

SenseTime Cloud API Platform

Development Guide

开发手册

北京市商汤科技开发有限公司
BEIJING SENSETIME CO., LTD.

简介

本手册为 SenseTime HTTP API 开发者文档，为开发者提供了详细的人脸技术 API 接口说明及使用示例。

本手册内容若有变动，恕不另行通知。本手册可能涉及北京市商汤科技开发有限公司的专利、商标、版权或其他知识产权，未得到北京市商汤科技开发有限公司明确的书面许可，不得以任何目的复制或传播手册中的内容。

如您在实际使用过程中有任何问题及建议，欢迎将您遇到的问题描述或者对我们的建议，附上您的详细信息发送到邮箱：cloud-api-support@sensetime.com。

版权所有© 北京市商汤科技开发有限公司

Copyright © BEIJING SENSETIME CO., LTD.

目录

简介.....	2
目录.....	3
功能介绍.....	5
人脸检测.....	5
关键点检测.....	6
属性提取.....	7
人脸验证.....	8
人脸搜索.....	9
人脸识别.....	10
API 入门.....	11
API 调用.....	12
HTTP 工具.....	12
权限认证.....	12
调用限制.....	13
参数.....	13
异步调用.....	13
API 返回处理.....	14
正常结果.....	14
错误处理.....	14
API 参考.....	18
信息查询.....	19
/info/api.....	19
/info/task.....	22
/info/image.....	25
/info/face.....	30
/info/list_persons.....	35
/info/list_groups.....	38
/info/list_facesets.....	41
/info/person.....	44
/info/group.....	46
/info/faceset.....	49
人脸检测与分析.....	51
/face/detection.....	51
/face/verification.....	57
/face/search.....	61
/face/identification.....	65
/face/training.....	70
人的管理.....	73
/person/create.....	73
/person/delete.....	76
/person/add_face.....	78

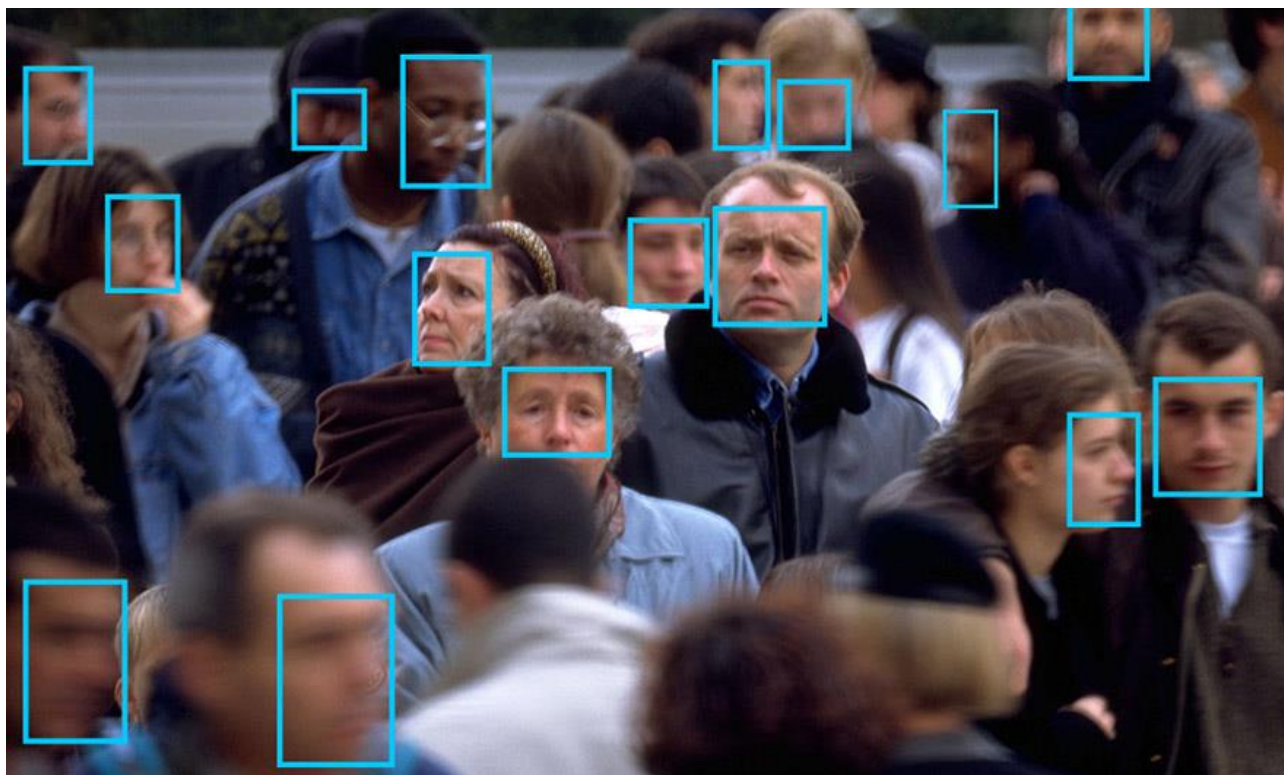
/person/remove_face	81
/person/change	84
组的管理	86
/group/create	86
/group/delete.....	89
/group/add_person	91
/group/remove_person.....	94
/group/change	97
人脸集合管理.....	99
/faceset/create.....	99
/faceset/delete.....	102
/faceset/add_face	104
/faceset/remove_face.....	107
/faceset/change.....	110
附录.....	112
21 个关键点	113
106 个关键点	114

功能介绍

SenseTime HTTP API 提供世界先进的人脸识别技术，目前的主要功能有：人脸检测、关键点检测、属性提取、人脸验证、人脸搜索、及人脸识别等，更多功能添加中。

人脸检测

人脸检测（Face Detection）能快速准确地检测出人脸在图片中的位置。SenseTime 的人脸检测技术对光线变化，角度变化，不同肤色，不同年龄的人脸都有较好的适应性。我们基于深度学习的人脸检测算法，一次性克服了人脸检测中的几项难题：侧脸、半遮挡、模糊人脸，极大提升了各种现实场景中的人脸检出效果。



关键点检测

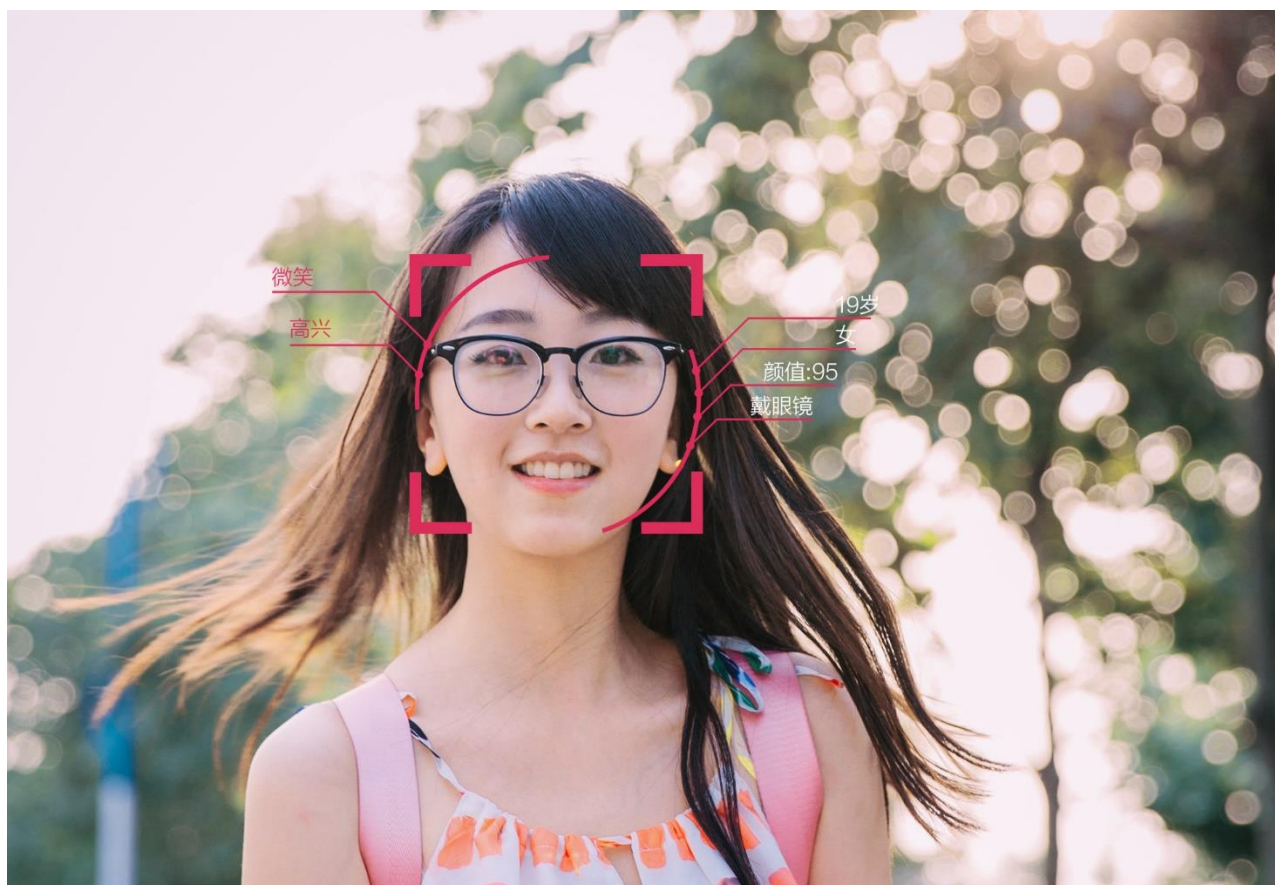
人脸可以由多个特征点序列表达（眼角，眉毛尖端，嘴角，鼻尖等），每个特征点可以以图像坐标的形式提取，特征点数量的多少（21 点，106 点）可以表达不同精度的人脸变化。

关键点检测（Face Alignment）能精确定位人脸上各关键部位。SenseTime 的人脸关键点定位算法可对各种表情、姿态、角度丰富多变的人脸进行精准关键点定位。SenseTime 的人脸关键点算法仍能进行精准定位。



