 98,
"Height": 136
}
],
"FaceModelVersion": "2.0",
```

```
"RequestId": "07d63403-a199-4bbe-b9e0-692356ac738d"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。

错误码	描述
FailedOperation.ImageFacedetectFailed	人脸检测失败。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

删除人脸

最近更新时间：2025-04-25 01:35:24

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

删除一个人员下的人脸图片。如果该人员只有一张人脸图片，则返回错误。

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：DeleteFace。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	人员ID
FaceIds.N	是	Array of String	待删除的人脸ID列表

3. 输出参数

参数名称	类型	描述
SucDeletedNum	Integer	删除成功的人脸数量
SucFaceIds	Array of String	删除成功的人脸ID列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

人员ID不存在

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteFace
&PersonId=5001
&FaceIds.0=2875186538564559728
```

```
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "ResourceNotFound.ErrorPersonNotExisted",
"Message": "个体不存在。"
},
"RequestId": "e2b603e5-51d0-4a8a-8319-2df026ae7518"
}
}
```

示例2 删除人脸接口

删除一个人员下的人脸图片

输入示例

```
https://iai.tencentcloudapi.com/?Action=DeleteFace
&PersonId=1001
&FaceIds.0=2875186538564559728
&Version=2018-03-01
&<公共请求参数>
```

输出示例

```
{
"Response": {
"SucDeletedNum": 1,
"SucFaceIds": [
"2875186538564559728"
],
"RequestId": "38f96439-d5ea-48ea-899b-44a284708f6b"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名

错误码	描述
	称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。

错误码	描述
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸搜索相关接口
人脸搜索

最近更新时间：2025-04-25 01:35:21

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，识别结果按照相似度从大到小排序。

支持一次性识别图片中的最多 10 张人脸，支持一次性跨 100 个人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

与人员搜索及人员搜索按库返回接口不同的是，本接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行验证，而人员搜索及人员搜索按库返回接口 会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个Person下有4张 Face，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使搜索更加准确。

本接口需与人员库管理相关接口结合使用。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：SearchFaces。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限100个。 示例值：["TencentShenZhenEmployee"]
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。 示例值：http://test.image.myqcloud.com/testA.jpg

参数名称	必选	类型	描述
MaxFaceNum	否	Integer	最多识别的人脸数目。默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。 例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。默认为34。低于34的人脸图片无法被识别。建议设置为80。 示例值：40
MaxPersonNum	否	Integer	单张被识别的人脸返回的最相似人员数量。默认值为5，最大值为100。 例，设MaxFaceNum为1，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。 示例值：5
NeedPersonInfo	否	Integer	是否返回人员具体信息。0 为关闭，1 为开启。默认为 0。其他非0非1值默认为0 示例值：0
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有超过FaceMatchThreshold值的结果才会返回。默认为0。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Results	Array of Result	识别结果。
FaceNum	Integer	搜索的人员库中包含的人脸数。 示例值：1
FaceModelVersion	String	人脸识别所用的算法模型版本。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

MaxFaceNum不能大于10

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchFaces
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=11
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.SearchFacesExceed",
"Message": "检索人脸个数超过限制。"
},
"RequestId": "f04ba2f3-532c-4207-9caa-6e818ded7fb9"
}
}
```

示例2 错误示例-2

图片URL错误

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchFaces
&Url=http://test.image.myqcloud.com/testA
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDownloadError",
"Message": "图片下载错误。"
},
"RequestId": "527ecffe-4c6a-47c9-8217-4dd2e3f018da"
}
}
```

示例3 人脸搜索接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchFaces
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=1
&MinFaceSize=40
&MaxPersonNum=5
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例


```
{
"Response": {
"Results": [
{
"RetCode": 0,
"Candidates": [
{
"PersonId": "1001",
"FaceId": "2875093635484912302",
"PersonName": "Jerry",
"Gender": 0,
"Score": 100,
"PersonGroupInfos": [
{
"GroupId": "abc",
"PersonExDescriptions": [
"description1",
"description2"
]
}
]
}
],
"FaceRect": {
"X": 370,
"Y": 46,
"Width": 75,
"Height": 75
}
}
],
"FaceNum": 1,
"FaceModelVersion": "3.0",
"RequestId": "57b42b73-b978-45b9-8095-8f50e9642d35"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸搜索分库返回

最近更新时间：2025-04-25 01:35:21

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，按照**人员库的维度**以人员相似度从大到小顺序排列。

支持一次性识别图片中的最多 10 张人脸，支持跨人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

与人员搜索及人员搜索按库返回接口不同的是，本接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行验证，而人员搜索及人员搜索按库返回接口 会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个Person下有4张 Face，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使搜索更加准确。

本接口需与人员库管理相关接口结合使用。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：SearchFacesReturnsByGroup。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限60个。 示例值：["TencentShenZhenEmployee\r\n"]
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。 示例值：http://test.image.myqcloud.com/testA.jpg
MaxFaceNum	否	Integer	最多识别的人脸数目。默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。

参数名称	必选	类型	描述
			例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。默认为34。低于34将影响搜索精度。建议设置为80。 示例值：40
MaxPersonNumPerGroup	否	Integer	被检测到的人脸，对应最多返回的最相似人员数目。默认值为5，最大值为10。 例，设MaxFaceNum为3，MaxPersonNumPerGroup为5，GroupIds长度为3，则最多可能返回3*5=45个人员。 示例值：5
NeedPersonInfo	否	Integer	是否返回人员具体信息。0 为关闭，1 为开启。默认为 0。其他非0非1值默认为0 示例值：0
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有大于等于FaceMatchThreshold值的结果才会返回。 默认为0。 取值范围[0.0,100.0) 。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
FaceNum	Integer	搜索的人员库中包含的人脸数。 示例值：1
ResultsReturnsByGroup	Array of ResultsReturnsByGroup	识别结果。
FaceModelVersion	String	人脸识别所用的算法模型版本。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸搜索分库返回接口

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"MaxPersonNumPerGroup": "5",
"GroupIds": [
"TencentShenZhenEmployee"
],
"MaxFaceNum": "1",
"MinFaceSize": "40"
}
```

输出示例

```
{
"Response": {
"FaceNum": 1,
"ResultsReturnsByGroup": [
{
"GroupCandidates": [
{
"Candidates": [
{
"PersonGroupInfos": [
{
"GroupId": "abc",
"PersonExDescriptions": [
"description1",
"description2"
]
}
],
"PersonId": "1001",
"Gender": 0,
"PersonName": "张三",
"Score": 100,
"FaceId": "2875093635484912302"
}
],
"GroupId": "Group1"
}
],
"FaceRect": {
```

```
"Y": 46,
"X": 370,
"Height": 1,
"Width": 1
},
"RetCode": 0
}
],
"RequestId": "57b42b73-b978-45b9-8095-8f50e9642d35",
"FaceModelVersion": "3.0"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。

错误码	描述
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.InArrears	账号已欠费。

版权所有：腾讯云计算（北京）有限责任公司

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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员搜索

最近更新时间：2025-04-25 01:35:21

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，按照相似度从大到小排列。

支持一次性识别图片中的最多 10 张人脸，支持一次性跨 100 个人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

本接口会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个 Person 下有4张 Face ，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员搜索（确定待识别的人脸图片是某人）更加准确。而人脸搜索及人脸搜索按库返回接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行搜索。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。
- 仅支持算法模型版本（FaceModelVersion）为3.0的人员库。

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：SearchPersons。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限100个。 示例值：["TencentShenZhenEmployee\r\n"]
Image	否	String	图片 base64 数据，base64 编码后大小不可超过5M。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 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。 示例值：http://test.image.myqcloud.com/testA.jpg
MaxFaceNum	否	Integer	最多识别的人脸数目。默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。 例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面

参数名称	必选	类型	描述
			积最大的5张人脸。 示例值：1
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。默认为34。低于34将影响搜索精度。建议设置为80。 示例值：40
MaxPersonNum	否	Integer	单张被识别的人脸返回的最相似人员数量。默认值为5，最大值为100。 例，设MaxFaceNum为1，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。 示例值：3
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
FaceMatchThreshold	否	Float	出参Score中，只有大于等于FaceMatchThreshold值的结果才会返回。默认为0。取值范围[0.0,100.0) 。 示例值：0
NeedPersonInfo	否	Integer	是否返回人员具体信息。0 为关闭，1 为开启。默认为 0。其他非0非1值默认为0 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Results	Array of Result	识别结果。
PersonNum	Integer	搜索的人员库中包含的人员数。若输入图片中所有人脸均不符合质量要求，则返回0。 示例值：5
FaceModelVersion	String	人脸识别所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

图片URL错误

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersons
&Url=http://test.image.myqcloud.com/testA
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDownloadError",
"Message": "图片下载错误。"
},
"RequestId": "527ecffe-4c6a-47c9-8217-4dd2e3f018da"
}
}
```

示例2 人员搜索接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersons
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=2
&MinFaceSize=40
&MaxPersonNum=3
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Results": [
{
"RetCode": 0,
"Candidates": [
{
"PersonId": "person1",
"FaceId": "",
"Score": 100,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
```

```
"PersonId": "person2",
"FaceId": "",
"Score": 60,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
"PersonId": "person3",
"FaceId": "",
"Score": 50,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
}
],
"FaceRect": {
"X": 139,
"Y": 59,
"Width": 124,
"Height": 162
}
},
{
"RetCode": 0,
"Candidates": [
{
"PersonId": "person2",
"FaceId": "",
"Score": 100,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
"PersonId": "person1",
"FaceId": "",
"Score": 60,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
},
{
"PersonId": "person3",
"FaceId": "",
"Score": 20,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
}
```


版权所有：腾讯云计算（北京）有限责任公司

```
],
"FaceRect": {
"X": 328,
"Y": 70,
"Width": 119,
"Height": 162
}
}
],
"PersonNum": 5,
"FaceModelVersion": "3.0",
"RequestId": "c4608852-ff60-4a01-8dc3-367f6046baaf"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。

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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.ImageEmptyError	图片为空。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UrlIllegal	URL格式不合法。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.GetAuthInfoError	获取认证信息失败。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。

错误码	描述
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人员搜索按库返回

最近更新时间：2025-04-25 01:35:21

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对一张待识别的人脸图片，在一个或多个人员库中识别出最相似的 TopK 人员，按照**人员库的维度**以人员相似度从大到小顺序排列。

支持一次性识别图片中的最多 10 张人脸，支持跨人员库（Group）搜索。

单次搜索的人员库人脸总数量和人员库的算法模型版本（FaceModelVersion）相关。算法模型版本为2.0的人员库，单次搜索人员库人脸总数量不得超过 100 万张；算法模型版本为3.0的人员库，单次搜索人员库人脸总数量不得超过 300 万张。

本接口会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个 Person 下有4张 Face ，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员搜索（确定待识别的人脸图片是某人）更加准确。而人脸搜索及人脸搜索按库返回接口将该人员（Person）下的每个人脸（Face）都作为单独个体进行搜索。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。
- 仅支持算法模型版本（FaceModelVersion）为3.0的人员库。

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：SearchPersonsReturnsByGroup。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
GroupIds.N	是	Array of String	希望搜索的人员库列表，上限60个。
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。
MaxFaceNum	否	Integer	最多识别的人脸数目。默认值为1（仅检测图片中面积最大的那张人脸），最大值为10。 MaxFaceNum用于，当输入的待识别图片包含多张人脸时，设定要搜索的人脸的数量。

参数名称	必选	类型	描述
			例：输入的Image或Url中的图片包含多张人脸，设MaxFaceNum=5，则会识别图片中面积最大的5张人脸。
MinFaceSize	否	Integer	人脸长和宽的最小尺寸，单位为像素。默认为34。低于34将影响搜索精度。建议设置为80。
MaxPersonNumPerGroup	否	Integer	被检测到的人脸，对应最多返回的最相似人员数目。默认值为5，最大值为10。 例，设MaxFaceNum为3，MaxPersonNumPerGroup为5，GroupIds长度为3，则最多可能返回3*5*3=45个人员。
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。
FaceMatchThreshold	否	Float	出参Score中，只有超过FaceMatchThreshold值的结果才会返回。默认为0。
NeedPersonInfo	否	Integer	是否返回人员具体信息。0 为关闭，1 为开启。默认为 0。其他非0非1值默认为0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。

3. 输出参数

参数名称	类型	描述
PersonNum	Integer	搜索的人员库中包含的人员数。若输入图片中所有人脸均不符合质量要求，则返回0。
ResultsReturnsByGroup	Array of ResultsReturnsByGroup	识别结果。
FaceModelVersion	String	人脸识别所用的算法模型版本。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 错误示例

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersonsReturnsByGroup
&Url=http://test.image.myqcloud.com/testB.jpg
&MaxFaceNum=1
&MinFaceSize=40
```

