# 开放平台架构设计说明书

编制部门：公共技术平台

编 制 人：梁晓发

审 核：

会 签：

批 准：

|  |  |  |  |
| --- | --- | --- | --- |
| **文件修订履行** | | | |
| 版本 | 修订内容简述 | 生效日期 | 修订部门/修订者 |
| 1 | 新制订 | 2017-8-30 | 公共技术平台/梁晓发 |
| 2 | 修改 | 2017-9-8 | 公共技术平台/梁晓发 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 一.引言

## 1.1背景与目的

要设计AI开放平台，先要清楚基于该平台这个层面上，需要依赖的是什么？被调用的客户端又是什么？根据目前得知的业务需求，前面由安卓使用HTTP协议调用，需要依赖的是C++编写的SDK，用于分析人脸识别结果。由于目前的用户基数是数千级别的，所以一开始在服务器配置选型上，还没涉及到超高性能的要求；但是，围绕着后期的持续发展的用户数以及并发请求数的增多，平台的架构设计必须一开始就定位为可伸缩，可扩展，高安全，可维护性强的架构系统。

可伸缩，表示可以线性添加服务器构成分布式集群部署的方式处理业务请求。

可扩展，意味着平台层面上的扩展力强，无论后期添加多个服务中间件或者并且后续有新需求，都能够在极少改动原有代码或基本不需要改动源代码的基础上实现。

高安全，因为客户端调用的是安卓系统，传输的人脸识别数据以及人脸图片数据属于机密属性，因此，需要考虑防窃听防篡改措施，因此需要使用SSL/TLS证书。

可维护性强，上线后需要维护，维护成本高低取决于代码质量以及逻辑分层设计，这可以通过编码规范，数据库设计规范等等手段去约束。

## 1.2产品信息

AI开放平台

## 1.3软件名称

UBT\_AI\_PLATFORM V1.0.0

## 1.4术语与略缩语

| 术语/缩略语 | 说明 |
| --- | --- |
| QPS | Query Per Seconds 服务器每秒能处理的请求数. |
| PV | Page View 页面浏览量,也即后台接口每天请求总数 |
| ART | Average Response Time 平均响应时间 |
| AC | Agency Center代理中心 |

## 1.5参考资料

【无】

# 二.系统概述

## 2.1平台吞吐量理论值预估【单机】

仅对单机进行吞吐量的理论值预估，后续究竟需要多少台服务器，主要根据实际的整体接口调用访问量进行动态线性伸缩。理论上整体吞吐量=单机吞吐量\*服务器数量。

要预估吞吐量，重要的参数是QPS，开放平台的每个API接口每秒的并发处理能力不一样，因此，QPS指所有API接口的QPS平均值。

QPS指代最佳线程数。超出此线程数的并发请求会影响吞吐量，此时需要横向扩展服务器。

### 2.1.1 最高理论值预估

QPS：

根据每秒并发请求数60个来评估。(具体需要经过压测，QPS=30是折中参考值)

公式：(一天总秒数\*QPS)=86400\*60=5184000(PV)

### 2.1.2 参考二八原则理论值预估

参考理论：二八原则，即每天80%的请求落在20%的时间上.

QPS：

根据每秒并发请求数60个来评估。(具体需要经过压测，QPS=60是折中参考值)

理论日PV值：

以QPS为60，根据二八原则可以得出服务器吞吐量每天51.84W次请求。

公式：(一天总秒数\*20%\*QPS)/80% = (86400\*0.2\*30)/0.8 =1296000 (PV)

## 2.2上线时整体技术指标预估

1. 图片数据存储量

(1).使用云存储，理论上无上限

(2).自建文件系统，上限即为磁盘的容量

2.存储吞吐量:256MBps，存储IOPS:20000

3.QPS>=150（服务器集群部署）

4.响应时间 < 1S （不含网络开销）

5.最大日均PV：350W左右

## 2.3 开发者帐号与访问限制

要使用优必选AI开放平台，需要先注册为平台开发者。开发者角色有两种：优必选公司内部帐号以及第三方帐号。

由于平台部署在云端，在网络访问速度上来讲，公司内部帐号以及第三方帐号的访问速度是一致的。但是，第三方帐号的一些约束条件，比如同一帐号的并发请求数，每天访问限次数，超过次数需要付费等等规则，在公司内部帐号中完全没有这种约束体现。

下面是公司内部帐号以及第三方开发者帐号在平台中的一些差异。

|  | 公司内部帐号 | 第三方帐号 |
| --- | --- | --- |
| 每个接口每天限次访问 | 无限 | 1000次/天免费 |
| 并发请求数 | 无限 | 2个并发 |
| 存储空间 | 无限 | 受限 |
| 超过限制付费 | 不需要 | 需要 |
| 服务降级,过载保护 | 不会 | 会 |
| 创建应用数量 | 不受限 | 20个上限 |