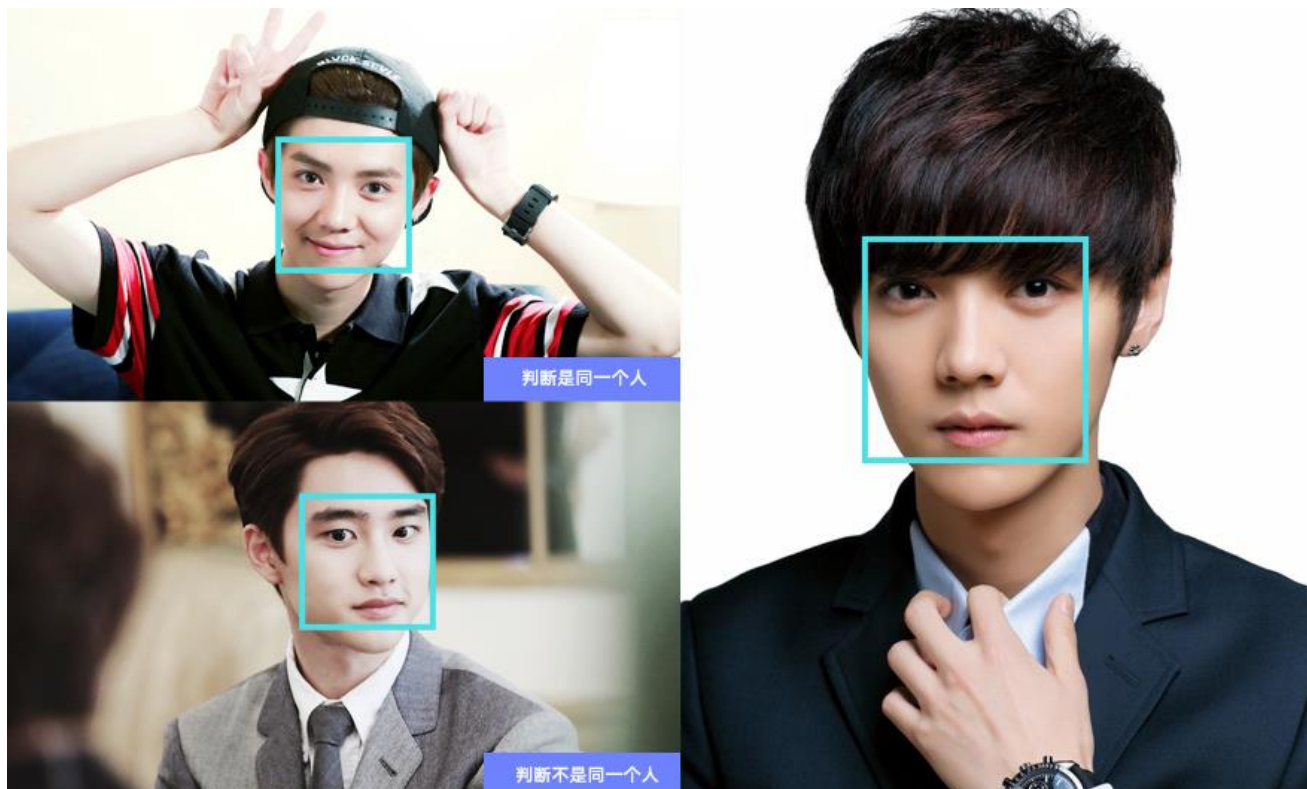
属性提取

属性提取（Face Attributes）能从人脸上提取出年龄、性别、戴眼镜、颜值等各种属性，我们正在添加更多属性，并持续不断提高准确度。



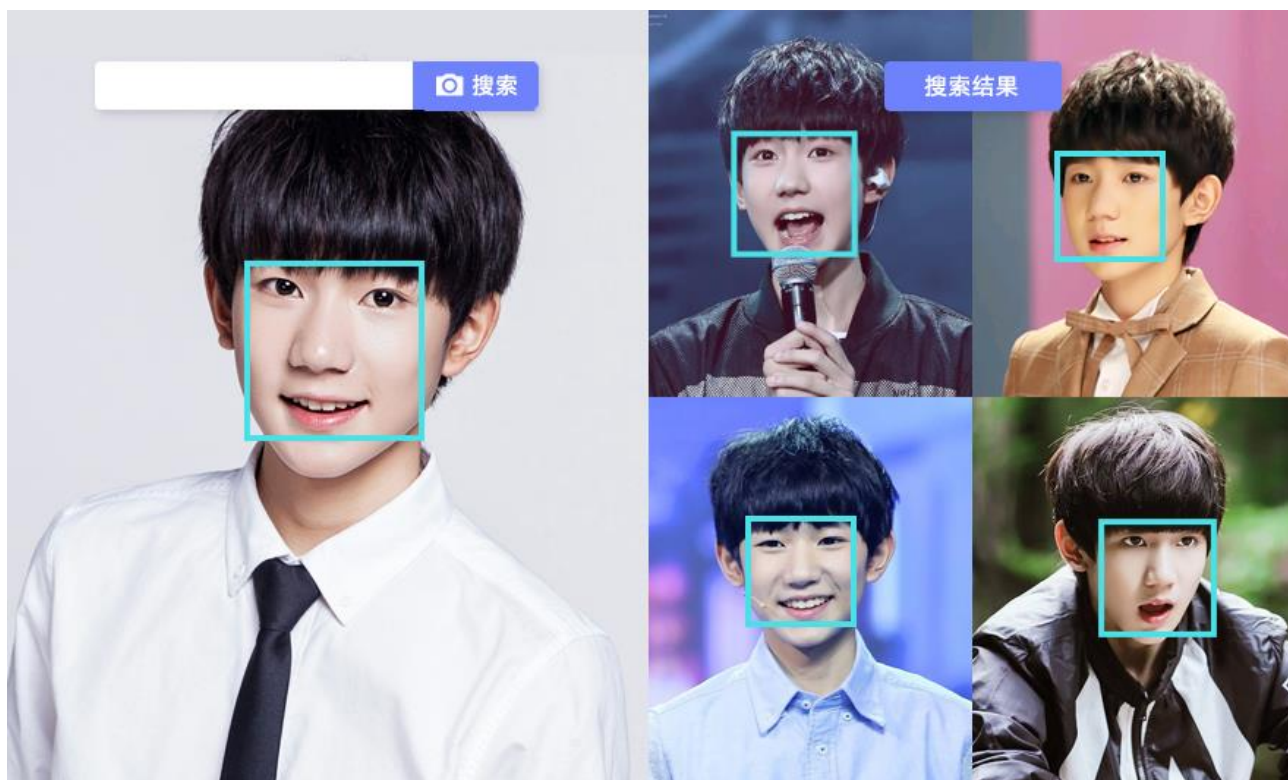
人脸验证

人脸验证（Face Verification）可以计算两张人脸的相似度，从而判断是否为同一个人。SenseTime 人脸验证技术采用深度学习方法对大量数据进行特征提取，利用特征匹配程度来判断相似或不同的人脸。



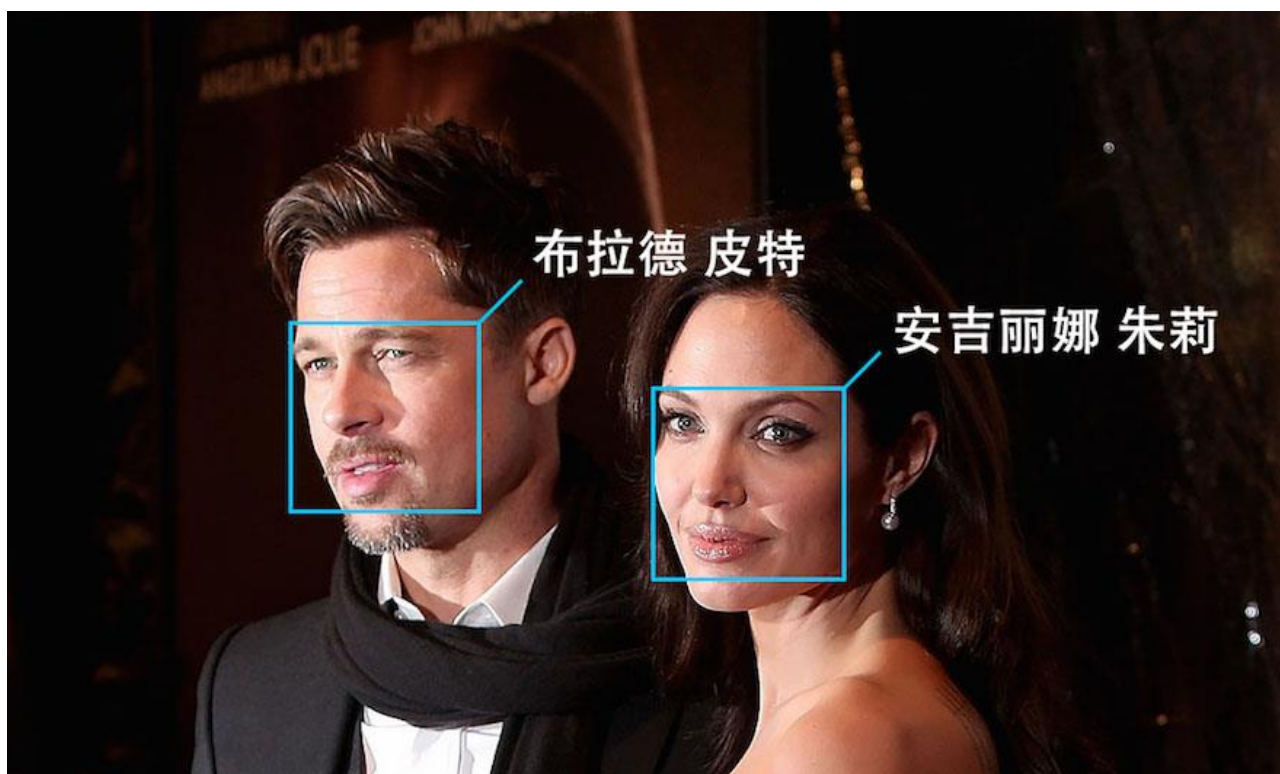
人脸搜索

人脸搜索（Face Search）用于在多个人脸中搜索出与目标人脸最相似的脸。SenseTime 的人脸搜索算法通过 1:N 的人脸相似度比对可在大量的数据中准确搜索出与目标人脸相似的人脸。



人脸识别

人脸识别（Face Identification）能够快速且准确的识别出该目标人脸所属的人。该技术能够广泛运用于社交网络、人脸解锁等应用场景中。



API 入门

本章从整体上描述 SenseTime HTTP API 的使用方法，包括 API 的调用方法、返回错误处理方法，以讲解的形式让读者学会基本的使用。而每一个 API 的详尽文档请参见下一章：[API 参考](#)。