```
&MaxPersonNumPerGroup=5
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Error": {
"Code": "FailedOperation.ImageDownloadError",
"Message": "图片下载错误。"
},
"RequestId": "527ecffe-4c6a-47c9-8217-4dd2e3f018da"
}
}
```

示例2 人员搜索分库返回接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=SearchPersonsReturnsByGroup
&Url=http://test.image.myqcloud.com/testA.jpg
&MaxFaceNum=1
&MinFaceSize=40
&MaxPersonNumPerGroup=5
&GroupIds.0=TencentShenZhenEmployee
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ResultsReturnsByGroup": [
{
"RetCode": 0,
"FaceRect": {
"X": 375,
"Y": 37,
"Width": 63,
"Height": 82
},
"GroupCandidates": [
{
"GroupId": "TencentShenZhenEmployee",
"Candidates": [
{
"PersonId": "testtest3",
"FaceId": "",
```

```
"Score": 100,
"PersonName": null,
"Gender": null,
"PersonGroupInfos": null
}
]
}
]
}
],
"PersonNum": 1,
"FaceModelVersion": "3.0",
"RequestId": "c8a06ec5-ecb4-4dd7-a8c8-ce5e675495c2"
}
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。

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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdNotExists	对应的人员库ID在库中不存在。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UrlIllegal	URL格式不合法。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸验证相关接口
人脸验证

最近更新时间：2025-04-25 01:35:20

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

给定一张人脸图片和一个 PersonId，判断图片中的人和 PersonId 对应的人是否为同一人。PersonId 请参考人员库管理相关接口。

与人脸比对接口不同的是，人脸验证用于判断 “此人是否是此人”，“此人”的信息已存于人员库中，“此人”可能存在多张人脸图片；而人脸比对用于判断两张人脸的相似度。

与人员验证接口不同的是，人脸验证将该人员（Person）下的每个人脸（Face）都作为单独个体进行验证，而人员验证会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个 Person下有4张 Face，人员验证接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员验证（确定待识别的人脸图片是某人员）更加准确。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

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：VerifyFace。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	待验证的人员ID。人员ID具体信息请参考人员库管理相关接口。 示例值：11111111
Image	否	String	图片 base64 数据，base64 编码后大小不可超过5M。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 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。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。 示例值：http://test.image.myqcloud.com/testA.jpg
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况；

参数名称	必选	类型	描述
			2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题； 默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。 示例值：0
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。 示例值：0

3. 输出参数

参数名称	类型	描述
Score	Float	给定的人脸图片与 PersonId 对应人脸的相似度。若 PersonId 下有多张人脸（Face），返回相似度最大的分数。 不同算法版本返回的相似度分数不同。 若需要验证两张图片中人脸是否为同一人，3.0版本误识率千分之一对应分数为40分，误识率万分之一对应分数为50分，误识率十万分之一对应分数为60分。 一般超过50分则可认定为同一人。 2.0版本误识率千分之一对应分数为70分，误识率万分之一对应分数为80分，误识率十万分之一对应分数为90分。 一般超过80分则可认定为同一人。 示例值：100
IsMatch	Boolean	是否为同一人判断，固定阈值分数为60分，若想更灵活地调整阈值可取Score参数返回进行判断 示例值：1
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸验证接口

输入示例

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg",
"PersonId": "11111111"
}
```

输出示例

```
{
"Response": {
"Score": 100,
"IsMatch": true,
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

人员验证

最近更新时间：2025-04-25 01:35:20

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

给定一张人脸图片和一个 PersonId，判断图片中的人和 PersonId 对应的人是否为同一人。PersonId 请参考人员库管理相关接口。
本接口会将该人员（Person）下的所有人脸（Face）进行融合特征处理，即若某个Person下有4张 Face，本接口会将4张 Face 的特征进行融合处理，生成对应这个 Person 的特征，使人员验证（确定待识别的人脸图片是某人员）更加准确。

和人脸比对相关接口不同的是，人脸验证相关接口用于判断 “此人是否是此人”，“此人”的信息已存于人员库中，“此人”可能存在多张人脸图片；而人脸比对相关接口用于判断两张人脸的相似度。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。
- 仅支持算法模型版本（FaceModelVersion）为3.0的人员库。

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：VerifyPerson。
Version	是	String	公共参数，本接口取值：2018-03-01。
Region	否	String	公共参数，本接口不需要传递此参数。
PersonId	是	String	待验证的人员ID。人员ID具体信息请参考人员库管理相关接口。
Image	否	String	图片 base64 数据。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
Url	否	String	图片的 Url 。图片的 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。 若图片中包含多张人脸，只选取其中人脸面积最大的人脸。 支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
QualityControl	否	Integer	图片质量控制。 0: 不进行控制； 1:较低的质量要求，图像存在非常模糊，眼睛鼻子嘴巴遮挡至少其中一种或多种的情况； 2: 一般的质量要求，图像存在偏亮，偏暗，模糊或一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，至少其中三种的情况； 3: 较高的质量要求，图像存在偏亮，偏暗，一般模糊，眉毛遮挡，脸颊遮挡，下巴遮挡，其中一到两种的情况； 4: 很高的质量要求，各个维度均为最好或最多在某一维度上存在轻微问题；

参数名称	必选	类型	描述
			默认 0。 若图片质量不满足要求，则返回结果中会提示图片质量检测不符要求。
NeedRotateDetection	否	Integer	是否开启图片旋转识别支持。0为不开启，1为开启。默认为0。本参数的作用为，当图片中的人脸被旋转且图片没有exif信息时，如果不开启图片旋转识别支持则无法正确检测、识别图片中的人脸。若您确认图片包含exif信息或者您确认输入图中人脸不会出现被旋转情况，请不要开启本参数。开启后，整体耗时将可能增加数百毫秒。

3. 输出参数

参数名称	类型	描述
Score	Float	给定的人脸照片与 PersonId 对应的相似度。若 PersonId 下有多张人脸（Face），会融合多张人脸信息进行验证。
IsMatch	Boolean	是否为同一人的判断。
FaceModelVersion	String	人脸识别所用的算法模型版本。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸验证接口

输入示例

```
https://iai.tencentcloudapi.com/?Action=VerifyPerson
&Url=http://test.image.myqcloud.com/testA.jpg
&PersonId=11111111
&<公共请求参数>
```

输出示例

```
{
"Response": {
"Score": 100,
"IsMatch": true,
"FaceModelVersion": "3.0",
"RequestId": "a8eb4545-a154-4f86-9510-57a8be9cae0c"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
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。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UrlIllegal	URL格式不合法。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.GetAuthInfoError	获取认证信息失败。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸静态活体检测相关接口
人脸静态活体检测

最近更新时间：2025-04-25 01:35:20

1. 接口描述

接口请求域名： iai.tencentcloudapi.com 。

用于对用户上传的静态图片进行人脸活体检测。与动态活体检测的区别是：静态活体检测中，用户不需要通过唇语或摇头眨眼等动作来识别。

静态活体检测适用于手机自拍的场景，或对防攻击要求不高的场景。如果对活体检测有更高安全性要求，请使用人脸核身 · 云智慧眼产品。

- 图片的宽高比请接近3：4，不符合宽高比的图片返回的分值不具备参考意义。本接口适用于类手机自拍场景，非类手机自拍照返回的分值不具备参考意义。

- 使用过程中建议正对摄像头，不要距离太远，使面部可以完整地显示在识别的框内，识别过程中不要移动设备或遮挡面部。不要选择光线过强或过弱的环境进行面部识别，识别时不要添加任何滤镜。

- 公共参数中的签名方式请使用V3版本，即配置SignatureMethod参数为TC3-HMAC-SHA256。

调用次数计费（资源包）

资源包规格	100万次	1000万次	5000万次	1亿次
人脸静态活体检测	1800元	17400元	75000元	110000元

调用次数计费（后付费）

月接口调用总量	0<调用量≤300万	300万<调用量≤1500万	调用量>1500万
人脸静态活体检测	0.0019元/次	0.0017元/次	0.0013元/次

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：DetectLiveFace。

参数名称	必选	类型	描述
Version	是	String	公共参数，本接口取值：2018-03-01。
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（图片的宽高比请接近3:4，不符合宽高比的图片返回的分值不具备参考意义）。 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。 （图片的宽高比请接近 3:4，不符合宽高比的图片返回的分值不具备参考意义） 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。 示例值：http://test.image.myqcloud.com/testA.jpg
FaceModelVersion	否	String	人脸识别服务所用的算法模型版本。 目前入参支持 “2.0”和“3.0 “ 两个输入。 2020年4月2日开始，默认为“3.0”，之前使用过本接口的账号若未填写本参数默认为“2.0”。 2020年11月26日后开通服务的账号仅支持输入“3.0”。 不同算法模型版本对应的人脸识别算法不同，新版本的整体效果会优于旧版本，建议使用“3.0”版本。 示例值：3.0

3. 输出参数

参数名称	类型	描述
Score	Float	活体打分，取值范围 [0,100]，分数一般落于[80, 100]区间内，0分也为常见值。推荐相大于 87 时可判断为活体。可根据具体场景自行调整阈值。 本字段当且仅当FaceModelVersion为2.0时才具备参考意义。 示例值：99
FaceModelVersion	String	人脸识别所用的算法模型版本。 示例值：3.0
IsLiveness	Boolean	活体检测是否通过。 本字段只有FaceModelVersion为3.0时才具备参考意义。 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 人脸静态活体检测接口

输入示例

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

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

{
"Url": "http://test.image.myqcloud.com/testA.jpg"
}
```

输出示例

```
{
"Response": {
"Score": 99,
"IsLiveness": true,
"FaceModelVersion": "3.0",
"RequestId": "048a5eb3-f7a9-4c1f-a982-00e90b6b892c"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
AuthFailure.InvalidAuthorization	认证失败。

错误码	描述
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Delivering	资源正在发货中。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
ResourceUnavailable.Freeze	账号已被冻结。
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。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

数据结构

最近更新时间：2024-11-22 11:48:46

AttributeItem

人脸属性信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Type	Integer	属性值
Probability	Float	Type识别概率值,[0,1],代表判断正确的概率。

Candidate

识别出的最相似候选人

被如下接口引用：SearchFaces, SearchFacesReturnsByGroup, SearchPersons, SearchPersonsReturnsByGroup。

名称	类型	描述
PersonId	String	人员ID 示例值：person1
FaceId	String	人脸ID，仅在SearchFaces/SearchFacesReturnsByGroup接口返回时有效。人员搜索类接口采用融合特征方式检索，该字段无意义 示例值：3820314501007076807
Score	Float	候选者的匹配得分。 1万大小人脸底库下，误识率百分之一对应分数为70分，误识率千分之一对应分数为80分，误识率万分之一对应分数为90分； 10万大小人脸底库下，误识率百分之一对应分数为80分，误识率千分之一对应分数为90分，误识率万分之一对应分数为100分； 30万大小人脸底库下，误识率百分之一对应分数为85分，误识率千分之一对应分数为95分。 一般80分左右可适用大部分场景，建议分数不要超过90分。您可以根据实际情况选择合适的分数。 示例值：50
PersonName	String	人员名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：hello
Gender	Integer	人员性别 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
PersonGroupInfos	Array of PersonGroupInfo	包含此人员的人员库及描述字段内容列表 注意：此字段可能返回 null，表示取不到有效值。

DenseFaceShape

稠密关键点详细信息

被如下接口引用：AnalyzeDenseLandmarks。

名称	类型	描述
X	Integer	人脸框左上角横坐标。
Y	Integer	人脸框左上角纵坐标。
Width	Integer	人脸框宽度。
Height	Integer	人脸框高度。
LeftEye	Array of Point	描述左侧眼睛轮廓的 XX 点。
RightEye	Array of Point	描述右侧眼睛轮廓的 XX 点。
LeftEyeBrow	Array of Point	描述左侧眉毛轮廓的 XX 点。
RightEyeBrow	Array of Point	描述右侧眉毛轮廓的 XX 点。
MouthOutside	Array of Point	描述外嘴巴轮廓的 XX 点， 从左侧开始逆时针返回。
MouthInside	Array of Point	描述内嘴巴轮廓的 XX 点，从左侧开始逆时针返回。
Nose	Array of Point	描述鼻子轮廓的 XX 点。
LeftPupil	Array of Point	左瞳孔轮廓的 XX 个点。
RightPupil	Array of Point	右瞳孔轮廓的 XX 个点。
CentralAxis	Array of Point	中轴线轮廓的 XX 个点。
Chin	Array of Point	下巴轮廓的 XX 个点。
LeftEyeBags	Array of Point	左眼袋的 XX 个点。
RightEyeBags	Array of Point	右眼袋的 XX 个点。
Forehead	Array of Point	额头的 XX 个点。

Eye

眼睛信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Glass	AttributeItem	识别是否佩戴眼镜。 AttributeItem对应的Type为 —— 0：无眼镜，1：普通眼镜，2：墨镜
EyeOpen	AttributeItem	识别眼睛的睁开、闭合状态。 AttributeItem对应的Type为 —— 0：睁开，1：闭眼
EyelidType	AttributeItem	识别是否双眼皮。 AttributeItem对应的Type为 —— 0：无，1：有。
EyeSize	AttributeItem	眼睛大小。 AttributeItem对应的Type为 —— 0：小眼睛，1：普通眼睛，2：大眼睛。

Eyebrow

眉毛信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
EyebrowDensity	AttributeItem	眉毛浓密。 AttributeItem对应的Type为 —— 0：淡眉，1：浓眉。
EyebrowCurve	AttributeItem	眉毛弯曲。 AttributeItem对应的Type为 —— 0：不弯，1：弯眉。
EyebrowLength	AttributeItem	眉毛长短。 AttributeItem对应的Type为 —— 0：短眉毛，1：长眉毛。