说明：

1. 每个接口每天限次访问。从每天0时开始算起，直至23时59分59秒为止，每个接口的调用次数会累计，超出限制次数则返回“访问失败：您的接口免费调用次数已用完”提示。公司内部帐号无限制。
2. 并发请求数。是指同一个开发者帐号对同一个接口的并发请求限制。第三方帐号只有2个并发请求数。公司内部帐号无限制。
3. 存储空间。注册人脸图片的存储空间。第三方帐号受限，比如500G免费空间，公司内部帐号不受限。
4. 超过限制付费。接口每天调用次数以及存储空间如果超出了免费调用次数，如果需要扩展则需要付费。公司内部帐号不需要付费。
5. 服务降级。当服务器在同一个时刻，并发请求数超出一个阀值，公司内部帐号的请求会进入队列排队，而第三方帐号的请求会直接拒绝(服务器繁忙请稍后再试)。

## 2.4 Access\_Token

Access\_Token即为访问令牌，相当于开放API的访问通行证，具备Access\_Token的访问都是合法的访问。要获取Access\_Token访问令牌，必须先成为平台的开发者。获取Access\_Token访问令牌的过程见下图。



上图中的创建应用之后，平台会为该应用分配api\_key和secret\_key密钥对，每创建一个应用，都会产生一对密钥对。然后以此密钥对请求获取Access\_token即可。

第三方开发者帐号创建应用的数量受限制，这里限定20个上限。

Access\_Token是具备超时时间限制的，这里超时时间为30天。Access\_Token在获取之后，一般由客户端缓存，后续多个请求中复用。在超时之后重新获取即可。Access\_Token的获取次数目前没有限制。