API 调用

SenseTime API 为 HTTP API，目前的版本并未完全按照 RESTful 风格设计。

调用方法：使用者（客户端）向我们的服务器发起 HTTP 请求（request），并附上合适的参数；服务器收到请求后进行相关计算，然后将结果返回（response）；客户端从 HTTP 返回中解析出需要的结果。

HTTP 工具

常见的编程语言都有 HTTP 库（自带或第三方），可以发起 HTTP 请求，即为可以调用 SenseTime API。

在编程之前，想先测试一下 API，有些读者会选择使用浏览器。浏览器地址栏只支持 HTTP GET 请求，而我们的许多 API 使用 POST 请求。所以要使用浏览器的话，请为浏览器安装支持其他 HTTP 方法的插件（通常叫 REST Client 之类的名字）。不编程、也不使用浏览器，也可以安装其他独立的图形或命令行 HTTP 工具。推荐两个命令行工具：curl (<http://curl.haxx.se/>) 和 httpie (<http://httpie.org/>)，命令格式请参阅各自文档。

权限认证

SenseTime API v1 版本的用户会获得一个账户（包含 api_id 和 api_secret），所有的 API 调用，都必须在参数中包含 api_id 和 api_secret 两个参数用于授权。参数使用方法见下一小节。刚获得帐户之后，建议先调用一下 [/info/api](#)，一方面测试调用方法是否正确，另一方面也可以验证自己的 api_id 和 api_secret 是否有误（下方的 ID 与 SECRET 请替换为自己的账户信息）：
curl 样例：

```
curl "https://v1-api.visioncloudapi.com/info/api?api_id=ID&api_secret=SECRET"
```

httpie 样例：

```
http GET "https://v1-api.visioncloudapi.com/info/api?api_id=ID&api_secret=SECRET"
```

返回结果 HTTP 状态码为 200 即为正常。

api_secret 相当于专属于某个用户的密码，请务必妥善保管，泄露后果自负。请经常关注调用统计，如怀疑 api_secret 泄露，请及时重置。

SenseTime API 全部使用加密的 https 链接，用户与我们服务器之间的通信数据完全加密。请不要关闭客户端的 SSL 证书验证。

调用限制

为防止滥用，本服务对用户调用 API 的频率有所限制，参见下一小节 [API 返回处理](#) 中的 [超过调用限制](#) 部分。

参数

SenseTime 每个 API 在调用时都需要附上特定的参数，以提供算法所需信息。每个 API 的参数详细定义参见 [API 参考](#) 一章，本小节说明的是调用 API 时附带参数的方法。SenseTime API 支持多种参数传递方式，除了特殊的情况下必须选择某种方式外（后面说明），用户可自行选择：

1. 以 Query String 的形式附加在 URL 之后。例：本章中“权限认证”小节。这种方式最简单，推荐使用。
2. POST 请求的 API，在请求体（body）中使用 JSON 格式来编码参数。
3. POST 请求的 API，在请求体中使用 application/x-www-form-urlencoded 格式来编码参数。
4. POST 请求的 API，在请求体中使用 multipart/form-data 格式来编码参数。

安全事项：使用第 1 种方式提供参数，如果在浏览器地址栏中进行测试，通常会被记录在浏览器历史中，有泄露账户的风险。

例外：当调用 API 上传文件时（例如 [/face/detection](#) 中的 file 参数），必须使用第 4 种方式提供参数，并且需要提供的是文件内容，而不是文件名。

其他编程语言的参数编码方式，请查阅 HTTP 协议相关资料和所用语言 HTTP 库的文档。

异步调用

为了避免阻塞，提高调用效率，SenseTime API 提供了异步调用方案。当发出的 HTTP 请求计算耗时较长时，服务器返回的 HTTP 状态码为 202，并会返回当前任务的相关信息。用户可以继续运行程序而无需等待所调用的 API 的计算结果返回，过段时间后，再根据获得的 task_id（有效期为 24 小时）调用 [/info/task](#) API 获取相应请求的计算结果。

目前支持异步调用的 API 有：

[/face/verification](#)、[/face/identification](#)、[/face/search](#)、[/face/training](#)。

API 返回处理

HTTP 请求的返回结果（response）分为：状态码（status code）、头部（headers）、消息体（body）。SenseTime API 的返回结果也是如此，这三部分都包含有重要信息，其中算法的结果会以 JSON 格式放在消息体中。

如何从 HTTP 返回中分别获取这三部分信息，请参见所用 HTTP 库的文档。解析 JSON 格式你需要寻找所用语言的 JSON 库，参见 <http://www.json.org/> 页面下方。

正常结果

状态码为 2xx（2 开头）的为正常返回，最常见的 200，表示 OK。此时解析消息体中的 JSON 数据就可以获得本次 API 调用的计算结果了。JSON 中也会有一个 status 字段值为 OK：

```
{
  "status": "OK"
  .....
}
```

错误处理

当状态码为 4xx 或 5xx 时，则意味着发生了错误，不同的状态码表示不同的错误类型。消息体中的 JSON 数据包含了详尽的错误信息：status 字段中包含了错误类型描述，如果还有更多错误信息则会在其他字段中提供。

收到 HTTP 返回后，一般先判断一下状态码，若属于错误则按需进行错误处理；部分 HTTP 库遇到属于错误的状态码后会抛出异常（Exception），这种情况下，直接处理异常即可。

HTTP 库遇错抛异常的情况，从异常对象中同样可以获得返回的状态码、头部和消息体，具体用法请查阅所用库的文档。

常见错误

每个 API 的调用都可能发生各种错误，其中一些公共的错误在这里列出，而 API 特定的错误请查看 [API 参考](#)。

- 400 请求错误（Bad Request）：最常见的错误，通常是提供的参数（数量、类型）不符合要求，或者操作内容无效。前一种情况在 [参数错误](#) 中详细说明。

- **401 未授权 (Unauthorized)**: 上一章 [权限认证](#) 小节中提到每一个 API 调用时必须带上 `api_id` 和 `api_secret` 参数。出现本错误则为缺少这两个参数或它们的值有误, 或者账号过期。
- **403 禁止访问 (Forbidden)**: 可能是本用户没有权限调用该 API, 或者超过了调用限额。后一种情况在[超过调用限制](#)中详细说明。
- **404 未找到 (Not Found)**: 没有找到 API, 通常是 URL 写错了, 请检查。
- **413 请求体过大 (Request Entity Too Large)**: 通常是上传的文件过大导致, 请参阅所用 API 中的文件大小限制说明。
- **5xx 服务器错误 (Server Error)**: 其中 **500** 错误最常见, 通常是我们的服务还不完善导致。遇到这种情况, 请记下出错调用的 API、参数、时间等信息, 向我们的技术支持反馈。

404, 413, 5xx 这些类别的错误, HTTP 返回的消息体可能为空或者不是 JSON 格式, 请在解析 JSON 时注意处理解析错误。

参数错误

[API 参考](#) 中对每个 API 接受的参数都做了详细的定义, 包括参数名、值类型、值范围、是否必需等。调用 API 时提供的参数不满足这些要求, 就会出现无效参数错误: HTTP 状态码为 400, 消息体 JSON 中 `status` 字段值为 `INVALID_ARGUMENT`, `reason` 字段给出更具体原因。

样例 1, 缺少参数 (省略部分头部, 后同):

```
HTTP/1.1 400 Bad Request
Content-Length: 66
Content-Type: application/json

{
  "status": "INVALID_ARGUMENT",
  "reason": "must specify 'file' or 'url' argument"
}
```

样例 2, 参数值不是可选值之一 (参数 `top` 的值只能在 1 到 10 中选取):

```
HTTP/1.1 400 Bad Request
Content-Length: 86
Content-Type: application/json

{
  "status": "INVALID_ARGUMENT",
  "reason": "argument 'top': not in range(1..10)"
}
```

超过调用限制

目前 SenseTime API 对调用频率和人（person）、组（group）、人脸集合（faceset）的相关数量都有限制，每个用户的限制不同，可以通过 </info/api> 查看相关信息：

样例输出：

```
{
  "status": "OK",
  "rate_limit": "50q/10s",
  "quota": {
    "groups": 10,
    "persons": 80,
    "facesets": 15,
    "persons_per_group": 30,
    "faces_per_person": 10,
    "faces_per_faceset": 50
  },
  "usage": {
    "groups": 1,
    "persons": 5,
    "facesets": 6
  }
}
```

表明频率限制为每 10 秒内不超过 50 次调用（错误的调用也统计）；所能创建的人、组、人脸集合的最大数量分别为 80、10 和 15；每个组最多包含 30 个人，每个人最多包含 10 张脸，每个人脸集合最多包含 50 张脸。现在已经创建了 1 个组，5 个人，6 个人脸集合。

每次 API 调用返回的头部中都包含频率限制的信息，例如：

```
HTTP/1.1 200 OK
Content-Length: 120
Content-Type: application/json
X-RateLimit-Limit: 50
X-RateLimit-Remaining: 7
X-RateLimit-Reset: 5

{
  "status": "OK",
  .....
}
```

表示当前统计周期内总共限制 50 次调用，还剩余 7 次调用（意味着已经调用了 43 次），5 秒后将重置计数器进入下一统计周期。

一般只有发生调用频率超限错误时，才需要关注调用频率信息。错误样例：

```
HTTP/1.1 403 Forbidden
Content-Length: 30
Content-Type: application/json
X-RateLimit-Limit: 50
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 3

{
  "status": "RATE_LIMIT_EXCEEDED"
}
```