FaceAttributesInfo

人脸属性信息，包含性别(gender)、年龄(age)、表情(expression)、
魅力(beauty)、眼镜(glass)、口罩（ mask ）、头发（ hair ）和姿态 (pitch，roll，yaw)。只有当 NeedFaceAttributes 设为 1 时才返回有效信息，最多返回面积最大的 5 张人脸属性信息，超过 5 张人脸（第 6 张及以后的人脸）的 FaceAttributesInfo 不具备参考意义。

被如下接口引用：DetectFace。

名称	类型	描述
Gender	Integer	性别[0~49]为女性，[50，100]为男性，越接近0和100表示置信度越高。 NeedFaceAttributes 不为 1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Age	Integer	年龄 [0~100]。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Expression	Integer	微笑[0(normal，正常))~100(laugh，大笑)]。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Glass	Boolean	是否有眼镜 [true,false]。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Pitch	Integer	上下偏移[-30,30]，单位角度。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 建议：人脸入库选择[-10,10]的图片。
Yaw	Integer	左右偏移[-30,30]，单位角度。 NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 建议：人脸入库选择[-10,10]的图片。
Roll	Integer	平面旋转[-180,180]，单位角度。 NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。 建议：人脸入库选择[-20,20]的图片。
Beauty	Integer	魅力[0~100]。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Hat	Boolean	是否有帽子 [true,false]。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。
Mask	Boolean	是否有口罩 [true,false]。NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。
Hair	FaceHairAttributesInfo	头发信息，包含头发长度（ length ）、有无刘海（ bang ）、头发颜色（ color ）。 NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	描述
EyeOpen	Boolean	双眼是否睁开 [true,false]。只要有超过一只眼睛闭眼，就返回false。 NeedFaceAttributes 不为1 或检测超过 5 张人脸时，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。

FaceDetailAttributesInfo

人脸属性信息，根据 FaceAttributesType 输入的类型，返回年龄（Age）、颜值（Beauty）
情绪（Emotion）、眼睛信息（Eye）、眉毛（Eyebrow）、性别（Gender）
头发（Hair）、帽子（Hat）、姿态（Headpose）、口罩（Mask）、嘴巴（Mouse）、胡子（Moustache）
鼻子（Nose）、脸型（Shape）、肤色（Skin）、微笑（Smile）等人脸属性信息。
若 FaceAttributesType 没有输入相关类型，则FaceDetaiAttributesInfo返回的细项不具备参考意义。

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Age	Integer	年龄 [0,65]，其中65代表“65岁及以上”。 FaceAttributesType 不含Age 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Beauty	Integer	美丑打分[0,100]。 FaceAttributesType 不含 Beauty 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Emotion	AttributeItem	情绪，可识别自然、高兴、惊讶、生气、悲伤、厌恶、害怕。 AttributeItem对应的Type为 —— 0：自然，1：高兴，2：惊讶，3：生气，4：悲伤，5：厌恶，6：害怕 FaceAttributesType 不含Emotion 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Eye	Eye	眼睛相关信息，可识别是否戴眼镜、是否闭眼、是否双眼皮和眼睛大小。 FaceAttributesType 不含Eye 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Eyebrow	Eyebrow	眉毛相关信息，可识别眉毛浓密、弯曲、长短信息。 FaceAttributesType 不含Eyebrow 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Gender	AttributeItem	性别信息。 AttributeItem对应的Type为 —— 0：男性，1：女性。 FaceAttributesType 不含Gender 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Hair	Hair	头发信息，包含头发长度、有无刘海、头发颜色。 FaceAttributesType 不含Hair 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Hat	Hat	帽子信息，可识别是否佩戴帽子、帽子款式、帽子颜色。 FaceAttributesType 不含Hat 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
HeadPose	HeadPose	姿态信息，包含人脸的上下偏移、左右偏移、平面旋转信息。 FaceAttributesType 不含Headpose 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Mask	AttributeItem	口罩佩戴信息。 AttributeItem对应的Type为 —— 0: 无口罩， 1: 有口罩不遮脸，2: 有口罩遮下巴，3: 有口罩遮嘴，4: 正确佩戴口罩。 FaceAttributesType 不含Mask 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Mouth	Mouth	嘴巴信息，可识别是否张嘴、嘴唇厚度。 FaceAttributesType 不含 Mouth 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Moustache	AttributeItem	胡子信息。 AttributeItem对应的Type为 —— 0：无胡子，1：有胡子。 FaceAttributesType 不含 Moustache 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Nose	AttributeItem	鼻子信息。 AttributeItem对应的Type为 —— 0：朝天鼻，1：鹰钩鼻，2：普通，3：圆鼻头

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	描述
		FaceAttributesType 不含 Nose 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Shape	AttributeItem	脸型信息。 AttributeItem对应的Type为 —— 0：方脸，1：三角脸，2：鹅蛋脸，3：心形脸，4：圆脸。 FaceAttributesType 不含 Shape 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Skin	AttributeItem	肤色信息。 AttributeItem对应的Type为 —— 0：黄色皮肤，1：棕色皮肤，2：黑色皮肤，3：白色皮肤。 FaceAttributesType 不含 Skin 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。
Smile	Integer	微笑程度，[0,100]。 FaceAttributesType 不含 Smile 或检测超过 5 张人脸时，此参数仍返回，但不具备参考意义。

FaceDetailInfo

人脸信息列表。

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
FaceRect	FaceRect	检测出的人脸框位置。
FaceDetailAttributesInfo	FaceDetailAttributesInfo	人脸属性信息，根据 FaceAttributesType 输入的类型，返回年龄（Age）、颜值（Beauty） 情绪（Emotion）、眼睛信息（Eye）、眉毛（Eyebrow）、性别（Gender） 头发（Hair）、帽子（Hat）、姿态（Headpose）、口罩（Mask）、嘴巴（Mouth）、胡子（Moustache） 鼻子（Nose）、脸型（Shape）、肤色（Skin）、微笑（Smile）等人脸属性信息。 若 FaceAttributesType 没有输入相关类型，则 FaceDetaiAttributesInfo返回的细项不具备参考意义。

FaceHairAttributesInfo

人脸属性中的发型信息。

被如下接口引用：DetectFace。

名称	类型	描述
Length	Integer	0：光头，1：短发，2：中发，3：长发，4：绑发 注意：此字段可能返回 null，表示取不到有效值。
Bang	Integer	0：有刘海，1：无刘海 注意：此字段可能返回 null，表示取不到有效值。
Color	Integer	0：黑色，1：金色，2：棕色，3：灰白色 注意：此字段可能返回 null，表示取不到有效值。

FaceInfo

人脸信息列表。

被如下接口引用：DetectFace。

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	描述
X	Integer	人脸框左上角横坐标。 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 若需截取完整人脸，可以在完整分completess满足需求的情况下，将负值坐标取0。
Y	Integer	人脸框左上角纵坐标。 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 若需截取完整人脸，可以在完整分completess满足需求的情况下，将负值坐标取0。
Width	Integer	人脸框宽度。
Height	Integer	人脸框高度。
FaceAttributesInfo	FaceAttributesInfo	人脸属性信息，包含性别(gender)、年龄(age)、表情(expression)、魅力(beauty)、眼镜(glass)、口罩（ mask ）、头发（ hair ）和姿态 (pitch，roll，yaw)。只有当 NeedFaceAttributes 设为 1 时才返回有效信息。
FaceQualityInfo	FaceQualityInfo	人脸质量信息，包含质量分（ score ）、模糊分（ sharpness ）、光照分（ brightness ）、遮挡分（ completeness ）。只有当NeedFaceDetection设为1时才返回有效信息。 注意：此字段可能返回 null，表示取不到有效值。

FaceQualityCompleteness

五官遮挡分，评价眉毛（ Eyebrow ）、眼睛（ Eye ）、鼻子（ Nose ）、脸颊（ Cheek ）、嘴巴（ Mouth ）、下巴（ Chin ）的被遮挡程度。

被如下接口引用：DetectFace。

名称	类型	描述
Eyebrow	Integer	眉毛的遮挡分数[0,100]，分数越高遮挡越少。 参考范围：[0,80]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。
Eye	Integer	眼睛的遮挡分数[0,100],分数越高遮挡越少。 参考范围：[0,80]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。
Nose	Integer	鼻子的遮挡分数[0,100],分数越高遮挡越少。 参考范围：[0,60]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。
Cheek	Integer	脸颊的遮挡分数[0,100],分数越高遮挡越少。 参考范围：[0,70]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。
Mouth	Integer	嘴巴的遮挡分数[0,100],分数越高遮挡越少。 参考范围：[0,50]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。
Chin	Integer	下巴的遮挡分数[0,100],分数越高遮挡越少。 参考范围：[0,70]表示发生遮挡。 注意：此字段可能返回 null，表示取不到有效值。

FaceQualityInfo

版权所有：腾讯云计算（北京）有限责任公司

人脸质量信息，包含质量分（score）、模糊分（sharpness）、光照分（brightness）、遮挡分（completeness）。只有当NeedFaceDetection设为1时才返回有效信息。

被如下接口引用：DetectFace。

名称	类型	描述
Score	Integer	质量分: [0,100]，综合评价图像质量是否适合人脸识别，分数越高质量越好。 正常情况，只需要使用Score作为质量分总体的判断标准即可。Sharpness、Brightness、Completeness等细项分仅供参考。 参考范围：[0,40]较差，[40,60] 一般，[60,80]较好，[80,100]很好。 建议：人脸入库选取70以上的图片。 注意：此字段可能返回 null，表示取不到有效值。
Sharpness	Integer	清晰分：[0,100]，评价图片清晰程度，分数越高越清晰。 参考范围：[0,40]特别模糊，[40,60]模糊，[60,80]一般，[80,100]清晰。 建议：人脸入库选取80以上的图片。 注意：此字段可能返回 null，表示取不到有效值。
Brightness	Integer	光照分：[0,100]，评价图片光照程度，分数越高越亮。 参考范围： [0,30]偏暗，[30,70]光照正常，[70,100]偏亮。 建议：人脸入库选取[30,70]的图片。 注意：此字段可能返回 null，表示取不到有效值。
Completeness	FaceQualityCompleteness	五官遮挡分，评价眉毛（Eyebrow）、眼睛（Eye）、鼻子（Nose）、脸颊（Cheek）、嘴巴（Mouth）、下巴（Chin）的被遮挡程度。 注意：此字段可能返回 null，表示取不到有效值。

FaceRect

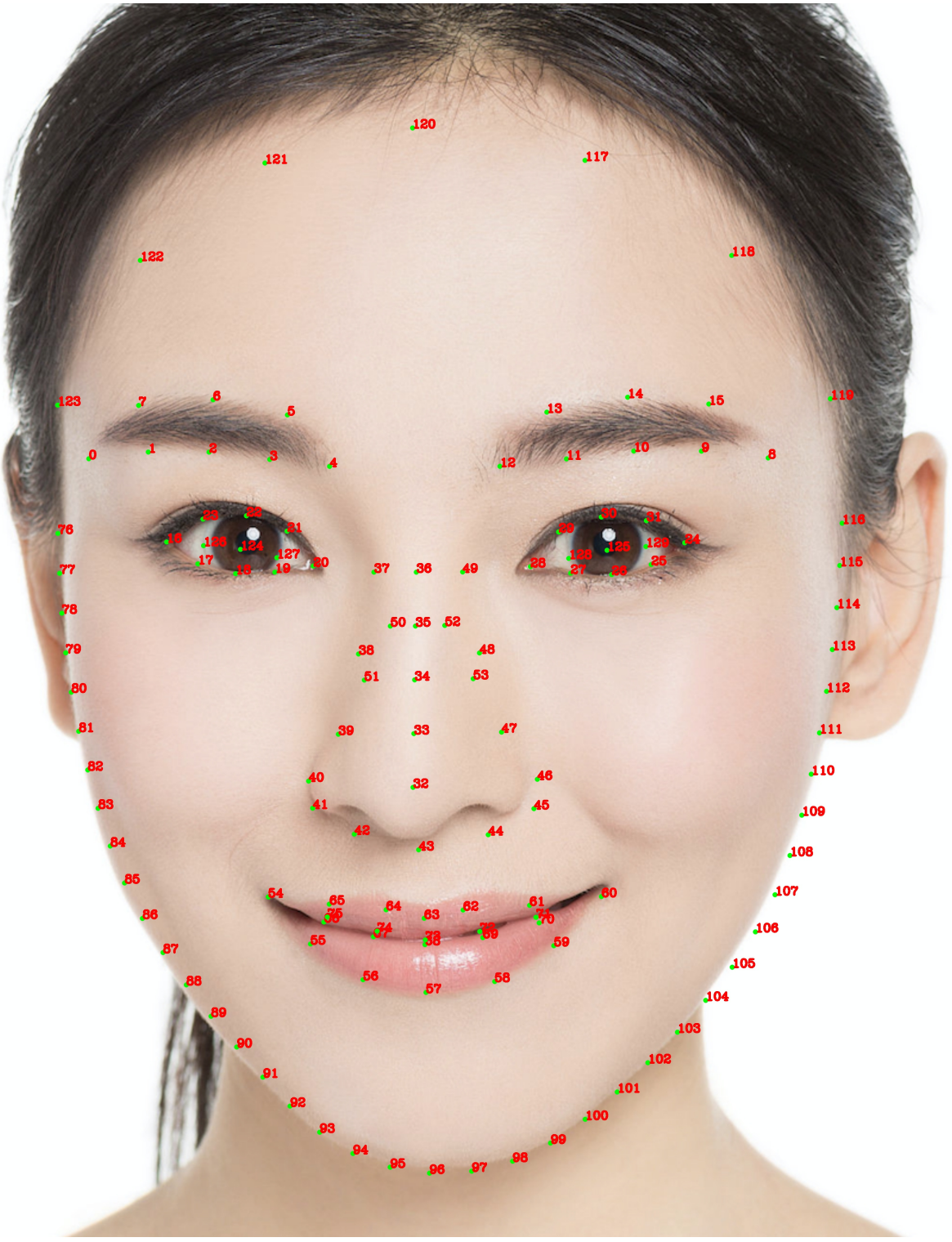
检测出的人脸框的位置

被如下接口引用：CreateFace, CreatePerson, DetectFaceAttributes, SearchFaces, SearchFacesReturnsByGroup, SearchPersons, SearchPersonsReturnsByGroup。