## 2.5技术选型

人脸识别后台技术选型：

开发语言：Java

MVC框架：SpringMVC

数据库：MySQL5.6

搜索引擎：ElasticSearch

文件系统：Fastdfs

缓存：Redis

Web容器：tomcat

反向代理服务器：nginx

日志组件：log4j2

消息队列：RocketMQ

## 2.6开发环境

操作系统：Windows7或Ubuntu/centos

Java：Jdk8

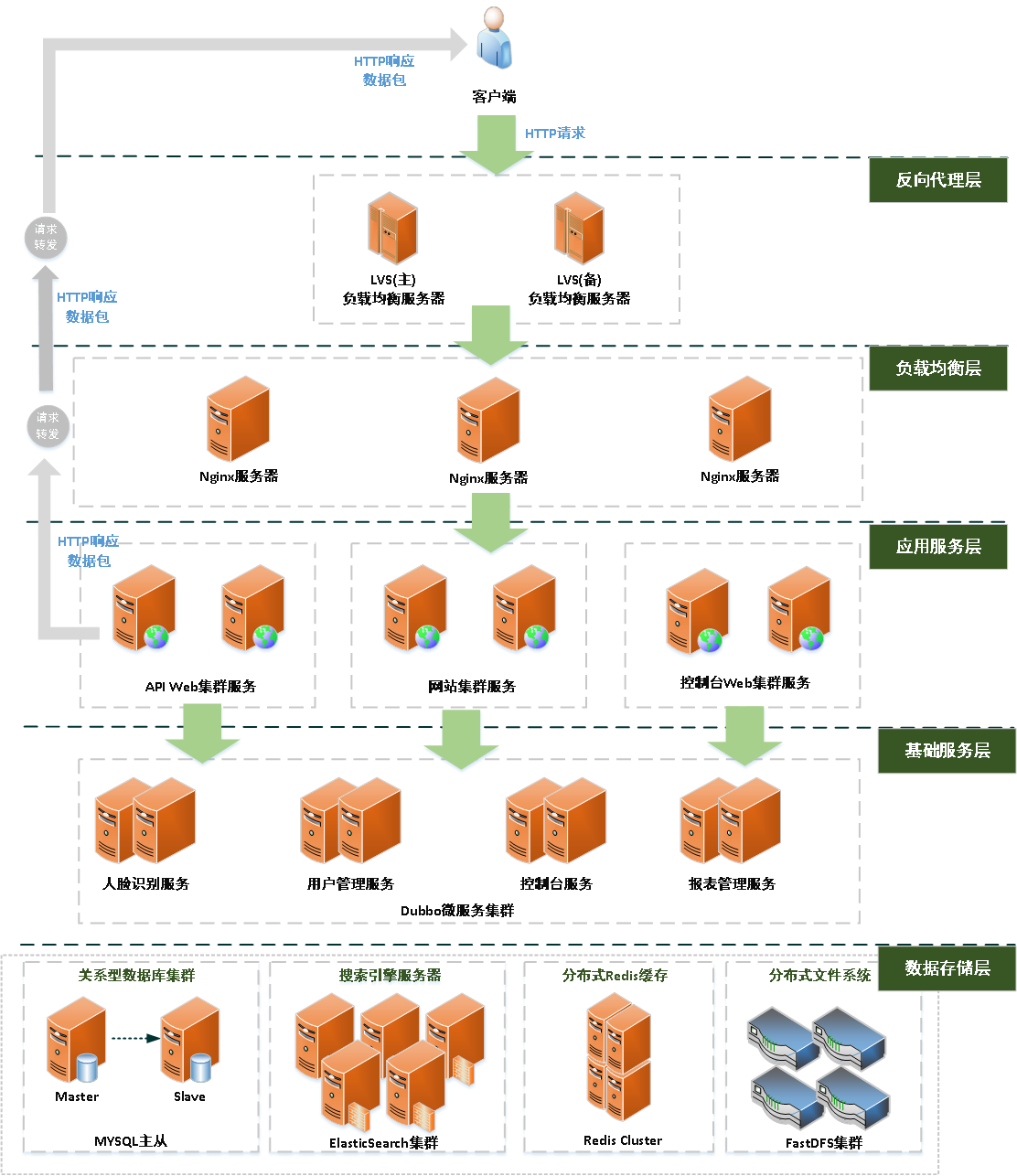
IDE：Eclipse

Maven：3.3

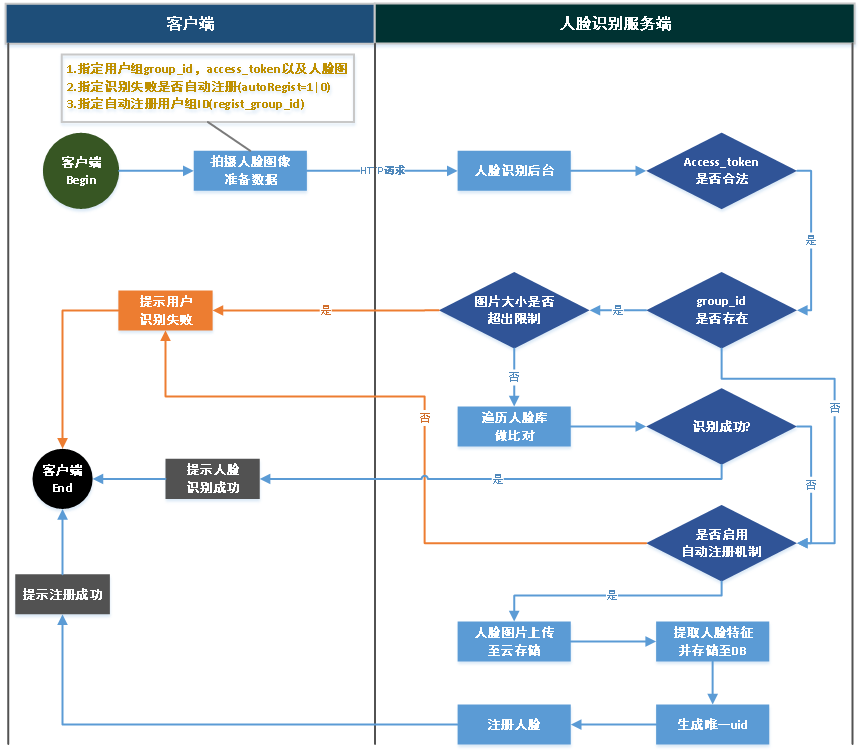
版本控制：Git

接口管理系统：RAP

# 3.逻辑架构图



# 人脸识别逻辑流程图



# 三.接口简介

HI，你好，欢迎使用优必选AI开放平台服务API。

本文档主要针对API开发者，描述优必选人脸识别服务API接口等技术内容。如果您有任务疑问，可以通过以下的方式联系我们。

* 在优必选开发平台控制台内提交工单。
* 加QQ客服，QQ号码：5201314

## 接口能力

| **接口名称** | **接口能力简要描述** |
| --- | --- |
| 人脸检测 | 检测人脸并返回人脸坐标位置 |
| 人脸比对 | 返回两两比对的人脸相似值 |
| 人脸识别 | 在人脸库中查找相似的人脸 |
| 人脸认证 | 识别上传的图片是否为指定用户 |
| 人脸库设置 | 对人脸库的相关操作，如注册、删除、更新、查找用户信息等 |
| 物体识别 | 检测图片中物体的坐标位置以及名称 |

## 请求格式

POST方式调用。

(1)如果只是提交表单

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

(2)如果需要上传图片

Content-Type:multipart/form-data

## 返回格式

JSON格式

## 请求限制

上传图片使用原生二进制流的方式上传

****请求格式支持：****PNG、JPG、JPEG、BMP，****不支持GIF图片****

| **接口名称** | **图片编码后大小限额** |
| --- | --- |
| 人脸检测 | 小于2M |
| 人脸比对 | 单次传入的两张图片，小于10M |
| 人脸识别 | 小于2M |
| 人脸认证 | 小于10M |
| 人脸库设置 | 小于10M |
| 人脸注册 | 小于2M |
| 物体识别 | 小于10M |