表明当前统计周期内已经超过调用频率限制，必须等待 3 秒后才能继续使用。

如果经常超过频率限制，可以考虑申请提高限制以满足使用需求。

无调用权限

系统对所有 API 的调用增加了限制，当您的账号调用了您权限以外的 API 时，系统会返回无调用权限的错误。

样例输出：

```
{
  "status": "NO_PERMISSION"
}
```

如果您确实需要用到某些 API 的功能时，您可以跟我们的商务部门联系，为您的账号开通权限。

API 参考

本章将对 SenseTime 提供的所有 API 逐一进行完整说明，供读者需要时查阅。

API 说明包括：功能介绍、参数、输出、错误信息等，不再包含上一章 [API 入门](#) 中提到过的常见的非特定 API 错误以及基本使用方法。

信息查询

/info/api

获得当前账户的使用信息，包括频率限制以及各种数量限制。

请求方式

GET

请求 URL

`https://v1-api.visioncloudapi.com/info/api`

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
rate_limit	<i>string</i>	频率限制, 50q/10s 表示每 10 秒限 50 次
expiration_date	<i>date</i>	账号到期时间, 格式满足 iso8601 标准。如果账号不存在期限限制, 则不返回此字段
quota	<i>object</i>	人、人脸集合、组的相关数量限制, 具体结构见下方返回值样例
usage	<i>object</i>	已经创建的人、人脸集合、组数量, 具体结构见下方返回值样例

```
{
  "status": "OK",
  "rate_limit": "50q/10s",
  "expiration_date": "2015-11-05T13:43:01+08:00",
  "quota": {
    "groups": 10,
    "persons": 80,
    "facesets": 15,
    "persons_per_group": 30,
    "faces_per_person": 10,
    "faces_per_faceset": 50
  },
  "usage": {
    "groups": 0,
    "persons": 0,
    "facesets": 0
  }
}
```

错误响应

状态码	status 字段	说明
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- curl 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/api?api_id=ID&api_secret=SECRET"
```

- httpie 样例

```
http GET "https://v1-api.visioncloudapi.com/info/api?api_id=ID&api_secret=SECRET"
```

/info/task

该 API 用于查询异步调用时的 HTTP 请求处理情况。异步调用具体情况请参考 [异步调用](#)。

`task_id` 有效期为 24 小时。

请求方式

GET

请求 URL

`https://v1-api.visioncloudapi.com/info/task`

请求参数

字段	类型	必需	描述
<code>api_id</code>	<i>string</i>	是	API 账户
<code>api_secret</code>	<i>string</i>	是	API 密钥
<code>task_id</code>	<i>string</i>	是	异步调用时获得的任务 ID

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
task_id	<i>string</i>	任务 ID
task_status	<i>string</i>	任务状态, 值可能为 pending (排队中)、processing (处理中)、done (已完成)、failed (失败)

当查询时任务已完成, 即 **task_status** 字段值为 done 时, 会同时返回对应的 HTTP 请求的计算结果, 结果的详细说明请参考具体使用的 API 的说明。

```
{
  "status": "OK",
  "task_id": "1df1149c3dac40b8af491721fcaa4621",
  "task_status": "done",
  "result": {
    .....
  }
}
```

错误响应

状态码	status 字段	说明
400	INVALID_TASK	task_id 无效
400	INVALID_ARGUMENT	请求参数错误, 具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期, 具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- **curl** 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/task?api_id=ID&api_secret=SECRET  
&task_id=e6ee9b9070804de486752a6eef9728cd"
```

- **httpie** 样例

```
http GET "https://v1-api.visioncloudapi.com/info/task?api_id=ID&api_secret=SECRET" \  
task_id=e6ee9b9070804de486752a6eef9728cd
```

/info/image

调用该 API，可以获得图片的相关信息，包括长度，宽度，用户自定义数据，以及图片中人脸的信息等。

请求方式

GET

请求 URL

<https://v1-api.visioncloudapi.com/info/image>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
image_id	string	是	图片 ID
with_faces	boolean	否	是否返回图片中人脸信息，为 1 时返回。默认为 0，不返回

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
image_id	<i>string</i>	图片的 ID
width	<i>integer</i>	图片的宽度 (px)
height	<i>integer</i>	图片的高度 (px)
user_data	<i>string</i>	用户自定义信息
faces	<i>array</i>	图片中包含的人脸的数组, 若未包含则为空数组

faces 数组中单个元素的结构:

字段	类型	描述
face_id	<i>string</i>	人脸的 ID
rect	<i>object</i>	人脸矩形框, top、bottom 分别表示人脸框上、下边框的纵坐标, left、right 分别表示人脸框左、右边框的横坐标
attributes	<i>object</i>	计算出的各种属性
eye_dist	<i>integer</i>	两眼瞳孔之间的距离
landmarks21	<i>array</i>	21 个关键点坐标数组, 具体情况请参考 附录-关键点
landmarks106	<i>array</i>	106 个关键点坐标数组, 具体情况请参考 附录-关键点

attributes 的结构:

字段	类型	描述
age	integer	年龄
gender	integer	性别, 小于 50 代表女性, 反之代表男性
attractive	integer	颜值
eyeglass	integer	戴眼镜, 小于 50 代表未戴眼镜, 反之代表戴眼镜
sunglass	integer	戴墨镜, 小于 50 代表未戴墨镜, 反之代表戴墨镜
smile	integer	微笑的程度, 值为 1~100, 越大代表微笑程度越深

```
{
  "status": "OK",
  "image_id": "3b0325f509124e8a8c601bfef4b052cf",
  "width": 840,
  "height": 840,
  "user_data": "test",
  "faces": [
    {
      "face_id": "91cd25ab26f6480981cb5fb0576ea31c",
      "rect": {
        "left": "152",
        "top": "152",
        "right": "762",
        "bottom": "762"
      },
      "landmarks21": [
        [
          "161.008438",
          "197.259644"
        ],
        .....
      ],
      "eye_dist": 281,
      "landmarks106": [
        [
          99.021118,
          241.338577
        ],
        .....
      ],
      "attributes": {
        "age": 20,
        "gender": 0,
        "attractive": 34,
        "eyeglass": 0,
        "sunglass": 0,
        "smile": 40
      }
    }
  ]
}
```


错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_IMAGE	<code>image_id</code> 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- **curl** 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/image?api_id=ID&api_secret=SECRET
&image_id=5053098aa6cd486bad3f042df24e63c4&with_faces=1"
```

- **httpie** 样例

```
http GET "https://v1-api.visioncloudapi.com/info/image?api_id=ID&api_secret=SECRET" \
image_id=348cbc24e57d4206b544e74b0784610f \
with_faces=1
```

/info/face

该 API 用于获取检测过的人脸的属性、关键点等相关信息，也可同时获取其所属图片的信息。

请求方式

GET

请求 URL

<https://v1-api.visioncloudapi.com/info/face>

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
face_id	<i>string</i>	是	人脸 ID
with_image	<i>boolean</i>	否	是否返回人脸所属图片信息，为 1 时返回。默认为 0，不返回

正常响应 (200)

字段	类型	描述
status	string	状态，正常为 OK
face_id	string	人脸的 ID
rect	object	人脸矩形框，top、bottom 分别表示人脸框上、下边框的纵坐标，left、right 分别表示人脸框左、右边框的横坐标
attributes	object	计算出的各种属性
eye_dist	integer	两眼瞳孔之间的距离
landmarks21	array	21 个关键点坐标数组，具体情况请参考 附录-关键点
landmarks106	array	106 个关键点坐标数组，具体情况请参考 附录-关键点
image	object	人脸所属图片信息，包含图片 ID image_id、图片宽度 width、图片高度 height 及用户自定义信息 user_data

attributes 的结构:

字段	类型	描述
age	integer	年龄
gender	integer	性别, 小于 50 代表女性, 反之代表男性
attractive	integer	颜值
eyeglass	integer	戴眼镜, 小于 50 代表未戴眼镜, 反之代表戴眼镜
sunglass	integer	戴墨镜, 小于 50 代表未戴墨镜, 反之代表戴墨镜
smile	integer	微笑的程度, 值为 1~100, 越大代表微笑程度越深

```
{
  "status": "OK",
  "face_id": "74de7d7977f44cc4a6bfa548b2b471f7",
  "rect": {
    "left": "184",
    "top": "161",
    "right": "368",
    "bottom": "345"
  },
  "landmarks21": [
    [
      "186.143021",
      "174.088364"
    ],
    .....
  ],
  "eye_dist": 108,
  "landmarks106": [
    [
      162.625763,
      192.660736
    ],
    .....
  ],
  "attributes": {
    "age": 32,
    "gender": 0,
    "attractive": 96,
    "eyeglass": 1,
    "sunglass": 0,
    "smile": 13
  },
  "image": {
    "image_id": "f495e1c7dbe14b5ba58a2df5855954e2",
    "width": 557,
    "height": 552,
    "user_data": "test"
  }
}
```

错误响应

状态码	status 字段	说明
400	INVALID_FACE	face_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- curl 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/face?api_id=ID&api_secret=SECRET
&face_id=0299a912621b44ff8ad6f0babc42f194&with_image=1"
```

- httpie 样例

```
http GET "https://v1-api.visioncloudapi.com/info/face?api_id=ID&api_secret=SECRET" \
  face_id=0299a912621b44ff8ad6f0babc42f194 \
  with_image=1
```

/info/list_persons

该 API 用以查询该账户目前所拥有的人的信息，包括人的 ID，姓名。

请求方式

GET

请求 URL

https://v1-api.visioncloudapi.com/info/list_persons

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥

正常响应（200）

字段	类型	描述
status	string	状态，正常为 OK
persons	array	该账户拥有的人的数组，若该账户下不含有人，则为空数组

persons 数组中单个元素的结构:

字段	类型	描述
person_id	string	人的 ID
name	string	人名

```
{
  "status": "OK",
  "persons": [
    {
      "person_id": "814a162c763e47849508218d822ff5ed",
      "name": "p1"
    },
    {
      "person_id": "74c1b207493d49d8bc69f7ab9996eb4f",
      "name": "test1"
    },
    {
      "person_id": "796360a1647f4924b8ecee26184810eb",
      "name": "test2"
    }
  ]
}
```

错误响应

状态码	status 字段	说明
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- **curl** 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/list_persons?api_id=ID&api_secret=SECRET"
```

- **httpie** 样例

```
http GET "https://v1-api.visioncloudapi.com/info/list_persons?api_id=ID&api_secret=SECRET"
```

/info/list_groups

该 API 用以查询该账户所拥有的组，可以获取所有组的 ID 及对应的组名。

请求方式

GET

请求 URL

https://v1-api.visioncloudapi.com/info/list_groups

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥

正常响应（200）

字段	类型	描述
status	string	状态，正常为 OK
groups	array	该账户拥有的组的数组，若不含有组，则返回空数组

groups 数组中单个元素的结构:

字段	类型	描述
group_id	string	组的 ID
name	string	组名

```
{
  "status": "OK",
  "groups": [
    {
      "group_id": "a8342f20e4cc4222962e783be54c8062",
      "name": "group1"
    },
    {
      "group_id": "c6fb3a01aad1415b9478ef3b2b34ca9a",
      "name": "group1"
    }
  ]
}
```

错误响应

状态码	status 字段	说明
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- **curl** 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/list_groups?  
api_id=ID&api_secret=SECRET"
```