名称	类型	必选	描述
X	Integer	否	人脸框左上角横坐标。 - 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 - 若需截取完整人脸，可以在完整分completess满足需求的情况下，将负值坐标取0。 示例值：40
Y	Integer	否	人脸框左上角纵坐标。 - 人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。 - 若需截取完整人脸，可以在完整分completess满足需求的情况下，将负值坐标取0。 示例值：60
Width	Integer	否	人脸宽度。 示例值：200
Height	Integer	否	人脸高度。 示例值：150

FaceShape

五官定位（人脸关键点）具体信息。



被如下接口引用：AnalyzeFace。

名称	类型	描述
FaceProfile	Array of Point	描述脸型轮廓的 21 点。
LeftEye	Array of Point	描述左侧眼睛轮廓的 8 点。
RightEye	Array of Point	描述右侧眼睛轮廓的 8 点。
LeftEyeBrow	Array of Point	描述左侧眉毛轮廓的 8 点。

名称	类型	描述
RightEyeBrow	Array of Point	描述右侧眉毛轮廓的 8 点。
Mouth	Array of Point	描述嘴巴轮廓的 22 点。
Nose	Array of Point	描述鼻子轮廓的 13 点。
LeftPupil	Array of Point	左瞳孔轮廓的 1 个点。
RightPupil	Array of Point	右瞳孔轮廓的 1 个点。

GroupCandidate

分组识别结果Item

被如下接口引用：SearchFacesReturnsByGroup, SearchPersonsReturnsByGroup。

名称	类型	描述
GroupId	String	人员库ID 。
Candidates	Array of Candidate	识别出的最相似候选人。

GroupExDescriptionInfo

需要修改的人员库自定义描述字段key-value

被如下接口引用：ModifyGroup。

名称	类型	必选	描述
GroupExDescriptionIndex	Integer	是	人员库自定义描述字段Index，从0开始
GroupExDescription	String	是	需要更新的人员库自定义描述字段内容

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	人员库名称
GroupId	String	人员库ID
GroupExDescriptions	Array of String	人员库自定义描述字段 注意：此字段可能返回 null，表示取不到有效值。
Tag	String	人员库信息备注 注意：此字段可能返回 null，表示取不到有效值。
FaceModelVersion	String	人脸识别所用的算法模型版本。 注意：此字段可能返回 null，表示取不到有效值。
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) 00:00:00。有关更多信息，请

名称	类型	描述
		参阅 Unix 时间。 注意：此字段可能返回 null，表示取不到有效值。

Hair

头发信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Length	AttributeItem	头发长度信息。 AttributeItem对应的Type为 —— 0：光头，1：短发，2：中发，3：长发，4：绑发。
Bang	AttributeItem	刘海信息。 AttributeItem对应的Type为 —— 0：无刘海，1：有刘海。
Color	AttributeItem	头发颜色信息。 AttributeItem对应的Type为 —— 0：黑色，1：金色，2：棕色，3：灰白色。

Hat

帽子信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Style	AttributeItem	帽子佩戴状态信息。 AttributeItem对应的Type为 —— 0：不戴帽子，1：普通帽子，2：头盔，3：保安帽。
Color	AttributeItem	帽子颜色。 AttributeItem对应的Type为 —— 0：不戴帽子，1：红色系，2：黄色系，3：蓝色系，4：黑色系，5：灰白色系，6：混色系。

HeadPose

姿态信息

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
Pitch	Integer	上下偏移[-30,30]。
Yaw	Integer	左右偏移[-30,30]。
Roll	Integer	平面旋转[-180,180]。

Mouth

嘴巴信息。

被如下接口引用：DetectFaceAttributes。

名称	类型	描述
MouthOpen	AttributeItem	是否张嘴信息。 AttributeItem对应的Type为 —— 0：不张嘴，1：张嘴。

PersonExDescriptionInfo

需要修改的人员描述字段内容，key-value

被如下接口引用：CreatePerson, ModifyPersonGroupInfo。

名称	类型	必选	描述
PersonExDescriptionIndex	Integer	是	人员描述字段Index，从0开始
PersonExDescription	String	是	需要更新的人员描述字段内容

PersonGroupInfo

包含此人员的人员库及描述字段内容列表

被如下接口引用：GetPersonGroupInfo, SearchFaces, SearchFacesReturnsByGroup, SearchPersons, SearchPersonsReturnsByGroup。

名称	类型	描述
GroupId	String	包含此人员的人员库ID
PersonExDescriptions	Array of String	人员描述字段内容

PersonInfo

返回的人员信息

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
Gender	Integer	人员性别
PersonExDescriptions	Array of String	人员描述字段内容
FaceIds	Array of String	包含的人脸照片列表
CreationTimestamp	Integer	人员的创建时间和日期 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) 00:00:00。有关更多信息，请参阅 Unix 时间。

Point

坐标

被如下接口引用：AnalyzeDenseLandmarks, AnalyzeFace。

名称	类型	描述
X	Integer	x坐标。 示例值：78
Y	Integer	Y坐标。 示例值：38

Result

人脸的识别结果

被如下接口引用：SearchFaces, SearchPersons。

名称	类型	描述
Candidates	Array of Candidate	识别出的最相似候选人
FaceRect	FaceRect	检测出的人脸框位置
RetCode	Integer	检测出的人脸图片状态返回码。0 表示正常。 -1601代表不符合图片质量控制要求，此时Candidate内容为空。

ResultsReturnsByGroup

识别结果。

被如下接口引用：SearchFacesReturnsByGroup, SearchPersonsReturnsByGroup。

名称	类型	描述
FaceRect	FaceRect	检测出的人脸框位置。
GroupCandidates	Array of GroupCandidate	识别结果。
RetCode	Integer	检测出的人脸图片状态返回码。0 表示正常。 -1601代表不符合图片质量控制要求，此时Candidate内容为空。

UpgradeJobInfo

人员库升级任务信息

被如下接口引用：GetUpgradeGroupFaceModelVersionJobList。

名称	类型	描述
JobId	String	人员库升级任务ID，用于查询、获取升级的进度和结果。
GroupId	String	人员库ID。
FromFaceModelVersion	String	当前算法模型版本。
ToFaceModelVersion	String	目标算法模型版本。
StartTime	Integer	升级起始时间。 StartTime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) 00:00:00。 有关更多信息，请参阅 Unix 时间。

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	描述
Status	Integer	0表示升级中，1表示升级完毕，2表示回滚完毕。

版权所有：腾讯云计算（北京）有限责任公司

错误码

最近更新时间：2024-12-20 01:32:35

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

业务错误码

在接口调用过程中，成功调用量和返回 InvalidParameterValue.NoFaceInPhoto（图片不包含人脸）错误码的失败调用量，会计入计费。

错误码	说明
FailedOperation.AcrossVersionsError	该操作不支持跨算法模型版本。
FailedOperation.ConflictOperation	操作冲突，请勿同时操作相同的Person。
FailedOperation.CreateFaceConcurrent	增加人脸不支持并发操作。
FailedOperation.DuplicatedGroupDescription	同一人员库中自定义描述字段不可重复。
FailedOperation.FaceQualityNotQualified	人脸图片质量不符要求。
FailedOperation.FaceSizeTooSmall	人脸框大小小于MinFaceSize设置，人脸被过滤。

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明
FailedOperation.GroupCannotUpgrade	指定人员库的升级操作无法执行。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。
FailedOperation.GroupPersonMapExist	组中已包含对应的人员Id。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员Id。
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.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5M。
FailedOperation.JobCannnotRollback	任务无法回滚。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.RpcFail	Rpc调用失败。
FailedOperation.SearchFacesExceed	检索人脸个数超过限制。
FailedOperation.ServerError	算法服务异常，请重试。
FailedOperation.UnKnowError	内部错误。
FailedOperation.UpgradeJobIdNotExist	人员库升级任务不存在。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.DeleteFaceNumExceed	删除人脸数量超出限制。每个人员至少需要包含一张人脸。
InvalidParameterValue.FaceMatchThresholdIllegal	FaceMatchThreshold参数不合法。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.GroupExDescriptionsExceed	人员库自定义描述字段数组长度超过限制。最多可以创建5个。
InvalidParameterValue.GroupExDescriptionsNameIdentical	人员库自定义描述字段名称不可重复。
InvalidParameterValue.GroupExDescriptionsNameIllegal	人员库自定义描述字段名称包含非法字符。人员库自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupExDescriptionsNameTooLong	人员库自定义描述字段名称长度超出限制。
InvalidParameterValue.GroupFaceNumExceed	人员库人脸数量超出限制。
InvalidParameterValue.GroupIdAlreadyExist	人员库ID已经存在。人员库ID不可重复。
InvalidParameterValue.GroupIdIllegal	人员库ID包含非法字符。人员库ID只支持英文、数字、-%@#&_。
InvalidParameterValue.GroupIdNotExist	人员库ID不存在。

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明
InvalidParameterValue.GroupIdNotExists	对应的人员库ID在库中不存在。
InvalidParameterValue.GroupIdTooLong	人员库ID超出长度限制。
InvalidParameterValue.GroupIdsExceed	传入的人员库列表超过限制。
InvalidParameterValue.GroupNameAlreadyExist	人员库名称已经存在。人员库名称不可重复。
InvalidParameterValue.GroupNameIllegal	人员库名称包含非法字符。人员库名称只支持中英文、-、_、数字。
InvalidParameterValue.GroupNameTooLong	人员库名称超出长度限制。
InvalidParameterValue.GroupNumExceed	人员库数量超出限制。如需增加，请联系我们。
InvalidParameterValue.GroupNumPerPersonExceed	人员库数量超出限制。单个人员最多可被添加至100个人员库。
InvalidParameterValue.GroupTagIllegal	人员库备注包含非法字符。人员库备注只支持中英文、-、_、数字。
InvalidParameterValue.GroupTagTooLong	人员库备注超出长度限制。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.ImageEmptyError	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInGroups	指定分组中没有人脸。
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.PersonExDescriptionInfosExceed	人员自定义描述字段数组长度超过限制。最多5个。
InvalidParameterValue.PersonExDescriptionsNameIdentical	人员自定义描述字段名称不可重复。
InvalidParameterValue.PersonExDescriptionsNameIllegal	人员自定义描述字段名称包含非法字符。人员自定义描述字段名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonExDescriptionsNameTooLong	人员自定义描述字段名称长度超出限制。
InvalidParameterValue.PersonExistInGroup	组中已包含对应的人员Id。
InvalidParameterValue.PersonFaceNumExceed	人员人脸数量超出限制。单个人员最多可以包含五张人脸。
InvalidParameterValue.PersonGenderIllegal	人员性别设置出错。0代表未填写，1代表男性，2代表女性。
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。人员名称只支持中英文、-、_、数字。
InvalidParameterValue.PersonNameTooLong	人员名称超出长度限制。
InvalidParameterValue.QualityControlIllegal	QualityControl参数不合法。

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明
InvalidParameterValue.SearchPersonsExceed	搜索的人员数目超过限制。
InvalidParameterValue.UniquePersonControlIllegal	UniquePersonControl参数不合法。
InvalidParameterValue.UnsupportedGroupFaceModelVersion	该操作不支持算法模型版本2.0及以下版本。
InvalidParameterValue.UploadFaceNumExceed	一次最多上传四张人脸。
InvalidParameterValue.UrlIllegal	URL格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.ChargeStatusException	账号已欠费。
ResourceUnavailable.Delivering	资源正在发货中。
ResourceUnavailable.Freeze	账号已被冻结。
ResourceUnavailable.GetAuthInfoError	获取认证信息失败。
ResourceUnavailable.InArrears	账号已欠费。
ResourceUnavailable.LowBalance	余额不足。
ResourceUnavailable.NotExist	计费状态未知，请确认是否已在控制台开通服务。
ResourceUnavailable.NotReady	服务未开通。
ResourceUnavailable.Recover	资源已被回收。
ResourceUnavailable.StopUsing	账号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

版权所有：腾讯云计算（北京）有限责任公司

人脸识别 API 2017
API 概览

最近更新时间：2020-06-28 17:34:30

为满足广大开发者的开发需求，腾讯云人脸识别为您提供了丰富的人脸识别 API 接口。

注意:

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人脸识别1.0版本已从2019年6月21日起停止维护，如果您使用的仍是以下旧接口，请尽快升级成 2.0 版本 。

人脸识别接口

功能	详细说明
人脸检测	查看文档
五官定位	查看文档
人脸比对	查看文档
个体信息管理	查看文档
人脸验证	查看文档
人脸检索	查看文档
多脸检索	查看文档
人脸静态活体检测	查看文档