# 四.调用方式

## 请求URL数据格式

向API服务器提交的请求URL，必须要带上accessToken参数。

accessToken：必填参数，参考“accessToken获取”

注意：accessToken有效期为30天，可以在客户端中缓存。

POST中请求参数请按照文档说明进行调用。

比如人脸识别API，使用https post请求发送的URL为：

**[https://open-api.ubtrobot.com/face/faceset/user/identify/v1?accessToken=TWaGbQs/PWV3B0W8RKNSeEmSN4VsbknHxToYouLkoKk=](https://open-api.ubtplatform.com/face/faceset/user/identify/v1?access_token=TWaGbQs/PWV3B0W8RKNSeEmSN4VsbknHxToYouLkoKk=)**

**把accessToken作为请求参数加在POST请求URL的后面。**

# 五.accessToken接口

## 3.1获取accessToken

**获取accessToken的授权服务地址：**

**<https://open-api.ubtrobot.com/oauth/2.0/token>**

向上面的地址发送POST请求，并在request body中提供如下的参数

* grantType：必须的参数，固定为client\_credentials
* apiKey：必须的参数，创建应用后生成的API\_KEY
* secretKey：必须的参数，创建应用后生成的SECRET\_KEY

服务器返回的json参数说明如下：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| data | object[] | 响应内容 |
| +accessToken | String | 返回的accessToken |
| +expireTime | Int32 | accessToken的有效期(秒) |

例如：

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "data": {  "accessToken": "DPAIHDPIAHDPAIHDPIAHPDIAD",  "expireTime": 2592000  }  } |

**如果请求失败，比如提交错误的api\_key和secret\_key，那么返回下面的json数据**

|  |
| --- |
| {  "resultCode": 1306,  "message": "非法的api\_key或secret\_key"  } |

**具体的状态码说明请参见“状态码说明表格”**

# 六.人脸识别相关API接口

## 4.1人脸注册

用于从人脸库中新增用户，可以设定多个用户所在组，及组内用户的人脸图片，

典型应用场景：构建您的人脸库，如会员人脸注册，已有用户补全人脸信息，公司上下班人脸打卡等。

人脸库、用户组、用户、用户下的人脸层级关系如下所示：

|  |
| --- |
| +人脸库  +用户组A  +用户01  +人脸01  +人脸02  +用户02  +人脸01  +用户组B  +用户01  +人脸01  +用户02  +人脸01  +用户组C  +用户组D |

**说明：关于人脸库的设置**

* **人脸库：一个开发者帐号只能创建一个人脸库**
* **用户组：一个人脸库下面的用户组数量没有限制。用户组使用groupId表示。**
* **用 户：一个用户组下面，可以注册300000个用户。用户使用uid表示。**
* **人 脸：一个用户下面，可以注册10张人脸**

**说明：**

1. **人脸注册后，生效时间最长为30秒，之后便可以进行识别和认证的操作。**
2. **注册的人脸，建议为正面照。**
3. **上传的图片中如果不存在人脸，会提示注册失败。**
4. **上传的图片中只能存在一张人脸，超出一张人脸会提示注册失败。**
5. **对同一个用户（uid）可以反复注册最多10张人脸。**
6. **注册时指定用户UID如果在人脸库中已经存在，那么默认会append添加一条记录。如果需要替换换来的uid人脸记录可以指定actionType=replace**
7. **同一个用户(uid)可以注册在多个用户组下**

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com/](https://open-api.ubtrobot.com/oauth/2.0/token)face/faceset/user/add/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 用户组ID(只能包含字母数字下划线)，长度限制为128B |
| uid | String | 是 | 用户ID(只能包含字母数字下划线)，长度限制为128B |
| userInfo | String | 是 | 用户信息，可以输入用户名，或者更多用户信息使用json包装，长度限制1024B |
| image | byte[] | 是 | 人脸图片的二进制字节数组，图片大小不能超过10M |
| actionType | String | 否 | 指定注册时的动作是新增或替换。  参数有:append(新增)，replace(替换)。默认为append(新增)。如指定替换则对已注册的相同uid的用户进行替换。 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |

**示例**

|  |
| --- |
| {  "resultCode":200,  "message": "人脸注册成功"  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.2人脸更新

指定用户组groupId以及该用户组下面的uid执行更新操作，新上传的数据会覆盖掉该用户组下uid的所有数据。

需要指定actionType。

说明：

actionType=append，无论用户是否存在，都会新增一条记录。

actionType=replace，如果用户已经存在，则修改记录，如果用户不存在，则新增记录

一般应用场景：

人脸注册后用户的信息需要变更，可以使用人脸更新来变更userInfo。

或者用户整容了，又或者用户剪光头了，可以使用人脸更新来变更人脸图片。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com/](https://open-api.ubtrobot.com/oauth/2.0/token)face/faceset/user/update/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 用户组ID(只能包含字母数字下划线)，长度限制为128B |
| uid | String | 是 | 用户ID(只能包含字母数字下划线)，长度限制为128B |
| userInfo | String | 是 | 用户信息，可以输入用户名，或者更多用户信息使用json包装，长度限制1024B |
| image | byte[] | 是 | 人脸图片的二进制字节数组。图片大小不能超过10M |
| actionType | String | 否 | 指定动作是新增或替换。  参数有:append，replace。默认为replace。 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |

**示例**

|  |
| --- |
| {  "resultCode":200,  "message": "人脸更新成功"  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.3人脸删除

指定用户组groupId和用户uid进行用户信息的删除操作。此操作需要谨慎，数据删除后不可逆。

说明：

1. 删除的内容，包括人脸图片信息和用户信息(userInfo)
2. 只会删除指定groupId下的uid，如果该uid在其他groupId下还存在，则其他groupId下的uid不会受到影响
3. 如果误删除，需要重新注册用户信息补充。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com/](https://open-api.ubtrobot.com/oauth/2.0/token)face/faceset/user/delete/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 用户组ID(只能包含字母数字下划线)，长度限制为128B |
| uid | String | 是 | 用户ID(只能包含字母数字下划线)，长度限制为128B |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |

**示例**

|  |
| --- |
| {  "resultCode":200,  "message": "人脸用户信息删除成功"  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.4用户信息查询

查询人脸库中指定groupId和uid用户的详细信息。

返回的内容包括用户注册时的userInfo 以及图片地址。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/user/get/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 用户组ID(只能包含字母数字下划线)，长度限制为128B |
| uid | String | 是 | 用户ID(只能包含字母数字下划线)，长度限制为128B |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回的记录数 |
| result | data[] | 数组，长度等于resultNum |
| + groupId | String | 用户组id |
| + uid | String | 用户id |
| + userInfo | String | 用户信息 |
| + faceUrl | String | 用户注册时的图片地址完整url |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 3,  result: [  {  "groupId": "KTC",  "uid": "FDSFB76965",  "userInfo": "{"name":"张三","age":33,"dept":"ITD"}",  "faceUrl": "http://assets.ubtrobot.com/images/00001.jpg"  },  {  "groupId": "KTC",  "uid": "FDSFB76965",  "userInfo": "{"name":"张三2","age":33,"dept":"ABC"}",  "faceUrl": "http://assets.ubtrobot.com/images/00002.jpg"  },  {  "groupId": "KTC",  "uid": "FDSFB76965",  "userInfo": "{"name":"张三3","age":33,"dept":"JJJ"}",  "faceUrl": "http://assets.ubtrobot.com/images/00003.jpg"  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.5组列表查询

用户查询用户组的列表，请求时需要指定start和limit，分页取。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/group/getlist/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| start | Int32 | 是 | 开始行 |
| limit | Int32 | 是 | 每页取多少行 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回的记录数 |
| result | data[] | 字符串数组，长度等于resultNum |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 3,  result: [  "GroupA","GroupB","GroupC"  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.6组内用户列表查询

指定用户组groupId，查询该用户组下的所有注册用户信息。需要指定start和limit，分页查询。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/group/getuser/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 用户组id |
| start | Int32 | 是 | 开始行 |
| limit | Int32 | 是 | 每页取多少行 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回的记录数 |
| result | data[] | 数组，长度等于resultNum |
| + groupId | String | 用户组id |
| + uid | String | 用户id |
| + userInfo | String | 用户信息 |
| + faceUrl | String | 用户注册时的图片地址完整url |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 3,  result: [  {  "groupId": "KTC",  "uid": "FDSFB76965",  "userInfo": "{"name":"张三","age":33,"dept":"ITD"}",  "faceUrl": "http://assets.ubtrobot.com/images/00001.jpg"  },  {  "groupId": "KTC",  "uid": "FDSFB76965",  "userInfo": "{"name":"张三2","age":33,"dept":"ABC"}",  "faceUrl": "http://assets.ubtrobot.com/images/00002.jpg"  },  {  "groupId": "KTC",  "uid": "FDSFB76965",  "userInfo": "{"name":"张三3","age":33,"dept":"JJJ"}",  "faceUrl": "http://assets.ubtrobot.com/images/00003.jpg"  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.7组间复制用户

用于在某注册用户的信息包括人脸图片以及userInfo从一个用户组复制到另外一个用户组。

注：

复制之后，该用户uid会同时存在于两个用户组中。

如果指定的目标用户组不存在，会自动创建这个用户组

下面几种情况，操作会失败：

1. 源用户组不存在
2. 指定uid的用户不存在

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com/](https://open-api.ubtrobot.com/oauth/2.0/token)face/faceset/user/delete/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 目标用户组ID，用户信息复制至此用户组下(只能包含字母数字下划线)，长度限制为128B |
| uid | String | 是 | 用户ID(只能包含字母数字下划线)，长度限制为128B |
| srcGroupId | String | 是 | 源用户组ID |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |

**示例**

|  |
| --- |
| {  "resultCode":200,  "message": "请求成功"  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.8组内用户删除

指定用户组groupId和多个用户的进行批量删除。

注：与人脸删除的区别是，人脸删除只能指定一个用户进行删除，组内用户删除可以指定多个用户。指定的参数uid是多个用户id的合成，使用逗号隔开,例如：uid=”11,22,33”

说明：

1.删除的内容，包括人脸图片信息和用户信息(userInfo)

2.只会删除指定groupId下的用户，该用户如果在其他组也注册了则不会收到影响

3.如果误删除，需要重新注册用户信息补充。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/group/deleteuser/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 用户组ID(只能包含字母数字下划线)，长度限制为128B |
| uid | String | 是 | 用户ID，如果需要删除多个用户则使用逗号隔开，例如：uid="A1,A2,A3"， |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |

**示例**

|  |
| --- |
| {  "resultCode":200,  "message": "请求成功"  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.9人脸检测

人脸检测接口请求得到图片中的人脸坐标位置(left,top,width,height)，以及人脸相关属性（性别，年龄）。一张图片中支持多张人脸的检测，一张图片最多支持100张人脸的检测。检测的响应速度与图片中的人脸数量有直接关系。

一般应用场景，人脸属性分析，人脸营销活动，人脸图片截取等等

说明：

默认只返回人脸坐标，如果需要返回更多的人脸属性比如性别，年龄等需要提供额外的参数。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/detect/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| image | byte[] | 是 | 图片的二进制字节数组，图片大小不得超过2MB |
| faceFields | String | 否 | 包括age,sex。逗号分隔。默认只返回人脸框，此参数需要指定返回额外的人脸属性。 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回数据数组的长度,表示图片中有多少张人脸 |
| result | Data[] | 返回结果 |
| +location | Object | 坐标对象 |
| +left | Int32 | 左上角X坐标 |
| +top | Int32 | 左上角Y坐标 |
| +width | Int32 | 宽度 |
| +height | Int32 | 高度 |
| +sex | Int32 | 1:男，0:女 |
| +age | Int32 | 年龄 |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 3,  "result": [  {  "location": {  "left": 100,  "top": 120,  "width": 200,  "height": 230  },  sex:1,  age:33  },  {  "location": {  "left": 220,  "top": 250,  "width": 260,  "height": 280  }  sex:0,  age:23  },  {  "location": {  "left": 360,  "top": 400,  "width": 300,  "height": 400  },  sex:1,  age:29  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.10人脸比对

该请求用于比对多张图片中的人脸相似度并返回两两比对的得分，可用于判断两张脸是否是同一人的可能性大小。

典型应用场景：如人证合一验证，用户认证等，可与您现有的人脸库进行比对验证。

说明：返回的分数是一个小于0的浮点数，1表示两张人脸完全相同。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/match/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| image1 | byte[] | 是 | 人脸图片的二进制字节数组，图片大小不得超过10MB |
| image2 | byte[] | 是 | 人脸图片的二进制字节数组，图片大小不得超过10MB |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回数据数组的长度,这里固定为1 |
| result | Data[] | 返回结果 |
| + | score | 分数 |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 1,  "result": [  {  "score": 0.9125487  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.11人脸识别

上传人脸图片，与指定用户组下的已注册的人脸图片进行比对，返回相似度最高的那个用户的相关信息。前提是已经创建了一个人脸库。

一般应用场景，考勤签到，安防监控等等。

性能说明：上传的图片中包括的人脸数越多，性能相对越慢。建议一张图片仅包含一张人脸。

说明：

1. 人脸识别返回的是相似度最高的那个用户的信息。如果该用户注册了多张人脸，返回指定人脸数的比对结果。
2. 开放平台不会返回是否同一个人，只返回相似度。
3. 根据经验值，相似度高于0.6的可以判断为同一个人。
4. 支持图片中多张人脸的识别场景。图片最多支持20张人脸的识别。

后台逻辑：

可以指定多个用户组，使用逗号隔开，后台会根据上传的图片做一次特征提取，然后与指定用户组下的已注册人脸数进行遍历，一一比对，最终返回相似度最高的用户信息。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/user/identify/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 指定用户组ID，多个使用逗号隔开. |
| image | byte[] | 是 | 人脸图片的二进制字节,图片大小不得超过2MB |
| userTopNum | Int32 | 否 | 用户注册多张人脸图片情况下，指定识别返回结果数。默认为1 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回数据数组的长度.这里只返回1 |
| result | Data[] | 返回结果 |
| +groupId | String | 用户组ID |
| +uid | String | 用户ID |
| +faces | Data[] | 用户注册的人脸信息,长度小于等于请求参数user\_top\_num |
| +userInfo | String | 用户信息 |
| +score | Float | 相似度分值 |
| +faceImagePath | String | 人脸图片的访问URL地址 |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 3,  "result": [  {  "groupId": "AA001",  "uid": "U6767896",  "faces": [  {  "userInfo": "小明人脸1",  "score": 0.871814,  "face\_image\_path": "http://assets-ai.ubtrobot.com/su/12313.jpg"  },  {  "userInfo": "小明人脸2",  "score": 0.758748,  "face\_image\_path": "http://assets-ai.ubtrobot.com/su/12334.jpg"  }  ]  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.12人脸相似度查找