- **httpie** 样例

```
http GET "https://v1-api.visioncloudapi.com/info/list_groups?api_id=ID&api_secret=SECRET"
```

/info/list_facesets

该 API 用以查询该账户所拥有的人脸集合，可以获取所有人脸集合的 ID 及对应的名称。

请求方式

GET

请求 URL

https://v1-api.visioncloudapi.com/info/list_facesets

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥

正常响应（200）

字段	类型	描述
status	string	状态，正常为 OK
facesets	array	该账户拥有的人脸集合的数组，若不含有人脸集合，则为空数组

facesets 数组中单个元素的结构:

字段	类型	描述
faceset_id	string	人脸集合的 ID
name	string	人脸集合的名称

```
{
  "status": "OK",
  "facesets": [
    {
      "faceset_id": "fd50405e24044bdb9846147145cf5c41",
      "name": "facesettest1"
    },
    {
      "faceset_id": "05f9b3eb2ec74016969b671ecad83bb5",
      "name": "faceset1"
    }
  ]
}
```

错误响应

状态码	status 字段	说明
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- **curl** 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/list_facesets?  
api_id=ID&api_secret=SECRET"
```

- **httpie** 样例

```
http GET "https://v1-api.visioncloudapi.com/info/list_facesets?  
api_id=ID&api_secret=SECRET"
```

/info/person

该 API 用以获取人的信息及其含有的人脸 ID。

请求方式

GET

请求 URL

<https://v1-api.visioncloudapi.com/info/person>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
person_id	string	是	人的 ID

正常响应 (200)

字段	类型	描述
status	string	状态，正常为 OK
name	string	人名
user_data	string	用户自定义数据
person_id	string	被查询的人的 ID
face_ids	array	该人拥有的人脸 ID 数组

```
{
  "status": "OK",
  "person_id": "5d8db5415b5a4bfdb5233edd433e13e1",
  "name": "xiaoming",
  "user_data": "ww",
  "face_ids": [
    "ed3689a183994bf0805cf6dd9e973a5b"
  ]
}
```

错误响应

状态码	status 字段	说明
400	INVALID_PERSON	person_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- curl 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/person?api_id=ID&api_secret=SECRET
&person_id=5d8db5415b5a4bfdb5233edd433e13e1"
```

- httpie 样例

```
http GET "https://v1-api.visioncloudapi.com/info/person?api_id=ID&api_secret=SECRET" \
  person_id=5d8db5415b5a4bfdb5233edd433e13e1
```

/info/group

该 API 用以查询组的相关信息，包括组的名字、组的自定义数据以及组所拥有的人的信息等。

请求方式

GET

请求 URL

<https://v1-api.visioncloudapi.com/info/group>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
group_id	string	是	组的 ID

正常响应 (200)

字段	类型	描述
status	string	状态，正常为 OK
group_id	string	组的 ID
name	string	组名
user_data	string	用户自定义数据
persons	array	人的数组，包含人的 ID 及人名

```
{
  "status": "OK",
  "group_id": "c6fb3a01aad1415b9478ef3b2b34ca9a",
  "name": "group1",
  "user_data": "xx",
  "persons": [
    {
      "person_id": "2bdfc2a84d6a4ece94566478ff06aa28",
      "name": "testp1"
    },
    {
      "person_id": "00db687180f7434da646acdc33da787a",
      "name": "xiaoming"
    }
  ]
}
```

错误响应

状态码	status 字段	说明
400	INVALID_GROUP	group_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- **curl** 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/group?api_id=ID&api_secret=SECRET  
&group_id=c6fb3a01aad1415b9478ef3b2b34ca9a"
```

- **httpie** 样例

```
http GET "https://v1-api.visioncloudapi.com/info/group?api_id=ID&api_secret=SECRET" \  
group_id=c6fb3a01aad1415b9478ef3b2b34ca9a
```

/info/faceset

该 API 用以获取人脸集合的信息及其含有的人脸的 ID。

请求方式

GET

请求 URL

<https://v1-api.visioncloudapi.com/info/faceset>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
faceset_id	string	是	人脸集合的 ID

正常响应 (200)

字段	类型	描述
status	string	状态，正常为 OK
faceset_id	string	人脸集合的 ID
name	string	人脸集合名
user_data	string	用户自定义数据
face_ids	array	该人脸集合含有的人脸 ID 数组。若未含有脸，则返回空数组

```
{
  "status": "OK",
  "faceset_id": "fd50405e24044bdb9846147145cf5c41",
  "name": "facesettest1",
  "user_data": "test",
  "face_ids": [
    "ed3689a183994bf0805cf6dd9e973a5b"
  ]
}
```

错误响应

状态码	status 字段	说明
400	INVALID_FACESET	faceset_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

- curl 样例

```
curl -X GET "https://v1-api.visioncloudapi.com/info/faceset?api_id=ID&api_secret=SECRET
&faceset_id=fd50405e24044bdb9846147145cf5c41"
```

- httpie 样例

```
http GET "https://v1-api.visioncloudapi.com/info/faceset?api_id=ID&api_secret=SECRET" \
faceset_id=fd50405e24044bdb9846147145cf5c41
```


人脸检测与分析

/face/detection

该 API 用于提供图片，进行人脸检测以及人脸分析。人脸分析包括关键点检测、属性提取，系统默认计算 21 个人脸关键点。图片的上传可采用本地上传图片或让服务器从网络地址抓取的方式。

图片必须满足如下条件：

1. 格式必须为 JPG（JPEG），BMP，PNG，GIF，TIFF 之一
2. 宽和高必须大于 8px，小于等于 4000px
3. 文件尺寸小于等于 5 MB

当上传的图片中包含有 `exif` 方向信息，我们之后会按此信息旋转、翻转后再做后续处理。并在返回结果中给出 `exif_orientation` 字段（如果没有则不输出）。

与此同时，我们还提供 `auto_rotate` 功能，用于照片方向混乱且 `exif` 方向信息不存在或不正确的情况下，根据照片中人脸方向来检查可能正确的方向，并按照正确的方向提供人脸检测结果。

请求方式

POST

请求 URL

`https://v1-api.visioncloudapi.com/face/detection`

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
file	<i>file</i>	见下方 注释	需上传的图片文件，上传本地图片进行检测时选取此参数
url	<i>string</i>	见下方 注释	图片网络地址，采用抓取网络图片方式时需选取此参数
landmarks106	<i>boolean</i>	否	值为 1 时，计算 106 个关键点。默认值为 0，不计算。 关键点具体情况请参考 附录-关键点
attributes	<i>boolean</i>	否	值为 1 时，提取人脸属性。默认值为 0，不提取
auto_rotate	<i>boolean</i>	否	值为 1 时，对图片进行自动旋转。默认值为 0 时，不旋转
user_data	<i>string</i>	否	用户自定义信息

请求参数 `file` 与 `url` 必须选择其中的一个。对于其他可选参数（如 `auto_rotate`，`landmarks106` 和 `attribute`），如果没有需求，请不要开启，这样会减少系统计算时间。

URL 中含有不少特殊字符，若将 URL 参数放入 Query String 中则需要对这些字符进行转义。目前支持 `http/https` 等协议的网络地址。下载限时 5s，超时时仍未下载完成则属于失败。参数 `file` 需把图片文件的内容以 `multipart/form-data` 的形式放到 POST 消息体中。

打开 `auto_rotate` 功能会增加运算时间，所以如果图片素材没有方向混乱且 `exif` 方向信息不存在或不正确，则不建议打开。如果有 `exif` 方向信息，则仍会先按其进行旋转、翻转后再做自动旋转检测，所得的旋转角度也是基于 `exif` 处理之后的。[/info/image](#) 接口不会返回此信息。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态，正常为 OK
image_id	<i>string</i>	图片的 ID
exif_orientation	<i>integer</i>	表示图片含有 exif 信息，返回值范围为 1~8。如果图片不含 exif 信息，则不返回该字段。返回值的具体含义请参考 http://jpegclub.org/exif_orientation.html
rotate	<i>string</i>	当 auto_rotate 功能打开时，如果发生了旋转，则会返回此字段，可能的值为 cw90, cw180, cw270，表明顺时针旋转的角度。如果没有发生旋转，则不返回此字段
width	<i>integer</i>	图片的宽度 (px)
height	<i>integer</i>	图片的高度 (px)
faces	<i>array</i>	图片中包含的人脸的数组，若未包含则为空数组

image_id、face_id 有效期均为 5 天。

当超过有效期时，若 face_id 在之前被加入到了 person 或者 faceset 中，该 face_id 所含的脸部信息永久有效，直至其被移除。但不可直接使用该 face_id 进行移除人脸外的其他操作。

faces 数组中单个元素的结构:

字段	类型	描述
face_id	string	人脸的 ID
rect	object	人脸矩形框, top、bottom 分别表示人脸框上、下边框的纵坐标, left、right 分别表示人脸框左、右边框的横坐标
attributes	object	计算出的各种属性, 结构见下方
eye_dist	integer	两眼瞳孔之间的距离
landmarks21	array	21 个关键点坐标数组, 具体情况请参考 附录-关键点
landmarks106	array	106 个关键点坐标数组, 具体情况请参考 附录-关键点

attributes 的结构:

字段	类型	描述
age	integer	年龄
gender	integer	性别, 小于 50 代表女性, 反之代表男性
attractive	integer	颜值
eyeglass	integer	戴眼镜, 小于 50 代表未戴眼镜, 反之代表戴眼镜
sunglass	integer	戴墨镜, 小于 50 代表未戴墨镜, 反之代表戴墨镜
smile	integer	微笑的程度, 值为 1~100, 越大代表微笑程度越深