版权所有：腾讯云计算（北京）有限责任公司

错误码说明

最近更新时间：2019-04-26 17:19:37

人脸错误码说明

错误码	含义
3	错误的请求；其中 message:account abnormal,errorno is:2 为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 appid/bucket 与操作目标不匹配
9	签名过期
10	appid 不存在
11	secretid 不存在
12	appid 和 secretid 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	Bucket 不存在
21	无效参数
23	请求包体过大
107	鉴权服务内部错误
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误
-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空

版权所有：腾讯云计算（北京）有限责任公司

-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制
-1312	对个体添加了几乎相同的人脸
-1403	图片下载失败
-1400	非法的图片格式

版权所有：腾讯云计算（北京）有限责任公司

鉴权签名

最近更新时间：2020-06-29 15:05:50

拼接签名串

签名与鉴权

人脸支付通过签名来验证请求的合法性。开发者将签名授权给客户端，使其具备上传下载及管理指定资源的能力。
签名分为两种：

- 多次有效签名：签名中绑定或者不绑定文件 fileid，需要设置大于当前时间的有效期，最长可设置三个月，在此期间内签名可多次使用。
- 单次有效签名：签名中绑定文件 fileid，有效期必须设置为0，此签名只可使用一次，且只能应用于被绑定的文件。

具体应用参见 签名适用场景 。

签名算法

获取签名所需信息

生成签名所需信息必须使用主账号的，包括 AppID、SecretID 和 SecretKey。

> ⚠ 注意：
>
> - 如果您已使用过 API 密钥，或在2018年4月1日后接入人脸支付，请使用 API 密钥 。
> - 如果您已使用过100、101等开头的项目ID，可以继续使用 项目密钥，但建议使用 API 密钥，2018年4月1日后创建的项目ID，不再支持使用 项目密钥 。
> - 目前仅支持使用主账号的 Secret ID 和 Secret Key，暂不支持子账号的使用，计划后续实现。

拼接签名串

拼接多次有效签名串：
a=[appid]&b=[bucket]&k=[SecretID]&e=[expiredTime]&t=[currentTime]&r=[rand]&f=[fileid]
拼接单次有效签名串：
a=[appid]&b=[bucket]&k=[SecretID]&e=[expiredTime]&t=[currentTime]&r=[rand]&f=[fileid]

> ⚠ 注意：
>
> - 多次有效签名串中 fileid 为可选参数。
> - fileid 为空，表示不绑定资源，例如上传签名和下载签名。
> - fileid 不为空，表示绑定资源，例如绑定资源的下载。

签名串中各字段含义如下：

字段	解释
a	开发者的 AppID，接入人脸支付时由系统生成。
b	Bucket，图片资源的组织管理单元，历史遗留字段，可不填。
k	Secret ID。
e	签名的有效期，是一个符合 UNIX Epoch 时间戳规范的数值，单位为秒；单次签名时，e 必须设置为0。
t	当前时间戳，是一个符合 UNIX Epoch 时间戳规范的数值，单位为秒，多次签名时，e 应大于 t。
r	随机串，无符号10进制整数，用户需自行生成，最长10位。
f	资源存储的唯一标识，单次签名必填；多次签名选填，如填写则会验证与当前操作的文件路径是否一致。

版权所有：腾讯云计算（北京）有限责任公司

> **注意：**
> - 拼接单次有效签名串时，有效期e必须设置为0，以保证此签名只能针对固定资源使用一次。
> - 删除和复制文件必须使用单次有效签名，上传必须使用多次有效签名。
> - 具体应用参见 签名适用场景 。

生成签名

1. 使用 HMAC-SHA1 算法对请求进行加密（SHA1算法加密后的输出必须是原始的二进制数据，否则签名失败）。
2. 对 orignal 使用 HMAC-SHA1 算法进行签名，然后将 orignal 附加到签名结果的末尾，再进行 Base64 编码，得到最终的 sign。
3. 生成签名的公式如下：
   SignTmp = HMAC-SHA1(SecretKey, orignal)
   Sign = Base64(SignTmp.orignal)

> **注意：**
> - 此处使用的是标准的 Base64 编码，不是 urlsafe 的 Base64 编码。
> - SecretKey 为 API 密钥，orignal 为 拼接签名串 节中拼接好的签名串。

PHP 签名示例

本节介绍生成签名的算法实例，实例中使用 PHP 语言，如果开发者使用其他与开发，请使用对应的算法。

获取签名所需信息

获取得到的签名所需信息如下：
AppID：YOUR AppID_ID
Bucket：tencentyun（可不填）
Secret ID： YOUR SECRET_ID
Secret Key： YOUR SECRET_KEY

拼接签名串

```
$appid = "YOUR APPID_ID";
$bucket = "tencentyun";
$secret_id = "YOUR SECRET_ID";
$secret_key = "YOUR SECRET_KEY";
$expired = time() + 2592000;
$onceExpired = 0;
$current = time();
$rdm = rand();
$userid = "0";
$fileid = "tencentyunSignTest";

$srcStr =
'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$expired.'&t='.$current.'&r='.$rdm.'&f=';

$srcWithFile =
'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$expired.'&t='.$current.'&r='.$rdm.'&f='.$fileid;

$srcStrOnce= 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$onceExpired
.'&t='.$current.'&r='.$rdm
.'&f='.$fileid;
```

生成签名

SHA1 算法加密后的输出必须是原始的二进制数据，否则签名失败：

```
$signStr = base64_encode(hash_hmac('SHA1', $srcStr, $secret_key, true).$srcStr);

$srcWithFile = base64_encode(hash_hmac('SHA1', $srcWithFile , $secret_key, true).$srcWithFile );

$signStrOnce = base64_encode(hash_hmac('SHA1',$srcStrOnce,$secret_key, true).$srcStrOnce);

echo $signStr."\n";

echo $srcWithFile ."\n";

echo $signStrOnce."\n";
```

Java 签名示例

```
/*
 * Copyright 2017, Tencent Inc
 * All rights reserved.
 *
 * Created on 2017年9月12日
 */
package sign;

import java.util.Base64;
import java.util.Random;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public class Sign {

    /**
     * 生成 Authorization 签名字段
     *
     * @param appId
     * @param secretId
     * @param secretKey
     * @param bucketName
     * @param expired
     * @return
     * @throws Exception
     */
    public static String appSign(long appId, String secretId, String secretKey, String bucketName,
            long expired) throws Exception {
        long now = System.currentTimeMillis() / 1000;
        int rdm = Math.abs(new Random().nextInt());
        String plainText = String.format("a=%d&b=%s&k=%s&t=%d&e=%d&r=%d", appId, bucketName,
                secretId, now, now + expired, rdm);
        byte[] hmacDigest = HmacSha1(plainText, secretKey);
        byte[] signContent = new byte[hmacDigest.length + plainText.getBytes().length];
        System.arraycopy(hmacDigest, 0, signContent, 0, hmacDigest.length);
        System.arraycopy(plainText.getBytes(), 0, signContent, hmacDigest.length,
                plainText.getBytes().length);
        return Base64Encode(signContent);
    }
```

```
    /**
     * 生成 base64 编码
     *
     * @param binaryData
     * @return
     */
    public static String Base64Encode(byte[] binaryData) {
        String encodedstr = Base64.getEncoder().encodeToString(binaryData);
        return encodedstr;
    }

    /**
     * 生成 hmacsha1 签名
     *
     * @param binaryData
     * @param key
     * @return
     * @throws Exception
     */
    public static byte[] HmacSha1(byte[] binaryData, String key) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA1");
        SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "HmacSHA1");
        mac.init(secretKey);
        byte[] HmacSha1Digest = mac.doFinal(binaryData);
        return HmacSha1Digest;
    }

    /**
     * 生成 hmacsha1 签名
     *
     * @param plainText
     * @param key
     * @return
     * @throws Exception
     */
    public static byte[] HmacSha1(String plainText, String key) throws Exception {
        return HmacSha1(plainText.getBytes(), key);
    }

}
```

Node JS 签名示例

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

var secretId  = 'YOUR SECRET_ID',
    secretKey = 'YOUR SECRET_KEY',
    appid     = 'APPID',
    pexpired  = 86400,
    userid    = 0;

var now = parseInt(Date.now() / 1000),
    rdm = parseInt(Math.random() * Math.pow(2, 32)),
    plainText = 'a=' + appid + '&k=' + secretId + '&e=' + (now+pexpired) + '&t=' + now + '&r=' +
rdm + userid + '&f=',
```

```
      data = new Buffer(plainText,'utf8'),
    res = crypto.createHmac('sha1',secretKey).update(data).digest(),
    bin = Buffer.concat([res,data]);

var sign = bin.toString('base64');
```

C++ 签名示例

```
//g++ -g sign_sample.cpp -o sign -lcrypto

#include <stdio.h>
#include <stdlib.h>     /* srand, rand */
#include <time.h>       /* time */
#include <openssl/hmac.h>
#include <openssl/pem.h>
#include <openssl/bio.h>
#include <openssl/evp.h>
#include <string>
#include <vector>
#include <sstream>

#define HMAC_LENGTH 20

std::vector<unsigned char> hmac_sha1(std::string& data, std::string& key)
{
    unsigned char* result;
    unsigned int len = HMAC_LENGTH;

    result = (unsigned char*)malloc(sizeof(char) * len);

    HMAC_CTX ctx;
    HMAC_CTX_init(&ctx);

    HMAC_Init_ex(&ctx, key.c_str(), key.length(), EVP_sha1(), NULL);
    HMAC_Update(&ctx, (unsigned char*)data.c_str(), data.length());
    HMAC_Final(&ctx, result, &len);
    HMAC_CTX_cleanup(&ctx);

    std::vector<unsigned char> sha1;
    for (int i = 0; i < len; i++){
        sha1.push_back(result[i]);
    }

    free(result);
    return sha1;
}

std::string base64_encode(const std::string& src){
    BUF_MEM * bptr = NULL;
    BIO* b64 = BIO_new(BIO_f_base64());
    BIO_set_flags(b64, BIO_FLAGS_BASE64_NO_NL);
    BIO* bmem = BIO_new(BIO_s_mem());
    if(NULL == b64 || NULL == bmem){
        return "";
    }
```

```
    bmem = BIO_push(b64, bmem);
    int ret = BIO_write(bmem, src.data(), src.length());
    if(ret <= 0){
        return "";
    }

    ret = BIO_flush(bmem);
    BIO_get_mem_ptr(bmem, &bptr);
    std::string res(bptr->data, bptr->length);
    BIO_free_all(bmem);
    return res;

}


int main() {
    std::string appid = "1000001";
    std::string secret_id = "YOUR SECRETID";
    std::string secret_key = "YOUR SECRETKEY";
    time_t now = time(NULL);
    long expired = (long)now + 2592000;
    long onceExpired = 0;
    long current = (long)now;
    int rdm = rand();
    std::string userid = "0";

    std::stringstream raw_stream;
    raw_stream << "a=" << appid << "&k=" << secret_id << "&e=" << expired << "&t=" << current
               << "&r=" << rdm;
    std::string raw = raw_stream.str();

    std::vector<unsigned char> sha1 = hmac_sha1(raw, secret_key);

    std::stringstream data_stream;
    for (int i = 0; i != sha1.size(); i++)
        data_stream << sha1[i];
    data_stream << raw;
    std::string data = data_stream.str();

    std::string sign = base64_encode(data);

    printf("%s\n", sign.c_str());

    return 0;
}
```

签名适用场景

签名的适用场景有如下限制：

场景	适用签名
智能鉴黄	多次有效签名
图片标签	多次有效签名

OCR 识别	多次有效签名
人脸识别	多次有效签名
人脸融合	多次有效签名
人脸核身	多次有效签名

版权所有：腾讯云计算（北京）有限责任公司

人脸检测

最近更新时间：2020-06-29 15:07:29

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本 。

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/detect`
本接口（detect）用于检测给定图片中的所有人脸(Face)的位置和相应的面部属性，位置包括(x，y，w，h)，面部属性包括性别(gender)、年龄(age)、表情(expression)、魅力(beauty)、眼镜(glass)和姿态 (pitch，roll，yaw)。

注意：
本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名。
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）。
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url。 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名 。

注意：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

输入参数

使用 application/json 格式，参数选择 url；使用 multipart/form-data 格式，参数选择 image。

参数名	必选	类型	参数说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
mode	否	Int	检测模式，0：所有人脸，1：最大的人脸。
image	否	Binary	图片内容。
url	否	String	图片的 url、image提供一个即可；如果都提供，只使用 url

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符，可用于结果查询。
data.image_width	Int	请求图片的宽度。
data.image_height	Int	请求图片的高度。
data.face	Array(faceItem)	检测出的人脸信息列表。
code	Int	返回状态码。
message	String	返回错误消息。

FaceItem 说明：

字段	类型	说明
face_id	String	人脸标识
x	Int	人脸框左上角 x
y	Int	人脸框左上角 y
width	Float	人脸框宽度
height	Float	人脸框高度
gender	Int	性别 [0(female)~100(male)]
age	Int	年龄 [0~100]
expression	Int	微笑[0(normal)~50(smile)~100(laugh)]
glass	Bool	是否有眼镜 [true,false]
pitch	Int	上下偏移[-30,30]
yaw	Int	左右偏移[-30,30]
roll	Int	平面旋转[-180,180]
Beauty	Int	魅力[0~100]

注意：
如果图片中包含超过5张人脸，在返回的 FaceItem 中，只返回5张人脸完整信息，其他人脸只返回位置信息(face_id，x，y，width，height)，属性信息不返回。

示例

输入示例

使用 url 的请求包