上传人脸图片，与指定用户组下的已注册的人脸图片进行比对，返回相似度最高的N个用户的相关信息。前提是已经创建了一个人脸库。

注：人脸相似度查找与人脸识别不一样，人脸相似度查找是返回相似度分值最高的N个用户的信息，人脸识别只返回相似度分值最高的那个用户的信息。

性能说明：上传的图片中包括的人脸数越多，性能相对越慢。建议一张图片仅包含一张人脸。

一般应用场景：考勤签到，安防监控等等。

说明：

1人脸相似度查找返回的是相似度最高的N个用户的信息。如果该用户注册了多张人脸， 返回指定人脸数的比对结果。

2开放平台不会返回是否同一个人，只返回相似度。

3根据经验值，相似度高于0.6的可以判断为同一个人。

4支持图片中多张人脸的识别场景。图片最多支持20张人脸的识别。

5返回的识别结果是一个二维数组，第1层表示上传图片中出现的人脸数，第2层表示人脸对应的识别结果。

后台逻辑：

可以指定多个用户组，使用逗号隔开，后台会根据上传的图片做一次特征提取，然后与指定用户组下的已注册人脸数进行遍历，一一比对，最终返回相似度最高的N个用户信息。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/user/similar/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 指定用户组ID，多个使用逗号隔开. |
| image | byte[] | 是 | 图片的二进制字节流，图片大小不得超过2MB |
| showTopNum | Int32 | 否 | 指定返回相似度分值最高的多少个用户信息.默认为1 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回数据数组的长度.数组长度等于上传图片中的人脸数 |
| result | Data[] | 返回结果.如果上传图片有2张人脸，那么该数组长度为2。每个数组元素也是一个数组，表示对应上传图片中人脸对应的识别结果分值从高到低排序。 |
| +groupId | String | 用户组ID |
| +uid | String | 用户ID |
| +userInfo | String | 用户信息 |
| +faceUrl | String | 人脸图片地址 |
| +score | Float | 相似度分值 |
|  |  |  |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 2,  "result": [  [  {  "groupId": "G001",  "uid": "U01010",  "userInfo": "JIM",  "faceUrl": "http://image-ai.ubtrobot.com/3054.jpg",  "score": 0.789456  },  {  "groupId": "G001",  "uid": "U01010",  "userInfo": "JIM",  "faceUrl": "http://image-ai.ubtrobot.com/3074.jpg",  "score": 0.711348  }  ],  [  {  "groupId": "G002",  "uid": "U01011",  "userInfo": "JACK",  "faceUrl": "http://image-ai.ubtrobot.com/9500.jpg",  "score": 0.889456  },  {  "groupId": "G002",  "uid": "U01011",  "userInfo": "JACK",  "faceUrl": "http://image-ai.ubtrobot.com/9501.jpg",  "score": 0.816048  }  ]  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

## 4.13人脸认证

用于识别上传的图片是否为指定用户，即查找前需要先确定要查找的用户在人脸库中的id。

一般应用场景，人脸登录，人脸签到等等。

说明：

人脸认证与人脸识别不一样，人脸认证是上传图片后与指定用户组(groupId)下的用户(uid)做比对，并返回比对结果，不需要遍历人脸库。人脸识别则需要遍历用户组下面的所有注册用户做比对然后返回相似度分值最高的那个用户信息。两者在后台处理逻辑上不一样。

使用注意：上传图片只支持一张人脸的情况。如果图片中存在多张人脸，只取其中一张人脸做认证。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/face/faceset/user/verify/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| groupId | String | 是 | 指定用户组ID.不支持指定多个。 |
| uid | String | 是 | 用户ID |
| image | byte[] | 是 | 图片的二进制字节流，图片大小不得超过2MB |
| topNum | Int32 | 否 | 指定返回多少个认证结果(分值从高到底排序).默认为1 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回数据数组的长度. |
| result | Data[] | 返回结果 |
| +groupId | String | 用户组ID |
| +uid | String | 用户ID |
| +userInfo | String | 用户信息 |
| +faceUrl | String | 人脸图片地址 |
| +score | Float | 相似度分值 |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 2,  "result": [  {  "groupId": "G002",  "uid": "U01011",  "userInfo": "JACK",  "faceUrl": "http://image-ai.ubtrobot.com/9500.jpg",  "score": 0.889456  },  {  "groupId": "G002",  "uid": "U01011",  "userInfo": "JACK",  "faceUrl": "http://image-ai.ubtrobot.com/9501.jpg",  "score": 0.816048  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