```
{
  "image_id": "a5e23c448c1a4facaf06ea3dec55467a",
  "exif_orientation": 1,
  "width": 248,
  "height": 609,
  "faces": [
    {
      "face_id": "aeb91c193eb54849b1061fd3cb16ee67",
      "rect": {
        "left": 61,
        "top": 347,
        "right": 99,
        "bottom": 385
      },
      "eye_dist": 13,
      "landmarks21": [
        [
          62.498043,
          358.901459
        ],
        .....
      ],
      "attributes": {
        "age": 14,
        "gender": 0,
        "attractive": 92,
        "eyeglass": 99,
        "sunglass": 0,
        "smile": 44
      }
    }
  ],
  "status": "OK"
}
```

错误响应

状态码	status 字段	说明
400	DOWNLOAD_TIMEOUT	网络地址图片获取超时
400	CORRUPT_IMAGE	文件不是图片文件或已经损坏
400	DOWNLOAD_ERROR	网络地址图片获取失败
400	IMAGE_FILE_SIZE_TOO_BIG	图片体积过大
400	INVALID_IMAGE_FORMAT_OR_SIZE	图片大小或格式不符合要求
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/face/detection?api_id=ID&api_secret=SECRET" \
-F file=@/PATH/TO/IMAGE
```

httpie 样例

```
http -f POST "https://v1-api.visioncloudapi.com/face/detection?api_id=ID&api_secret=SECRET" \
file@/PATH/TO/IMAGE
```

/face/verification

该 API 的主要功能是将一个人脸与一个人（或者一张脸）进行对比，来判断是否为同一个人。

请求方式

POST

请求地址

<https://v1-api.visioncloudapi.com/face/verification>

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
face_id	<i>string</i>	是	人脸 ID
face2_id	<i>string</i>	见下方注释	人脸 2 的 ID，脸与脸对比时选择此参数
person_id	<i>string</i>	见下方注释	人的 ID，脸与人对比时选择此参数

请求参数 `face2_id` 与 `person_id` 必须选择其中的一个。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
same_person	<i>boolean</i>	是否为同一人, 是为 true, 否为 false。confidence 大于 0.6 时该值为 true
confidence	<i>float</i>	置信度, 值为 0~1, 值越大表示置信度越高
face_id	<i>string</i>	人脸 ID
person_id	<i>string</i>	人的 ID, 脸与人对比返回该字段
face2_id	<i>string</i>	人脸 2 的 ID, 脸与脸对比返回该字段

置信度阈值与错误率对应关系:

阈值	0.4	0.5	0.6	0.7	0.8	0.9
错误率	十分之一	百分之一	千分之一	万分之一	十万分之一	百万分之一

脸与人对比

```
{
  "same_person": false,
  "confidence": 0.368128,
  "face_id": "c74df9b413a44a9ab3060b7170989fa9",
  "person_id": "5d8db5415b5a4b4db5233edd433e13e1",
  "status": "OK"
}
```


脸与脸对比

```
{
  "same_person": false,
  "confidence": 0.368128,
  "face_id": "c74df9b413a44a9ab3060b7170989fa9",
  "face2_id": "ed3689a183994bf0805cf6dd9e973a5b",
  "status": "OK"
}
```

异步调用 (202)

当脸与脸对比时，系统会同步返回计算结果，不会发生异步调用。当脸与人对比时，只有当人所包含的人脸全部训练过时，系统会同步返回计算结果，不会发生异步调用。其他条件下都有可能发生异步调用。异步调用的详细说明请参考 [异步调用](#)。

字段	类型	描述
status	<i>string</i>	状态，正常为 OK
task_id	<i>string</i>	查询的任务 ID
task_status	<i>string</i>	任务状态，值可能为 pending（排队中）、processing（处理中）、failed（失败）

```
{
  "status": "OK",
  "task_id": "e6ee9b9070804de486752a6eef9728cd",
  "task_status": "pending"
}
```

错误响应

状态码	status 字段	说明
400	INVALID_FACE	face_id 无效
400	INVALID_FACE2	face2_id 无效
400	EMPTY_PERSON	此 person 未包含人脸
400	INVALID_PERSON	person_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/face/verification?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "face_id": "c74df9b413a44a9ab3060b7170989fa9",
      "face2_id": "ed3689a183994bf0805cf6dd9e973a5b" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/face/verification?api_id=ID&api_secret=SECRET" \
face_id=c74df9b413a44a9ab3060b7170989fa9 \
face2_id=ed3689a183994bf0805cf6dd9e973a5b
```

/face/search

该 API 用于在众多人脸中搜索出与目标人脸最为相似的一张或者多张人脸。

请求方式

POST

请求 URL

<https://v1-api.visioncloudapi.com/face/search>

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
face_id	<i>string</i>	是	人脸 ID
face_ids	<i>array</i>	见下方注释	人脸 ID 数组，用逗号分隔
faceset_id	<i>string</i>	见下方注释	人脸集合 ID
top	<i>integer</i>	否	最多返回 top 个相似人脸，值为 1~10，默认为 3

请求参数 **face_ids** 与 **faceset_id** 必须选择其中的一个。如果同时提供，系统只根据 **faceset_id** 进行搜索。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
face_id	<i>string</i>	人脸 ID
candidates	<i>array</i>	相似的人脸数组, 数组每个元素包括人脸 ID 以及置信度, 其中置信度的值为 0~1, 越大代表该脸与目标人脸相似的可信度越高

```
{
  "candidates": [
    {
      "face_id": "c74df9b413a44a9ab3060b7170989fa9",
      "confidence": 1
    }
  ],
  "face_id": "c74df9b413a44a9ab3060b7170989fa9",
  "status": "OK"
}
```

异步调用（202）

当人脸集合或 `face_ids` 中的人脸全部训练过时，系统会同步返回计算结果，不会发生异步调用。其他条件下可能发生异步调用。异步调用的详细说明请参考 [异步调用](#)。

字段	类型	描述
status	<i>string</i>	状态，正常为 <code>OK</code>
task_id	<i>string</i>	查询的任务 ID
task_status	<i>string</i>	任务状态，值可能为 <code>pending</code> （排队中）、 <code>processing</code> （处理中）、 <code>failed</code> （失败）

```
{
  "status": "OK",
  "task_id": "e6ee9b9070804de486752a6eef9728cd",
  "task_status": "pending"
}
```

错误响应

状态码	status 字段	说明
400	EMPTY_FACESET	face_ids 为空或 faceset 未包含人脸
400	INVALID_FACESET	faceset_id 无效
400	INVALID_FACE	face_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/face/search?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "face_id": "c74df9b413a44a9ab3060b7170989fa9",
      "faceset_id": "ed3689a183994bf0805cf6dd9e973a5b",
      "top": 1 }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/face/search?api_id=ID&api_secret=SECRET" \
face_id=c74df9b413a44a9ab3060b7170989fa9 \
faceset_id=ed3689a183994bf0805cf6dd9e973a5b \
top=1
```

/face/identification

该 API 将一个目标人脸与某个组中的所有人进行对比，找出几个与该人脸最相似的人，并将结果返回。

请求方式

POST

请求 URL

<https://v1-api.visioncloudapi.com/face/identification>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
face_id	string	是	人脸 ID
group_id	string	是	组的 ID
top	integer	否	最多返回 top 个人，值为 1~5，默认为 2

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
face_id	<i>string</i>	人脸 ID
group_id	<i>string</i>	组的 ID
candidates	<i>array</i>	相似的人的数组, 包括人的 ID、人名以及置信度, 其中置信度的值为 0~1, 越大代表搜索出的人与目标人脸为同一个人的可信度越高

当搜索出的人与目标人脸的置信度大于 0.6 时 candidates 中才会有结果返回。

```
{
  "candidates": [
    {
      "person_id": "00db687180f7434da646acdc33da787a",
      "name": "xiaoming",
      "confidence": 1
    }
  ],
  "face_id": "c74df9b413a44a9ab3060b7170989fa9",
  "group_id": "c6fb3a01aad1415b9478ef3b2b34ca9a",
  "status": "OK"
}
```


异步调用（202）

当组中的人脸全部训练过时，系统会同步返回计算结果，不会发生异步调用。其他条件下可能发生异步调用。异步调用的详细说明请参考 [异步调用](#)。

字段	类型	描述
status	<i>string</i>	状态，正常为 OK
task_id	<i>string</i>	查询的任务 ID
task_status	<i>string</i>	任务状态，值可能为 pending（排队中）、processing（处理中）、failed（失败）

```
{
  "status": "OK",
  "task_id": "e6ee9b9070804de486752a6eef9728cd",
  "task_status": "pending"
}
```

错误响应

状态码	status 字段	说明
400	EMPTY_GROUP	group 中未包含 person 或者所有的 person 都未包含人脸
400	INVALID_GROUP	group_id 无效
400	INVALID_FACE	face_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/face/identification?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "face_id": "c74df9b413a44a9ab3060b7170989fa9",
      "group_id": "c6fb3a01aad1415b9478ef3b2b34ca9a",
      "top": "1" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/face/identification?api_id=ID&api_secret=SECRET" \
face_id=c74df9b413a44a9ab3060b7170989fa9 \
group_id=c6fb3a01aad1415b9478ef3b2b34ca9a \
top=1
```

/face/training

可以对人脸、人、人脸集合、组进行训练，提取其中的人脸信息。训练完之后，再进行 人脸对比等操作时，会减少系统执行所用的时间。当数据量比较大时，例如人脸信息较多， 组中包含较多的人时，我们建议先进行对其进行训练，之后再行其他操作。

请求方式

POST

请求 URL

`https://v1-api.visioncloudapi.com/face/training`

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
face_id	<i>array</i>	见下方注释	人脸 ID 数组，多个人脸信息时以逗号分隔
person_id	<i>array</i>	见下方注释	人的 ID 数组，多个人时以逗号分隔
faceset_id	<i>array</i>	见下方注释	人脸集合 ID 数组，多个人脸集合时以逗号分隔
group_id	<i>array</i>	见下方注释	组的 ID 数组，多个组时以逗号分隔

请求参数 `face_id`、`person_id`、`faceset_id`、`group_id` 至少存在一个，可多选。

正常响应 (200)

字段	类型	必需	描述
status	<i>string</i>	是	状态, 正常为 OK

```
{
  "status": "ok"
}
```

异步调用 (202)

异步调用的详细说明请参考 [异步调用](#)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
task_id	<i>string</i>	查询的任务 ID
task_status	<i>string</i>	任务状态, 值可能为 pending (排队中)、processing (处理中)、failed (失败)

```
{
  "status": "OK",
  "task_id": "e6ee9b9070804de486752a6eef9728cd",
  "task_status": "pending"
}
```

错误响应

状态码	status 字段	说明
400	INVALID_GROUP	group_id 中包含无效 group
400	INVALID_PERSON	person_id 中包含无效 person
400	INVALID_FACESET	faceset_id 中包含无效 faceset
400	INVALID_FACE	face_id 中包含无效人脸
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/face/training?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "face_id": ["c74df9b413a44a9ab3060b7170989fa9"],
      "faceset_id": ["ec72017732e447a08d4516a9b294750f"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/face/training?api_id=ID&api_secret=SECRET" \
face_id=c74df9b413a44a9ab3060b7170989fa9 \
faceset_id=ec72017732e447a08d4516a9b294750f
```