```
POST /face/detect HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
```

```
  "appid":"123456",
  "mode":1,
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

使用 image 的请求包

```
POST /face/detect HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="mode";

1
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 313
Content-Type: application/json

{
  "data":{
    "session_id": "",
    "image_height": 200,
    "image_width": 200,
    "face": [{
      "face_id": "1005338790489817087",
      "x": 44,
      "y": 33,
      "height": 64.0,
      "width": 64.0,
      "pitch": 5,
      "roll": 0,
      "yaw": 6,
      "age": 37,
      "gender": 99,
      "glass": true,
      "expression": 6
      "beauty": 80
}]
  },
```

```
  "code":0,
  "message":"OK"
}
```

错误码

错误码	含义
3	错误的请求，其中 message:account abnormal,errorno is:2 为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 AppID /存储桶与操作目标不匹配
9	签名过期
10	AppID 不存在
11	SecretId 不存在
12	AppID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	存储桶不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误
-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误

-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制
-1312	对个体添加了几乎相同的人脸
-1313	参数不合法（特殊字符例如空格、斜线、tab、换行符）
-1400	非法的图片格式
-1403	图片下载失败

更多其他 API 错误码请查看 **错误码说明** 。

五官定位

最近更新时间：2019-06-21 11:46:19

> **注意：**
> - 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
> - 新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
> - 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本 。

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/shape`

本接口（shape）用于对请求图片进行五官定位，计算构成人脸轮廓的 88 个点，包括眉毛（左右各 8 点）、眼睛（左右各 8 点）、鼻子（ 13 点）、嘴巴（ 22 点）、脸型轮廓（ 21 点）。各部分的点的顺序如下图所示：



> **注意：**
> 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

输入参数

参数名	必选	类型	参数说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
mode	否	Int	检测模式，0：所有人脸，1：最大的人脸。
image	否	Binary	图片内容。
url	否	String	图片的 url，image 和 url 只需提供一个；如果都提供，只使用 url。

版权所有：腾讯云计算（北京）有限责任公司

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符，可用于结果查询
data.image_width	Int	请求图片的宽度
data.image_height	Int	请求图片的高度
data.face_shape.face_profile	Array(Point)	描述脸型轮廓的 21 点
data.face_shape.left_eye	Array(Point)	描述左测眼睛轮廓的 8 点
data.face_shape.right_eye	Array(Point)	描述右测眼睛轮廓的 8 点
data.face_shape.left_eyebrow	Array(Point)	描述左测眉毛轮廓的 8 点
data.face_shape.right_eyebrow	Array(Point)	描述右测眉毛轮廓的 8 点
data.face_shape.mouth	Array(Point)	描述嘴巴轮廓的 22 点
data.face_shape.nose	Array(Point)	描述鼻子轮廓的 13 点
code	Int	返回状态码
message	String	返回错误消息

Point 说明：

字段	类型	说明
x	Int	坐标 x，图片左上角为0。
y	Int	坐标 y，图片左上角为0。

示例

输入示例

使用 url

```
POST /face/shape HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "mode":1,
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

使用 image

```
POST /face/shape HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
```

```
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="mode";
1
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 313
Content-Type: application/json

{
  "data":{
    "session_id": "",
    "image_height": 200,
    "image_width": 200,
    "face_shape": [{
      "face_profile": [{"x":48,"y":55},{"x":49,"y":61},{"x":49,"y":66},{"x":50,"y":71},
{"x":51,"y":76},{"x":54,"y":81},{"x":56,"y":86},{"x":60,"y":90},{"x":65,"y":93},{"x":71,"y":95},
{"x":77,"y":96},{"x":82,"y":95},{"x":87,"y":93},{"x":92,"y":90},{"x":94,"y":85},{"x":97,"y":81},
{"x":99,"y":76},{"x":100,"y":71},{"x":101,"y":65},{"x":101,"y":60},{"x":101,"y":55}],
      "left_eye":[{"x":62,"y":55},{"x":64,"y":56},{"x":66,"y":57},{"x":68,"y":57},{"x":70,"y":56},
{"x":69,"y":54},{"x":66,"y":54},{"x":64,"y":54}],
      "right_eye":[{"x":92,"y":55},{"x":90,"y":56},{"x":88,"y":56},{"x":86,"y":56},{"x":84,"y":56},
{"x":85,"y":54},{"x":88,"y":53},{"x":90,"y":54}],
      "left_eyebrow":[{"x":56,"y":50},{"x":61,"y":50},{"x":65,"y":50},{"x":68,"y":50},
{"x":73,"y":50},{"x":69,"y":47},{"x":64,"y":46},{"x":60,"y":47}],
      "right_eyebrow":[{"x":97,"y":50},{"x":93,"y":50},{"x":89,"y":50},{"x":86,"y":50},
{"x":82,"y":50},{"x":85,"y":47},{"x":90,"y":46},{"x":95,"y":46}],
      "mouth":[{"x":67,"y":80},{"x":70,"y":83},{"x":74,"y":85},{"x":78,"y":85},{"x":81,"y":84},
{"x":83,"y":82},{"x":86,"y":80},{"x":84,"y":78},{"x":81,"y":77},{"x":78,"y":78},{"x":74,"y":78},
{"x":71,"y":79},{"x":71,"y":81},{"x":74,"y":81},{"x":78,"y":81},{"x":80,"y":80},{"x":83,"y":80},
{"x":83,"y":79},{"x":80,"y":79},{"x":78,"y":80},{"x":74,"y":80},{"x":71,"y":80}],
      "nose":[{"x":79,"y":68},{"x":77,"y":56},{"x":75,"y":60},{"x":74,"y":64},{"x":72,"y":67},
{"x":70,"y":70},{"x":74,"y":72},{"x":78,"y":72},{"x":81,"y":72},{"x":85,"y":70},{"x":83,"y":66},
{"x":81,"y":63},{"x":81,"y":59}]
}]
  },
  "code":0,
  "message":"OK"
}
```

错误码

API 错误码请查看 **错误码说明** 。

人脸比对

最近更新时间：2020-06-29 15:08:52

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本 。

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/compare`
本接口（compare）用于计算两个 Face 的相似性以及五官相似度。

说明：
本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

请求包 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名。
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）。
content-type	是	application/json 或 multipart/form-data	根据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url。 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法 。

注意：
选择 multipart/form-data 时，请使用 HTTP 框架/库推荐的方式设置请求的 content-type。如果直接调用 setHeader 等方法设置，可能会导致 boundary 缺失引起请求失败。

输入参数

使用 application/json 格式，参数选择 url；使用 multipart/form-data 格式，参数选择 image。

参数名	必选	类型	参数说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
imageA	否	Binary	A 图片内容。
imageB	否	Binary	B 图片内容。
urlA	否	String	A 图片的 url，image 和 url 只需提供一个；如果都提供，只使用 url。
urlB	否	String	B 图片的 url，image 和 url 只需提供一个；如果都提供，只使用 url。

输出参数

字段	类型	说明

data.session_id	String	相应请求的 session 标识符。
data.similarity	Float	两个 face 的相似度。
data.fail_flag	Int	标志失败图片，1 为第一张，2 为第二张（失败时返回）。
code	Int	返回码。
message	String	返回错误消息。

示例

输入示例

使用 url

```
POST /face/compare HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "urlA":"http://test-123456.image.myqcloud.com/testA.jpg",
  "urlB":"http://test-123456.image.myqcloud.com/testB.jpg"
}
```

使用 image

```
POST /face/compare HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="imageA"; filename="testA.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468
Content-Disposition: form-data; name="imageB"; filename="testB.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
```

```
Content-Length: 109
Content-Type: application/json

{
  "data":{
    "similarity":100.0,
    "session_id":""
  },
  "code":0,
  "message":"OK"
}
```

错误码

错误码	含义
3	错误的请求；其中 message:account abnormal,errorno is:2 为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 APPID /存储桶与操作目标不匹配
9	签名过期
10	APPID 不存在
11	SecretId 不存在
12	APPID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	存储桶不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用 新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误

-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制
-1312	对个体添加了几乎相同的人脸
-1313	参数不合法（特殊字符例如空格、斜线、tab、换行符）
-1400	非法的图片格式
-1403	图片下载失败

更多其他 API 错误码请查看 **错误码说明** 。

个体信息管理

最近更新时间：2021-05-26 10:10:15

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 人脸识别新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人员库管理（原个体信息管理）新老版本的人脸特征数据不互通，老版本已停止升级，建议您尽快迁移至新版本。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本接口 。

个体创建

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/newperson`

本接口（newperson）用于创建一个 person，并将 person 放置到 group_ids 指定的组当中，不存在的 group_id 会自动创建。

注意：

本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法

注意：

选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式，参数选择 url；使用 multipart/form-data 格式，参数选择 image。

参数名	必选	类型	参数说明
AppId	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看
group_ids	是	Array(String)	加入到组的列表
person_id	是	String	指定的个体 ID

image	否	binary	图片内容
url	否	String	图片的 url、image提供一个即可；如果都提供，只使用 url
person_name	否	String	名字
tag	否	String	备注信息

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符
data.suc_group	Int	成功被加入的 group 数量
data.suc_face	Int	成功加入的 face 数量
data.person_id	String	相应 person 的 ID
data.face_id	String	创建所用图片生成的 face_id
data.group_ids	Array(string)	加入成功的组 ID
code	Int	返回码
message	String	返回错误消息

示例

输入示例

使用 url 的输入示例：

```
POST /face/newperson HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 187
Content-Type: application/json

{
  "appid":"123456",
  "group_ids":[ "tencent", "qq" ],
  "person_id":"person0",
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

使用 image 的输入示例：

```
POST /face/newperson HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 735
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
```

```
----------------acebdf13572468
Content-Disposition: form-data; name="group_ids[0]";

tencent
----------------acebdf13572468
Content-Disposition: form-data; name="group_ids[1]";

qq
----------------acebdf13572468
Content-Disposition: form-data; name="person_id";

person0
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

返回示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 204
Content-Type: application/json

{
  "data":{
    "person_id":"person0",
    "suc_group":2,
    "suc_face":1,
    "session_id":"",
    "face_id":"1009550071676600319",
    "group_ids":["tencent", "qq"]
  },
  "code":0,
  "message":"OK"
}
```

删除个体

接口描述

接口请求示例： `https://recognition.image.myqcloud.com/face/delperson`
本接口（delperson）用于删除一个 Person。

> **⚠ 注意：**
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述

host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url 。 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法

注意：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	待删除个体 ID

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符
data.person_id	String	成功删除的 person_id
data.deleted	Int	成功删除的 person 数量
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

```
POST /face/delperson HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 66
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person0"
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 114
Content-Type: application/json

{
  "data":{
    "deleted": 1,
    "person_id": "person0",
    "session_id":""
  },
  "code":0,
  "message":"OK"
}
```

增加人脸

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/addface`
本接口（addface）用于将一组 face 加入到一个 person 中。一个 person 最多允许包含 20 个 face 。

> **注意：**
> - 加入相似度为[99%-100%]的人脸会返回错误。
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url。 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

> **注意：**
> 选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数
使用 application/json 格式，参数选择 url；使用multipart/form-data 格式，参数选择 image。

参数名	必选	类型	参数说明
AppId	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
person_id	是	String	待增加人脸的个体 ID
images	否	Array(Binary)	图片内容
urls	否	String	图片的 urls、image提供一个即可；如果都提供，只使用 urls
tag	否	String	备注信息

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符
data.added	Int	成功加入的 face 数量
data.face_ids	Array(String)	增加的人脸 ID 列表
data.ret_codes	array(Int)	每张图片增加人脸的返回码
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

使用 url 的输入示例：

```
POST /face/addface HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 191
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person0",
  "urls":[
    "http://test-123456.image.myqcloud.com/face1.jpg",
    "http://test-123456.image.myqcloud.com/face2.jpg"
  ]
}
```

使用 image 的输入示例：

```
POST /face/addface HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 641
Content-Type: multipart/form-data;boundary=----------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";
```

```
123456
----------------acebdf13572468
Content-Disposition: form-data; name="person_id";

person0
----------------acebdf13572468
Content-Disposition: form-data; name="images[0]"; filename="face1.jpg"
Content-Type: image/jpeg

yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy
----------------acebdf13572468
Content-Disposition: form-data; name="images[1]"; filename="face2.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 168
Content-Type: application/json

{
  "data":{
    "added":2,
    "session_id":"",
    "face_ids":["1009550071676600319","109387462576600319"],
    "ret_codes":[0, 0]
  },
  "code":0,
  "message":"OK"
}
```

删除人脸

接口描述

接口请求域名： `https://recognition.image.myqcloud.com/face/delface`
本接口（delface）用于删除一个 person 下的 face，包括特征、属性和 face id。

> **⚠ 注意：**
> - **本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。**
> - **如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。**

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名

content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

注意：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	待删除人脸的 person id
face_ids	是	Array(String)	删除人脸 ID 的列表