# 七.物体识别相关API接口

## 5.1物体检测

物体检测接口为用户上传图片后，能够检测出图片中相关物品的坐标位置，相似度以及名称。能检测出来的物品必须写通过算法实现，目前已经实现检测出81类物品，后续还会实现更多的物品品类的检测。

物体检测返回结果中的相似度分值为浮点数，1为满分。分值越高表明检测的物品越精准。

**请求说明**

****请求示例****

HTTP方法：**POST**

请求URL：**[https://open-api.ubtrobot.com](https://open-api.ubtrobot.com/oauth/2.0/token)/object/detect/v1**

URL参数：

|  |  |
| --- | --- |
| **参数** | **说明** |
| accessToken | 通过api\_key和secret\_key获取的accessToken.  参见 **Access Token获取** |

Header如下：

| **参数** | **值** |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Request Body参数详情如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必须** | **说明** |
| image | byte[] | 是 | 图片的二进制字节流，图片大小不得超过10MB |
| language | String | 否 | 指定物品名称的语言。中文(zh)或英文(en)。默认为zh。 |

**响应说明**

****返回参数如下：****

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| resultCode | Int32 | 响应状态码 |
| message | String | 响应内容描述 |
| resultNum | Int32 | 返回数据数组的长度,表示图片中有多少张人脸 |
| result | Data[] | 返回结果 |
| +x | Int32 | x坐标 |
| +y | Int32 | y坐标 |
| +height | Int32 | 高度 |
| +width | Int32 | 宽度 |
| +score | Float | 相似度 |
| +name | String | 物品名称 |

**示例**

|  |
| --- |
| {  "resultCode": 200,  "message": "请求成功",  "resultNum": 3,  "result": [  {  "x": 196,  "y": 342,  "height": 107,  "width": 91,  "score": 0.91354,  "name": "杯子"  },  {  "x": 392,  "y": 366,  "height": 50,  "width": 60,  "score": 0.78454,  "name": "蛋糕"  },  {  "x": 196,  "y": 342,  "height": 107,  "width": 91,  "score": 0.891545,  "name": "桌子"  }  ]  } |

**操作成功返回状态码为200.**

**如果出现异常情况，具体的状态码说明请参见“状态码说明表格”**

附：目前算法实现的81类能被检测的物品列表。

左边是ID，右边是名称

0-background

1-person

2-bicycle

3-car

4-motorcycle

5-airplane

6-bus

7-train

8-truck

9-boat

10-traffic light

11-fire hydrant

12-stop\_sign

13-parking meter

14-bench

15-bird

16-cat

17-dog

18-horse

19-sheep

20-cow

21-elephant

22-bear

23-zebra

24-giraffe

25-backpack

26-umbrella

27-handbag

28-tie

29-suitcase

30-frisbee

31-skis

32-snowboard

33-sports ball

34-kite

35-baseball bat

36-baseball glove

37-skateboard

38-surfboard

39-tennis racket

40-bottle

41-wine glass

42-cup

43-fork

44-knife

45-spoon

46-bowl

47-banana

48-apple

49-sandwich

50-orange

51-broccoli

52-carrot

53-hot dog

54-pizza

55-donut

56-cake

57-chair

58-couch

59-potted plant

60-bed

61-dining table

62-toilet

63-tv

64-laptop

65-mouse

66-remote

67-keyboard

68-cell phone

69-microwave

70-oven

71-toaster

72-sink

73-refrigerator

74-book

75-clock

76-vase

77-scissors

78-teddy bear

79-hair drier

80-toothbrush

# 八.响应状态码

## 状态码说明表格

| **状态码** | **消息说明** | **消息描述** |
| --- | --- | --- |
| 200 | Request success | 请求成功 |
| 500 | System error | 系统错误 |
| 700 | No accessToken | 没有传递accessToken |
| 701 | accessToken invalid | 无效的accessToken |
| 702 | accessToken expire | accessToken已过期 |
| 1301 | Open api daily request limit reached | 每天流量超限 |
| 1302 | Open api qps request limit reached | QPS流量超限 |
| 1304 | Invalid param | 参数无效 |
| 1305 | Not enough | 缺少必须的参数 |
| 1311 | param is longer than limit | 请求参数的长度超出限制 |
| 1306 | invalid api\_key and secret\_key | 非法的api\_key和secret\_key |
| 1307 | image size is bigger than limit | 上传的图片大小超出最大大小限制 |
| 1308 | the file is not an image | 上传的文件不是一张图片 |
| 1309 | there is no face in the image | 图片中不存在人脸 |
| 1310 | there is no this user | 不存在该用户 |
| 1311 | No group exists | 没存在该用户组 |