人的管理

/person/create

一些 API 需要对属于同一个人的多张人脸进行操作，所以 SenseTime API 中引入了人这一概念，并提供相应的人的增删、人所含人脸的增删查等管理操作的 API。

本 API 用于创建一个人。创建人时，可以同时为其添加人脸信息，也可以加上自定义信息。

请求方式

POST

请求 URL

<https://v1-api.visioncloudapi.com/person/create>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
name	string	是	人名
face_id	array	否	人脸的 ID 数组，多个人脸时以逗号分隔
user_data	string	否	用户自定义信息

当在创建人的同时添加人脸，如果其中有一个或多个 face_id 无效，将会导致创建人的操作失败。一个 face_id 只能添加到一个 person 中。如果将一个 face_id 重复添加给另一个人时，那么此 face_id 将属于最新添加的这个人，且系统不会返回错误信息。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
person_id	<i>string</i>	人的 ID
face_count	<i>integer</i>	该人所拥有的人脸数量, 若为 0, 则不返回

```
{
  "status": "OK",
  "person_id": "542fe8c0cae841c99dddaefb14faf32b",
  "face_count": 1
}
```

错误响应

状 态 码	status 字段	说明
400	QUOTA_EXCEEDED	账户内 person 数量超限或 person 中人脸数量超限, 具体原因见 reason 字段内容
400	INVALID_FACE	face_id 中包含无效人脸
400	INVALID_ARGUMENT	请求参数错误, 具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期, 具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/person/create?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "name": "xiaoming",
      "face_id":["c74df9b413a44a9ab3060b7170989fa9"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/person/create?api_id=ID&api_secret=SECRET" \
name=xiaoming \
face_id=c74df9b413a44a9ab3060b7170989fa9
```

/person/delete

本 API 的作用是删除一个人。

删除人后，该人所拥有的人脸 ID 若未过期，则人脸 ID 不会失效。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/person/delete

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
person_id	string	是	要删除的人的 ID

正常响应（200）

字段	类型	描述
status	string	状态，正常为 OK

```
{
  "status": "OK"
}
```

错误响应

状态码	status 字段	说明
400	INVALID_PERSON	person_id 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/person/delete?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "person_id": "c74df9b413a44a9ab3060b7170989fa9" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/person/delete?api_id=ID&api_secret=SECRET" \
person_id=c74df9b413a44a9ab3060b7170989fa9
```

/person/add_face

人创建成功后，调用本 API，可以为该人添加人脸信息。

一个人脸 ID 只能属于一个人。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/person/add_face

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
person_id	string	是	人的 ID
face_id	array	是	人脸的 ID 数组，多个人脸时用逗号分隔

当同时添加多个人脸时，如果其中有一个或多个 face_id 无效时，那么其他的 face_id 也会添加失败。

一个 face_id 只能添加到一个 person 中。如果将一个 face_id 重复添加给另一个人，那么此 face_id 将属于最新添加的这个人，且系统不会返回错误信息。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态，正常为 OK
person_id	<i>string</i>	人的 ID
face_count	<i>integer</i>	本次添加人脸后该人所拥有的总人脸数量
added_count	<i>integer</i>	本次成功添加的人脸数量，之前已添加的人脸不计入在内

```
{
  "status": "OK",
  "person_id": "fb5c88ab48df4f23955fb15f01e843ed",
  "face_count": 1,
  "added_count": 0
}
```

错误响应

状态码	status 字段	说明
400	INVALID_PERSON	person_id 无效
400	EMPTY_FACE_ID	face_id 为空
400	INVALID_FACE	face_id 中包含无效人脸
400	QUOTA_EXCEEDED	person 中人脸数量超限，具体原因见 reason 字段内容
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/person/add_face?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "person_id": "fb5c88ab48df4f23955fb15f01e843ed",
      "face_id": ["0299a912621b44ff8ad6f0bab42f194"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/person/add_face?api_id=ID&api_secret=SECRET" \
person_id=fb5c88ab48df4f23955fb15f01e843ed \
face_id=0299a912621b44ff8ad6f0bab42f194
```

/person/remove_face

本 API 的作用是移除人脸，使其不再属于这个人。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/person/remove_face

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
person_id	string	是	人的 ID
face_id	array	是	人脸的 ID 数组，多个人脸信息时用逗号分隔

当同时移除多个人脸信息时，如果其中有一个或多个 face_id 无效时，不会报错，系统会移除其他有效的 face_id。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态，正常为 OK
person_id	<i>string</i>	人的 ID
face_count	<i>integer</i>	本次移除人脸后该人所拥有的人脸数量
removed_count	<i>integer</i>	此次成功移除的人脸数量

```
{
  "status": "OK",
  "person_id": "fb5c88ab48df4f23955fb15f01e843ed",
  "face_count": 0,
  "removed_count": 1
}
```


错误响应

状态码	status 字段	说明
400	INVALID_PERSON	person_id 无效
400	EMPTY_FACE_ID	face_id 为空
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/person/remove_face?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "person_id": "c74df9b413a44a9ab3060b7170989fa9",
      "face_id": ["0299a912621b44ff8ad6f0babc42f194"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/person/remove_face?api_id=ID&api_secret=SECRET" \
person_id=c74df9b413a44a9ab3060b7170989fa9 \
face_id=0299a912621b44ff8ad6f0babc42f194
```

/person/change

该 API 用于修改 person 的 `user_data` 和 `name` 信息。

请求方式

POST

请求 URL

`https://v1-api.visioncloudapi.com/person/change`

请求参数

字段	类型	必需	描述
<code>api_id</code>	<code>string</code>	是	API 账户
<code>api_secret</code>	<code>string</code>	是	API 密钥
<code>person_id</code>	<code>string</code>	是	人的 ID
<code>name</code>	<code>string</code>	见下方备注	人名
<code>user_data</code>	<code>string</code>	见下方备注	用户自定义数据

请求参数 `name`、`user_data` 两者中至少存在一个。

正常响应 (200)

字段	类型	描述
<code>status</code>	<code>string</code>	状态，正常为 <code>OK</code>

```
{
  "status": "OK"
}
```

错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_PERSON	<code>person_id</code> 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/person/change?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "person_id": "c74df9b413a44a9ab3060b7170989fa9",
      "name": "xiaohong" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/person/change?api_id=ID&api_secret=SECRET" \
person_id=c74df9b413a44a9ab3060b7170989fa9 \
name=xiaohong
```

组的管理

/group/create

为了方便同时对多个人进行操作，SenseTime API 引入了组的概念，一个组可以包含多个人。

该 API 用于创建一个组。创建组的同时可以向该组中添加人。

请求方式

POST

请求 URL

<https://v1-api.visioncloudapi.com/group/create>

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
name	string	是	组名
person_id	array	否	人的 ID 数组，多个数据时用逗号分隔
user_data	string	否	用户自定义信息

当在创建组的同时添加人时，如果其中有一个或多个 person_id 无效，将会导致创建组的操作失败。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
group_id	<i>string</i>	生成的组的 ID
name	<i>string</i>	组名
person_count	<i>integer</i>	含有的人的数量, 若为 0, 则不返回该字段

```
{
  "status": "OK",
  "group_id": "32d7b7776aa24661841cd649a968d9c4",
  "name": "group3",
  "person_count": 1
}
```

错误响应

状 态 码	status 字段	说明
400	QUOTA_EXCEEDED	账户内 group 数量超限或 group 中 person 数量超限, 具体原因见 reason 字段内容
400	INVALID_PERSON	person_id 中包含无效 person
400	INVALID_ARGUMENT	请求参数错误, 具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期, 具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/group/create?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "name": "group1",
      "person_id": ["796360a1647f4924b8ecee26184810eb"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/group/create?api_id=ID&api_secret=SECRET" \
name=group1 \
person_id=977c9277768246b2ab566d445b87e117
```

/group/delete

删除一个组。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/group/delete

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
group_id	string	是	要删除的组的 ID

正常响应（200）

字段	类型	描述
status	string	状态，正常为 OK

```
{
  "status": "OK"
}
```

错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_GROUP	<code>group_id</code> 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/group/delete?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "group_id": "32d7b7776aa24661841cd649a968d9c4" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/group/delete?api_id=ID&api_secret=SECRET" \
group_id=32d7b7776aa24661841cd649a968d9c4
```


/group/add_person

组创建成功后，调用本 API，可以为组添加人。

一个人可以属于多个组。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/group/add_person

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
group_id	string	是	组的 ID
person_id	array	是	人的 ID 数组，多个人时用逗号分隔

当同时添加多个人时，如果其中有一个或多个 person_id 无效时，那么其他的 person_id 也会添加失败。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
group_id	<i>string</i>	组的 ID
name	<i>string</i>	组名
person_count	<i>integer</i>	本次添加人后该组拥有的人的数量
added_count	<i>integer</i>	本次成功添加的人的数量, 之前已添加的人不计入在内

```
{
  "status": "OK",
  "group_id": "c6fb3a01aad1415b9478ef3b2b34ca9a",
  "name": "group1",
  "person_count": 2,
  "added_count": 0
}
```

错误响应

状 态 码	status 字段	说明
400	INVALID_GROUP	group_id 无效
400	EMPTY_PERSON_ID	person_id 为空
400	INVALID_PERSON	person_id 中包含无效 person
400	QUOTA_EXCEEDED	group 中 person 数量超限, 具体原因见 reason 字段内容
400	INVALID_ARGUMENT	请求参数错误, 具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期, 具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/group/add_person?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "group_id": "fb5c88ab48df4f23955fb15f01e843ed",
      "person_id": ["0299a912621b44ff8ad6f0babc42f194"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/group/add_person?api_id=ID&api_secret=SECRET" \
group_id=fb5c88ab48df4f23955fb15f01e843ed \
person_id=0299a912621b44ff8ad6f0babc42f194
```

/group/remove_person

本 API 的作用是移除组中的人。

移除人后，人的 ID 依然有效，只是不再属于这个组。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/group/remove_person

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
group_id	string	是	组的 ID
person_id	array	是	人的 ID 数组，多个人时用逗号分隔