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符
data.deleted	Int	成功删除的 face 数量
data.face_ids	Array(String)	成功删除的 face_id 列表
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

```
POST /face/delface HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 125
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person0",
  "face_ids":[
    "199d1efd19ce4ee67a7ec7655f859b1a"
  ]
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 150
Content-Type: application/json

{
  "data":{
    "deleted": 1,
    "face_ids":[
      "199d1efd19ce4ee67a7ec7655f859b1a"
    ],
    "session_id":""
  },
  "code":0,
  "message":"OK"
}
```

设置信息

接口描述

接口请求域名： `https://recognition.image.myqcloud.com/face/setinfo`
本接口（setinfo）用于设置 person 的 name 。

> **注意：**
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

> **注意：**
> 选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	相应 person 的 ID
person_name	否	String	新的 name
tag	否	String	备注信息

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符
data.person_id	String	相应 person 的 ID
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

```
POST /face/setinfo HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 97
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person0",
  "person_name":"NicolasCage"
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 96
Content-Type: application/json

{
  "data":{
    "person_id": "person0",
    "session_id":""
  },
  "code":0,
  "message":"OK"
}
```

获取信息

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/getinfo`
本接口（getinfo）用于获取一个 person 的信息，包括 name，ID，tag，相关的 face，以及 groups 等信息。

> **注意：**
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

> **注意：**
> 选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	待查询个体的 ID

输出参数

字段	类型	说明
data.person_name	String	相应 person 的 name
data.person_id	String	相应 person 的 ID
data.face_ids	Array(String)	包含的人脸列表
data.group_ids	Array(String)	包含此个体的组 ID
data.session_id	String	相应请求的 session 标识符
data.tag	String	为此个体设置的备注信息
code	Int	返回状态码

版权所有：腾讯云计算（北京）有限责任公司

message	String	返回错误消息

示例

输入示例

```
POST /face/getinfo HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 66
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person0"
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 293
Content-Type: application/json

{
  "data":{
    "person_id": "person0",
    "person_name": "NicolasCage",
    "face_ids": [
      "199d1efd19ce4ee67a7ec7655f859b1a",
      "19321efd19ce4ee67a7127655f85934a"
    ],
    "group_ids":[
      "tencent"
    ],
    "tag":"",
    "session_id":""
  },
  "code":0,
  "message":"OK"
}
```

获取组列表

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/getgroupids`
本接口（getgroupids）用于获取一个 AppID 下所有 group 列表。

> ⚠ 注意：
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

注意：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID

输出参数

字段	类型	说明
data.group_ids	Array(String)	相应 AppId 的 group_id 列表
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

```
POST /face/getgroupids HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 41
Content-Type: application/json

{
  "appid":"123456"
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 136
```

```
Content-Type: application/json

{
  "data":{
    "group_ids": [
      "tencent",
      "tencent-beijing",
      "tencent-shanghai"
]
  },
  "code":0,
  "message":"OK"
}
```

获取人列表

接口描述

接口请求域名： `https://recognition.image.myqcloud.com/face/getpersonids`
本接口（getpersonids）用于获取一个组 group 中所有 person 列表。

> **注意：**
> - **本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。**
> - **如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。**

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

> **注意：**
> **选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。**

请求参数
使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID

group_id	是	String	组 ID

输出参数

字段	类型	说明
data.person_ids	Array(String)	相应 person 的 ID 列表
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

```
POST /face/getpersonids HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 65
Content-Type: application/json

{
  "appid":"123456",
  "group_id":"tencent"
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 120
Content-Type: application/json

{
  "data":{
    "person_ids": [
      "person0",
      "person1",
      "person2"
]
  },
  "code":0,
  "message":"OK"
}
```

获取人脸列表

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/getfaceids`
本接口（getfaceids）用于获取一个组 person 中所有 face 列表。

> **注意：**
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

- 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

注意：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	个体 ID

输出参数

字段	类型	说明
data.face_ids	Array(String)	相应 face 的 ID 列表
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

```
POST /face/getfaceids HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 66
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person1"
```

```
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 154
Content-Type: application/json

{
  "data":{
    "face_ids": [
      "1005338790489817087",
      "1005338790489817088",
      "1005338790489817089"
]
  },
  "code":0,
  "message":"OK"
}
```

获取人脸信息

接口描述

接口请求域名： `https://recognition.image.myqcloud.com/face/getfaceinfo`
本接口（getfaceinfo）用于获取一个 face 的相关特征信息。

> ⚠ **注意：**
> - **本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。**
> - **如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。**

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

> ⚠ **注意：**
> **选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。**

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
face_id	是	String	人脸 ID

输出参数

字段	类型	说明
data.face_info	faceItem	人脸信息
code	Int	返回状态码
message	String	返回错误消息

faceItem 说明

字段	类型	说明
face_id	String	人脸标识
x	Int	人脸框左上角 x
y	Int	人脸框左上角 y
width	Float	人脸框宽度
height	Float	人脸框高度
gender	Int	性别 [0(female)~100(male)]
age	Int	年龄 [0~100]
expression	Int	微笑[0(normal)~~50(smile)~~100(laugh)]
glass	Bool	是否有眼镜 [true,false]
pitch	Int	上下偏移[-30,30]
yaw	Int	左右偏移[-30,30]
roll	Int	平面旋转[-180,180]

示例

输入示例

```
POST /face/getfaceinfo HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 76
Content-Type: application/json

{
  "appid":"123456",
  "face_id":"1005338790489817087"
}
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 313
Content-Type: application/json

{
  "data":{
    "face_info": {
      "face_id": "1005338790489817087",
      "x": 44,
      "y": 33,
      "height": 64.0,
      "width": 64.0,
      "pitch": 5,
      "roll": 0,
      "yaw": 6,
      "age": 37,
      "gender": 99,
      "glass": true,
      "expression": 6
}
  },
  "code":0,
  "message":"OK"
}
```

Person 新增组信息

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/addgroupids`
本接口（addgroupids）用于为一个 person 新增 group_id。

> ⚠ **注意：**
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。

authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法

> **注意：**
> 选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	人脸 ID
group_ids	是	Array	group ID
session_id	否	String	会话 ID

输出参数

字段	类型	说明
data	groupsAddItem	结果信息
code	Int	返回状态码
message	String	返回错误消息

groupsAddItem 说明

字段	类型	说明
person_id	String	person ID
added	Int	成功新增组数
group_ids	Array	当前人所在组信息
session_id	String	会话 ID

Person 删除组信息

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/delgroupids`

本接口（delgroupids）用于为一个 person 删除 group_id。

> **注意：**
> - 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
> - 如果开发者使用的是原域名（service.image.myqcloud.com）且已产生调用，则无需更换域名。

输入参数

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	鉴权签名，详见 签名与鉴权

注意：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

请求参数

使用 application/json 格式

参数名	必选	类型	参数说明
AppId	是	String	项目 ID
person_id	是	String	人脸 ID
group_ids	是	Array	group ID
session_id	否	String	会话 ID

输出参数

字段	类型	说明
data	groupsadditem	结果信息
code	Int	返回状态码
message	String	返回错误消息

Groupsadditem 说明

字段	类型	说明
person_id	String	personId
deleted	Int	成功删除组数
group_ids	Array	当前人所在组信息
session_id	String	会话 ID

错误码

错误码	含义

3	错误的请求；其中 message:account abnormal,errorno is:2为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 AppID/存储桶与操作目标不匹配
9	签名过期
10	AppID 不存在
11	SecretId 不存在
12	APPID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	存储桶不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用 新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误
-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长

版权所有：腾讯云计算（北京）有限责任公司

-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制
-1312	对个体添加了相似度为99%及以上的人脸
-1313	参数不合法（特殊字符例如空格、斜线、tab、换行符）
-1400	非法的图片格式
-1403	图片下载失败

更多其他 API 错误码请查看 **错误码说明** 。

人脸验证

最近更新时间：2025-01-03 14:43:29

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 人脸识别新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人员库管理（原个体信息管理）新老版本的人脸特征数据不互通，老版本已停止升级，建议您尽快迁移至新版本。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本接口 。

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/verify`
本接口（verify）用于给定一个图片和一个 Person 时，检查是否是同一个人。

注意：
本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

基本概念

概念	解释
appid	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看 。
group_id	个体(person)以组（ group ）的形式存储，一个组可以包含多个个体，一个个体也可以存在于多个组。group_id 即用来标识 group。组(group)没有专门的创建接口，创建个体(person)时，指定 group_id 则会自动创建。
person_id	人脸以个体（ person ）的形式存储，一个个体下可以存储多张人脸。person_id 即用来标识 person。
face_id	标识每张人脸的 ID

说明：
如果开发者使用的是 V1版本，则 AppID 为其当时生成的 AppID。

group 信息管理接口如下：

接口	描述
创建个体	创建一个 person，并将 person 放置到 group_ids 指定的组当中，不存在的 group_id 会自动创建。
删除个体	删除一个 person。
增加人脸	将一组 face 加入到一个 person 中。一个 face 只能被加入到一个 person 中。一个 person 最多允许包含20个 face；加入几乎相同的人脸会返回错误。
删除人脸	删除一个 person 下的 face，包括特征，属性和 face_id 。
设置信息	设置 person 的 name 。
获取信息	获取一个 person 的信息，包括 name、ID、tag、相关的 face、以及 groups 等信息。
获取组列表	获取一个 AppID 下所有 group 列表。
获取人列表	获取一个 group 中所有 person 列表。

获取人脸列表	获取一个组 person 中所有 face 列表。
获取人脸信息	获取一个 face 的相关特征信息。

说明:

- 一个 AppID 下建立的 group_id 数量限制为5000个。
- 一个 group_id 下建立的 person_id 数量限制为20000个。
- 一个 person_id 下建立的人脸数量限制为20个。
- 每个请求的包体大小限制为1.5MB，不支持 .gif 类型的动图。

输入参数

请求头 header

使用 application/json 格式，参数选择 url；使用 multipart/form-data 格式，参数选择 image。

参数名	必选	类型	参数说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
person_id	是	String	待验证的 person。
image	否	Binary	图片内容。
url	否	String	图片的 url、image 和 url 只提供一个就可以了，如果都提供，只使用 url。

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符，可用于结果查询
data.confidence	Float	两个人的相似度
data.ismatch	Bool	两个输入是否为同一人的判断
code	Int	返回状态码
message	String	返回错误消息

示例

输入示例

使用 url

```
POST /face/verify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "person_id":"person1",
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
```

```
}
```

使用 image

```
POST /face/verify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="person_id";

Person1
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 409
Content-Type: application/json

{
  "data":{
    "session_id":"session_id",
"confidence":90.0,
"ismatch":true
  },
  "code":0,
  "message":"OK"
}
```

错误码

错误码	含义
3	错误的请求；其中 message:account abnormal,errorno is:2 为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 AppID /存储桶与操作目标不匹配
9	签名过期
10	AppID 不存在

11	SecretId 不存在
12	AppID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	存储桶不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用 新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误
-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制

版权所有：腾讯云计算（北京）有限责任公司

-1311	组个数超过限制
-1312	对个体添加了几乎相同的人脸
-1313	参数不合法（特殊字符例如空格、斜线、tab、换行符）
-1400	非法的图片格式
-1403	图片下载失败

更多其他 API 错误码请查看 **错误码说明** 。

人脸检索

最近更新时间：2020-06-29 15:08:11

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 人脸识别新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人员库管理（原个体信息管理）新老版本的人脸特征数据不互通，老版本已停止升级，建议您尽快迁移至新版本。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本接口 。

接口描述

接口请求域名： `https://recognition.image.myqcloud.com/face/identify` 本接口（identify）用于对一张待识别的人脸图片，在一个或多个 group 中识别出最相似的 Top5 person 作为其身份返回，返回的 Top5中按照相似度从大到小排列。

注意：

本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

基本概念

概念	解释
appid	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
group_id	个体(person)以组（ group ）的形式存储，一个组可以包含多个个体，一个个体也可以存在于多个组。group_id 即用来标识 group 。组(group)没有专门的创建接口，创建个体(person)时，指定 group_id 则会自动创建。
person_id	人脸以个体（ person ）的形式存储，一个个体下可以存储多张人脸。person_id 即用来标识 person。
face_id	标识每张人脸的 ID。

说明：

如果开发者使用的是 V1 版本，则 AppID 为其当时生成的 AppID。

group 信息管理接口如下：

接口	描述
创建个体	创建一个 person ，并将 person 放置到 group_ids 指定的组当中，不存在的 group_id 会自动创建。
删除个体	删除一个 person 。
增加人脸	将一组 face 加入到一个 person 中。注意，一个 face 只能被加入到一个 person 中。 一个 person 最多允许包含 20 个 face ； 加入几乎相同的人脸会返回错误。
删除人脸	删除一个 person 下的 face ，包括特征，属性和 face_id 。
设置信息	设置 person 的 name 。
获取信息	获取一个 person 的信息，包括 name、ID、tag、相关的 face、以及 groups 等信息。
获取组列表	获取一个 AppID 下所有 group 列表。
获取人列表	获取一个组 group 中所有 person 列表。

获取人脸列表	获取一个组 person 中所有 face 列表。
获取人脸信息	获取一个 face 的相关特征信息。

说明:
- 一个 AppID 下建立的 group_id 数量限制为5000个。
- 一个 group_id 下建立的 person_id 数量限制为20000个。
- 一个 person_id 下建立的人脸数量限制为20个。
- 每个请求的包体大小限制为1.5MB，不支持 .gif 类型的动图。