当同时移除多个人时，如果其中有一个或多个 person_id 无效时，不会报错，系统会移除其他有效的 person_id。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
group_id	<i>string</i>	组的 ID
name	<i>string</i>	组名
person_count	<i>integer</i>	本次移除人后该组所拥有的人的数量
removed_count	<i>integer</i>	此次成功移除的人的数量

```
{
  "status": "OK",
  "group_id": "c6fb3a01aad1415b9478ef3b2b34ca9a",
  "name": "group1",
  "person_count": 1,
  "removed_count": 1
}
```

错误响应

状态码	status 字段	说明
400	INVALID_GROUP	group_id 无效
400	EMPTY_PERSON_ID	person_id 为空
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/group/remove_person?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "group_id": "fb5c88ab48df4f23955fb15f01e843ed",
      "person_id": ["0299a912621b44ff8ad6f0babc42f194"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/group/remove_person?api_id=ID&api_secret=SECRET" \
group_id=32d7b7776aa24661841cd649a968d9c4 \
person_id=fb5c88ab48df4f23955fb15f01e843ed
```

/group/change

该 API 用于修改 group 的 user_data 和 name 信息。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/group/change

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
group_id	string	是	组的 ID
name	string	见下方注释	组名
user_data	string	见下方注释	用户自定义数据

请求参数 name、user_data 两者中至少存在一个。

正常响应（200）

字段	类型	描述
status	string	状态，正常为 OK

```
{
  "status": "OK"
}
```

错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_GROUP	<code>group_id</code> 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/group/change?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "group_id": "32d7b7776aa24661841cd649a968d9c4",
      "name": "groupname" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/group/change?api_id=ID&api_secret=SECRET" \
group_id=c74df9b413a44a9ab3060b7170989fa9 \
name=xiaohong
```


人脸集合管理

/faceset/create

该 API 的功能是创建一个人脸集合。创建的同时可以给该人脸集合添加人脸。

请求方式

POST

请求 URL

<https://v1-api.visioncloudapi.com/faceset/create>

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
name	<i>string</i>	是	人脸集合名
face_id	<i>array</i>	否	人脸的 ID 数组，多个人脸时以逗号分隔
user_data	<i>string</i>	否	用户自定义信息

当在创建人脸集合的同时添加人脸，如果其中有一个或多个 **face_id** 无效，将会导致创建人脸集合的操作失败。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
faceset_id	<i>string</i>	人脸集合的 ID
name	<i>string</i>	人脸集合名
face_count	<i>integer</i>	该人脸集合所拥有的人脸数量, 若为 0, 则不返回该字段

```
{
  "status": "OK",
  "faceset_id": "ec72017732e447a08d4516a9b294750f",
  "name": "faceset1",
  "face_count": 1
}
```

错误响应

状态码	status 字段	说明
400	INVALID_FACE	face_id 中包含无效人脸
400	QUOTA_EXCEEDED	账户内 faceset 数量超限或 faceset 中人脸数量超限。具体原因请参考 reason 字段
400	INVALID_ARGUMENT	请求参数错误, 具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期, 具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/faceset/create?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "name": "faceset1",
      "face_id":
      ["5b0baa9cfd7e4359b51923a4361b9a3d", "5b9kxxfw337e4359b51923a4361b9a3d"],
      "user_data": "USER_DATA" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/faceset/create?api_id=ID&api_secret=SECRET" \
name=faceset1 \
face_id=5b0baa9cfd7e4359b51923a4361b9a3d\
user_data=USER_DATA
```

/faceset/delete

该 API 的功能是删除一个人脸集合。

请求方式

POST

请求 URL

https://v1-api.visioncloudapi.com/faceset/delete

请求参数

字段	类型	必需	描述
api_id	string	是	API 账户
api_secret	string	是	API 密钥
faceset_id	string	是	要删除的人脸集合的 ID

正常响应 (200)

字段	类型	描述
status	string	状态，正常为 OK

```
{
  "status": "OK"
}
```

错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_FACESET	<code>faceset_id</code> 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/faceset/delete?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "faceset_id": "ec72017732e447a08d4516a9b294750f" }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/faceset/delete?api_id=ID&api_secret=SECRET" \
faceset_id=ec72017732e447a08d4516a9b294750f
```

/faceset/add_face

人脸集合创建成功后，调用本 API，可以向人脸集合中添加人脸。

请求方式

POST

请求 URL

`https://v1-api.visioncloudapi.com/faceset/add_face`

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
faceset_id	<i>string</i>	是	人脸集合的 ID
face_id	<i>array</i>	是	人脸的 ID 数组，多个人脸时以逗号分隔

当同时添加多个人脸时，如果其中有一个或多个 `face_id` 无效，那么其他的 `face_id` 也会添加失败。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
faceset_id	<i>string</i>	人脸集合的 ID
name	<i>string</i>	人脸集合名
face_count	<i>integer</i>	本次添加人脸后该人脸集合拥有的人脸数量
added_count	<i>integer</i>	本次成功添加的人脸数量, 之前已添加的人脸不计入在内

```
{
  "status": "OK",
  "faceset_id": "ec72017732e447a08d4516a9b294750f",
  "name": "faceset1",
  "face_count": 1,
  "added_count": 0
}
```

错误响应

状态码	status 字段	说明
400	INVALID_FACESET	faceset_id 无效
400	EMPTY_FACE_ID	face_id 为空
400	INVALID_FACE	face_id 中包含无效人脸
400	QUOTA_EXCEEDED	faceset 中人脸数量超限，具体原因见 reason 字段内容
400	INVALID_ARGUMENT	请求参数错误，具体原因见 reason 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 reason 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/faceset/add_face?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "faceset_id": "ec72017732e447a08d4516a9b294750f",
      "face_id": ["5b0baa9cfd7e4359b51923a4361b9a3d", "5b9kxxfw337e4359b51923a4361b9a3d"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/faceset/add_face?api_id=ID&api_secret=SECRET" \
faceset_id=ec72017732e447a08d4516a9b294750f \
face_id=5b0baa9cfd7e4359b51923a4361b9a3d,5b9kxxfw337e4359b51923a4361b9a3d
```


/faceset/remove_face

调用本 API 会移除人脸集合所含有的人脸，使其不再属于该人脸集合。

请求方式

POST

请求 URL

`https://v1-api.visioncloudapi.com/faceset/remove_face`

请求参数

字段	类型	必需	描述
api_id	<i>string</i>	是	API 账户
api_secret	<i>string</i>	是	API 密钥
faceset_id	<i>string</i>	是	人的 ID
face_id	<i>array</i>	是	人脸的 ID 数组，多个人脸时以逗号分隔

当同时移除多个人脸时，如果其中有一个或多个 `face_id` 无效时，不会报错，系统会移除其他有效的 `face_id`。

正常响应 (200)

字段	类型	描述
status	<i>string</i>	状态, 正常为 OK
faceset_id	<i>string</i>	人脸集合的 ID
name	<i>string</i>	人脸集合名
face_count	<i>integer</i>	本次移除人脸后该人脸集合所拥有的人脸数量
removed_count	<i>integer</i>	此次成功移除的人脸数量

```
{
  "status": "OK",
  "faceset_id": "ec72017732e447a08d4516a9b294750f",
  "name": "faceset1",
  "face_count": 0,
  "removed_count": 1
}
```

错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_FACESET	<code>faceset_id</code> 无效
400	EMPTY_FACE_ID	<code>face_id</code> 为空
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 样例

```
curl -X POST "https://v1-api.visioncloudapi.com/faceset/remove_face?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "faceset_id": "ec72017732e447a08d4516a9b294750f",
      "face_id": ["5b0baa9cfd7e4359b51923a4361b9a3d"] }'
```

httpie 样例

```
http POST "https://v1-api.visioncloudapi.com/faceset/remove_face?api_id=ID&api_secret=SECRET" \
faceset_id=ec72017732e447a08d4516a9b294750f \
face_id=5b0baa9cfd7e4359b51923a4361b9a3d
```

/faceset/change

该 API 用于修改人脸集合的 `user_data` 和 `name` 信息。

请求方式

POST

请求 URL

`https://v1-api.visioncloudapi.com/faceset/change`

请求参数

字段	类型	必需	描述
<code>api_id</code>	<code>string</code>	是	API 账户
<code>api_secret</code>	<code>string</code>	是	API 密钥
<code>faceset_id</code>	<code>string</code>	是	人脸集合的 ID
<code>name</code>	<code>string</code>	见下方注释	人脸集合名
<code>user_data</code>	<code>string</code>	见下方注释	用户自定义数据

请求参数 `name` 与 `user_data` 二者中至少存在一个。

正常响应 (200)

字段	类型	描述
<code>status</code>	<code>string</code>	状态，正常为 <code>OK</code>

```
{
  "status": "OK"
}
```

错误响应

状态码	<code>status</code> 字段	说明
400	INVALID_FACESET	<code>faceset_id</code> 无效
400	INVALID_ARGUMENT	请求参数错误，具体原因见 <code>reason</code> 字段内容
401	UNAUTHORIZED	账号或密钥错误
401	KEY_EXPIRED	账号过期，具体情况见 <code>reason</code> 字段内容
403	RATE_LIMIT_EXCEEDED	调用频率超出限额
403	NO_PERMISSION	无调用权限
404	NOT_FOUND	请求路径错误
500	INTERNAL_ERROR	服务器内部错误

使用样例

curl 示例

```
curl -X POST "https://v1-api.visioncloudapi.com/faceset/change?api_id=ID&api_secret=SECRET" \
-H "Content-Type: application/json" \
-d '{ "faceset_id": "ec72017732e447a08d4516a9b294750f",
      "name": "NAME",
      "user_data": "USER_DATA" }'
```

httpie 示例

```
http POST "https://v1-api.visioncloudapi.com/faceset/change?api_id=ID&api_secret=SECRET" \
faceset_id=ec72017732e447a08d4516a9b294750f \
name=NAME \
user_data=USER_DATA
```

附录

利用 [/face/detection](#) API 进行 关键点检测时，可根据需要选择 21、106 个关键点，不同数量关键点的具体标记位置请见示意图。

关键点标号从 0 开始，请注意您使用的编程语言数组下标是否从 0 开始。

21 个关键点

当关键点数量为 21 时，关键点标记眉中间及两端、左右眼瞳孔及眼角、鼻尖及鼻翼、嘴角及嘴唇中间的位置， 其具体标记情况如示意图所示：



106 个关键点

当关键点数量为 106 时，标记眉、眼、鼻、嘴等脸部关键位置的轮廓以及脸部轮廓，具体标记情况如示意图所示：