请求头 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名。
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）。
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url。 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法 。

注意:
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

输入参数

使用 application/json 格式时，参数可选择 url 和 image， 此时 image 须为经 base64编码后的 String，参数类型为 String。
使用 multipart/form-data 格式时， 参数可选择 url 和 image， 此时 image 须为 Binary 格式， 参数类型为 Binary。

参数名	必选	类型	参数说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
group_id	否	String	候选人组 ID，与 group_ids 二选一即可。
group_ids	否	Array(String)	候选人组 ID 列表，与 group_id 二选一即可。
image	否	Binary 或 String	图片二进制内容或经 base64转码后的字符串。
url	否	String	image 和 url 只需提供一个；如果都提供，只使用 url。

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符，可用于结果查询。

data.candidates	Array(IdentifyItem)	识别出的 Top5候选人。
code	Int	返回状态码。
message	String	返回错误消息。

IdentifyItem 说明：

字段	类型	说明
person_id	String	候选者 person_id
face_id	String	候选者 face_id
confidence	Float	候选者的置信度
tag	String	人脸备注信息

示例

输入示例

使用 url

```
POST /face/identify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "group_id":"tencent",
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

```
POST /face/identify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "group_ids":["tencent","qq"],
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

使用 image

```
POST /face/identify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
```

```
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="group_id";

tencent
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

```
POST /face/identify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="group_ids[0]";

tencent
----------------acebdf13572468
Content-Disposition: form-data; name="group_ids[1]";

qq
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 409
Content-Type: application/json

{
  "data":{
    "session_id":"session_id",
    "candidates":[
      {
        "person_id":"person3",
        "face_id":"1031567119985213439",
        "confidence":54.90695571899414,
        "tag":"new tag"
      },
```

```
      {
        "person_id":"person1",
        "face_id":"1031587105968553983",
        "confidence":54.86775207519531,
        "tag":"new tag"
      }
    ]
  },
  "code":0,
  "message":"OK"
}
```

错误码

错误码	含义
3	错误的请求；其中 message:account abnormal,errorno is:2为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 AppID /存储桶与操作目标不匹配
9	签名过期
10	AppID 不存在
11	SecretId 不存在
12	AppID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	存储桶不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用 新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误

-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制
-1312	对个体添加了几乎相同的人脸
-1313	参数不合法（特殊字符例如空格、斜线、tab、换行符）
-1400	非法的图片格式
-1403	图片下载失败

更多其他 API 错误码请查看 **错误码说明** 。

多脸检索

最近更新时间：2020-06-29 15:06:22

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 人脸识别新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人员库管理（原个体信息管理）新老版本的人脸特征数据不互通，老版本已停止升级，建议您尽快迁移至新版本。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本接口 。

接口描述

接口请求域名：`https://recognition.image.myqcloud.com/face/multidentify`

本接口（multidentify）用于对一张包含多个待识别的人脸的图片，在一个 group 或多个 groups 中识别出最相似的 person 作为其身份返回，返回的 Top5 中按照相似度从大到小排列。

说明：

本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。

基本概念

概念	解释
appid	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看。
group_id	个体（person）以组（group）的形式存储，一个组可以包含多个个体，一个个体也可以存在于多个组。group_id 即用来标识 group 。组（group）没有专门的创建接口，创建个体（person）时，指定 group_id 则会自动创建。
person_id	人脸以个体（person）的形式存储，一个个体下可以存储多张人脸，即用来标识 person。
face_id	标识每张人脸的 ID。

说明：

如果开发者使用的是 V1 版本，则 appid 为其当时生成的 AppID。

group 信息管理接口如下：

接口	描述
创建个体	创建一个 person ，并将 person 放置到 group_ids 指定的组当中，不存在的 group_id 会自动创建。
删除个体	删除一个 person 。
增加人脸	将一组 face 加入到一个 person 中。注意，一个 face 只能被加入到一个 Person 中。一个 person 最多允许包含 20 个 face ；加入几乎相同的人脸会返回错误。
删除人脸	删除一个 person 下的 face ，包括特征，属性和 face_id 。
设置信息	设置 person 的 name 。
获取信息	获取一个 person 的信息，包括 name、ID、tag、相关的 face、以及 groups 等信息。
获取组列表	获取一个 appid 下所有 group 列表。
获取人列表	获取一个组 group 中所有 person 列表。

获取人脸列表	获取一个组 person 中所有 face 列表。
获取人脸信息	获取一个 face 的相关特征信息。

说明：
- 一个 appid 下建立的 group_id 数量限制为 5000 个。
- 一个 group_id 下建立的 person_id 数量限制为 20000 个。
- 一个 person_id 下建立的人脸数量限制为 20 个。
- 每个请求的包体大小限制为 1.5MB ， 不支持 .gif 类型的动图。

请求包 header

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名
content-length	否	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url； 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法

说明：
选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setHeader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

输入参数

使用 application/json 格式，参数选择 url；使用multipart/form-data 格式，参数选择 image。

参数名	必选	类型	参数说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看
group_id	否	String	候选人组 ID、group_id 和 group_ids 只提供一个即可
group_ids	否	Array(String)	候选人组 ids
image	否	Binary	图片内容
url	否	String	图片的 url , image 和 url 只提供一个即可，如果都提供，只使用 url
session_id	否	String	session_id

输出参数

字段	类型	说明
data.session_id	String	相应请求的 session 标识符，可用于结果查询
data.results	Array (multiIdentifyItem)	识别结果

code	Int	返回状态码
message	String	返回错误消息

multiIdentifyItem 说明：

字段	类型	说明
candidates	identifyItem	候选者 识别结果
face_rect	facerectItem	候选者 人脸框位置

identifyitem 说明：

字段	类型	说明
person_id	String	候选者 person_id
face_id	String	候选者 face_id
confidence	Float	候选者的置信度
tag	String	人脸备注信息

facerectitem 说明：

字段	类型	说明
x	Int32	人脸位置左上角横坐标
y	Int32	人脸位置左上角纵坐标
width	Int32	人脸宽度
height	Int32	人脸高度

说明：
如果图片中包含超过 5 张人脸，在返回的 FaceItem 中，只返回 5 张人脸完整信息，其他人脸只返回位置信息(face_id，x，y，width，height)，属性信息不返回。

示例

输入示例

使用 url 和 group_id

```
POST /face/multidentify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "group_id":"tencent",
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

使用 url 和 group_ids

```
POST /face/multidentify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
  "appid":"123456",
  "group_ids":["tencent","qq"]
  "url":"http://test-123456.image.myqcloud.com/test.jpg"
}
```

使用 image 和 group_id

```
POST /face/multidentify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="group_id";

tencent
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

使用 image 和 group_ids

```
POST /face/multidentify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456
----------------acebdf13572468
Content-Disposition: form-data; name="group_ids[0]";

tencent
----------------acebdf13572468
Content-Disposition: form-data; name="group_ids[1]";

qq
```

```
----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 409
Content-Type: application/json

{
  "data":{
    "session_id": "111",
    "results": [{
      "candidates": [{
        "person_id": "person1",
        "face_id": "10001",
        "confidence": 64,
        "tag": ""
      }, {
        "person_id": "person2",
        "face_id": "100002",
        "confidence": 12,
        "tag": ""
      }],
      "face_rect": {
        "x": 60,
        "y": 80,
        "width": 50,
        "height": 60
      }
    }]
  },
  "code":0,
  "message":"OK"
}
```

错误码

错误码	含义
3	错误的请求；其中 message:account abnormal,errorno is:2 为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 AppID /存储桶与操作目标不匹配
9	签名过期
10	AppID 不存在
11	SecretId 不存在

版权所有：腾讯云计算（北京）有限责任公司

12	AppID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	存储桶不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用 新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误
-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url 图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制

版权所有：腾讯云计算（北京）有限责任公司

-1312	对个体添加了几乎相同的人脸
-1313	参数不合法（特殊字符例如空格、斜线、tab、换行符）
-1400	非法的图片格式
-1403	图片下载失败

更多其他 API 错误码请查看 **错误码说明** 。

人脸静态活体检测

最近更新时间：2019-06-21 11:45:43

接口描述

注意：

- 人脸识别全面升级接口，算法更强、性能更优，欢迎立即体验 人脸识别 。
- 新老版本的接口不兼容，控制台的数据统计不互通，但计费模式相同，且共享计费阶梯。
- 人脸识别1.0版本接口已从2019年6月21日起停止维护，如果您使用的仍是当前的旧接口，请尽快升级成 2.0版本 。

接口请求域名： `https://recognition.image.myqcloud.com/face/livedetectpicture`
本接口（livedetectpicture）用于对用户上传的静态照片进行人脸活体检测。与动态活体检测的区别是：静态活体检测中，用户不需要通过唇语或摇头眨眼等动作来识别。
静态活体检测仅用于对防攻击要求不高的场景，如果对活体检测有更高安全性要求，请使用 活体人脸比对 。

注意：

- 本接口支持 HTTPS 协议，如果您现在使用的是 HTTP 协议，为了保障您的数据安全，请切换至 HTTPS。
- 如果开发者使用的是原域名（service.image.myqcloud.com），为获得更好的体验，请及时切换到以上域名。

请求头 header

所有请求都要求含有下表列出的头部信息：

参数名	必选	值	描述
host	是	recognition.image.myqcloud.com	腾讯云人脸识别服务器域名。
content-length	否	包体总长度	图片大小建议小于1280 * 720px；每个请求的包体大小限制为6MB；不支持 .gif 类型的动图。
content-type	是	application/json 或 multipart/form-data	据不同接口选择： 1. 使用 application/json 格式，参数为 url，其值为图片的 url。 2. 使用 multipart/form-data 格式，参数为 image，其值为图片的二进制内容。
authorization	是	鉴权签名	多次有效签名，用于鉴权，生成方式见 鉴权签名方法 。

注意：

如选择 multipart/form-data，请使用 HTTP 框架/库推荐的方式设置请求的 content-type，不推荐直接调用 setheader 等方法设置，否则可能导致 boundary 缺失引起请求失败。

输入参数

使用 application/json 格式，参数选择 url；使用 multipart/form-data 格式，参数选择 image。

参数名	必选	类型	说明
appid	是	String	接入项目的唯一标识，可在 账号信息 或 云 API 密钥 中查看

image	否	Binary	图片内容（图片的宽高比请接近3：4，不符合宽高比的图片返回的分值不具备参考意义）
url	否	String	图片的 image 和 url 只提供一个即可，如果都提供，只使用 url（图片的宽高比请接近3：4，不符合宽高比的图片返回的分值不具备参考意义）

> **注意：**
> 图片的宽高比请接近3：4，不符合宽高比的图片返回的分值不具备参考意义。

输出参数

字段	类型	说明
data.score	Int	活体打分，取值范围 [0,100]，分数一般落于[80, 100]区间内，0分也为常见值。推荐值大于 87 时可判断为活体。可根据具体场景自行调整阈值。
code	Int	错误码。
message	String	错误描述。

示例

输入示例

使用 url

```
POST /face/livedetectpicture HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 117
Content-Type: "application/json"

{"appid":"1000001","url":"http://test-123456.image.myqcloud.com/test.jpg"}
```

使用 image

```
POST /face/livedetectpicture HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: recognition.image.myqcloud.com
Content-Length: 641
Content-Type: multipart/form-data;boundary=--------------acebdf13572468

----------------acebdf13572468
Content-Disposition: form-data; name="appid";

123456----------------acebdf13572468
Content-Disposition: form-data; name="image"; filename="face.jpg"
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
----------------acebdf13572468--
```

输出示例

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 168
Content-Type: application/json
{
"data":{
"score":30,
},
"code":0,
"message":"OK"
}
```

错误码

错误码	含义
3	错误的请求；其中 message:account abnormal,errorno is:2 为账号欠费停服
4	签名为空
5	签名串错误
6	签名中的 AppID/Bucket 与操作目标不匹配
9	签名过期
10	AppID 不存在
11	SecretId 不存在
12	AppID 和 SecretId 不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	Bucket 不存在
21	无效参数
23	请求包体过大
24	本接口即将下线，请使用 新版人脸识别
107	鉴权服务不可用
108	鉴权服务不可用
213	内部错误
-4006	视频中自拍照特征提取失败
-4007	视频中自拍照之间对比失败
-4009	Card 照片提取特征失败
-4010	自拍照与Card照片相似度计算失败
-4015	自拍照人脸检测失败

-4016	自拍照解码失败
-4017	Card 照片人脸检测失败
-4018	Card 照片解码失败
-5001	视频无效
-5002	唇语失败
-5005	自拍照解析照片不足
-5007	视频没有声音
-5008	声音识别失败
-5009	视频人脸检测失败，没有嘴或者脸
-5010	唇动失败
-5011	活体检测失败(活体其他错误都归类到里面)
-5012	视频中噪声太大
-5013	视频里的声音太小
-5014	活体检测 level 参数无效
-5015	视频像素太低，最小320 * 480
-5801	请求缺少身份证号码或身份证姓名
-5802	服务器内部错误，服务暂时不可用
-5803	身份证姓名与身份证号码不一致
-5804	身份证号码无效
-5805	用户未输入图像或者 url 下载失败
-5806	身份证号码或者身份证姓名格式错误
-5807	查询身份证信息错误

更多其他 API 错误码请查看 **错误码说明** 。