

汉王云 API 使用指南

一、调用 API 的通用步骤

1. 您从服务商处获得域名、Path、参数说明等信息，您创建的 APP 作为您的请求身份，AppKey 和 AppSecret 用于计算加密签名，签名的计算 demo 链接在下方。

确定您的 APP 获得了授权后，您就可以调用 API 了。API 的请求结构如下：

```
//假如，服务地址： aliapi.hanvon.com
```

```
//假如，Path： /rt/ws/v1/ocr/bcard/recg
```

```
//假如，Query 内容是： key=163114c8-31b5-4424-bb85-617f81cf54d9,  
code=cf22e3bb-d41c-47e0-aa44-a92984f5829d
```

```
//则请求的 url 如下
```

```
http:// aliapi.hanvon.com/rt/ws/v1/ocr/bcard/recg?  
key=163114c8-31b5-4424-bb85-617f81cf54d9&code=cf22e3bb-d41c-47e0-aa44-a92984f58  
29d
```

```
//请求的 Method
```

```
POST HttpMethod: POST
```

```
//Headers 中需要包含签名信息和部分参数
```

```
X-Ca-Version: 1 //API 版本
```

```
X-Ca-Signature-Headers: X-Ca-Version, X-Ca-Key, X-Ca-Stage, X-Ca-Timestamp
```

```
//参与签名的 Header
```

```
X-Ca-Key: 60028305 //AppKey
```

```
X-Ca-Stage: test //Stage
```

```
X-Ca-Timestamp: 1456905123049 //时间戳
```

```
X-Ca-Signature: UAaH/qteir4G9UK4YR+NWdyq+c1rj10Pvt0/C1Qo68U= //签名
```

```
//Http 标准头
```

```
Host: a13db7999e494a90819cce500130034d.com //服务地址
```

```
Date: Wed 02 Mar 2016 07:52:02 GMT
```

```
User-Agent: Apache-HttpClient/4.1.2 (java 1.6)
```

```
Content-Type: application/octet-stream //此项必填
```

```
//Body 内容
```

```
{"uid": "118.12.0.12", "lang": "auto", "color": "gray", "image": "您要识别的图像的base64字符串"}
```

2. 系统级 Header 参数说明

- **【必选】X-Ca-Key:** AppKey
- **【必选】X-Ca-Signature:** 签名字符串
- **【可选】X-Ca-Timestamp:** API 调用者传递时间戳，为时间转换为毫秒的值，也就是从 1970 年 1 月 1 日起至今的时间转换为毫秒，可选，默认 15 分钟内有效
- **【可选】X-Ca-Nonce:** API 调用者生成 UUID，结合时间戳进行防重放，可选
- **【可选】Content-MD5:** 当请求 Body 非 Form 表单时，可以计算 Body 的 MD5 值传递给云网关进行 Body MD5 校验
- **【可选】X-Ca-Stage:** 请求 API 所属 Stage，目前仅支持 test 和 release，默认 release。**若您调用的 API 不在线上环境，请一定要指定该参数的值，否则会报 URL 错误。**

3. 签名校验

签名计算的详细 demo (JAVA) 请参照链接:

<https://github.com/aliyun/api-gateway-demo-sign-java>。

签名计算方法步骤如下:

组织参与签名计算的字符串

```
String stringToSign=  
  
HTTPMethod + "\n" +  
  
Accept + "\n" +  
  
Content-MD5 + "\n"  
  
Content-Type + "\n" +  
  
Date + "\n" +  
  
Headers +  
  
Url
```

说明

HTTPMethod 为全大写

Accept、Content-MD5、Content-Type、Date 如果为空也需要添加换行符 "\n"，Headers 如果为空不需要添加 "\n"，不为空的 Headers 中包含了 "\n"，详见下面组织 Headers 的描述

Content-MD5

Content-MD5 是指 Body 的 MD5 值，只有当 Body 非 Form 表单时才计算 MD5，计算方式为：

```
String content-MD5 = Base64.encodeBase64(MD5(bodyStream.getBytes("UTF-8")));
```

bodyStream 为字节数组

Headers

Headers 指所有参与签名计算的 Header 的 Key、Value，这里要注意参与签名计算的 Header 是不包含 X-Ca-Signature、X-Ca-Signature-Headers 的。

Headers 组织方法：

先对所有参与签名计算的 Header 的 Key 按照字典排序，然后按照如下方式拼接：

```
String headers =  
  
HeaderKey1 + ":" + HeaderValue1 + "\n"+  
  
HeaderKey2 + ":" + HeaderValue2 + "\n"+  
  
...  
  
HeaderKeyN + ":" + HeaderValueN + "\n"
```

Url

Url 指 Path + Query + Body 中 Form 参数，组织方法：

对 Query+Form 参数按照字典对 Key 进行排序后按照如下方法拼接，如果 Query 或 Form 参数为空，则 Url = Path，不需要添加“？”

```
String url =  
  
Path +  
  
"?" +  
  
Key1 + "=" + Value1 +  
  
"&" + Key2 + "=" + Value2 +  
  
...  
  
"&" + KeyN + "=" + ValueN
```

注意这里 Query 或 Form 参数的 Value 可能有多个，多个的时候只取第一个 Value 参与签名计算

计算签名

```
Mac hmacSha256 = Mac.getInstance("HmacSHA256");

byte[] keyBytes = secret.getBytes("UTF-8");

hmacSha256.init(new SecretKeySpec(keyBytes, 0, keyBytes.length,
    "HmacSHA256"));

String sign = new
String(Base64.encodeBase64(Sha256.doFinal(stringToSign.getBytes("UTF-8")),
    UTF-8));
```

secret 为 APP 对应的密钥

传递签名

将计算的签名放到 Request 的 Header 中，Key 为：X-Ca-Signature

将所有参与加签的 Header 的 Key 使用英文逗号分割放到 Request 的 Header 中，不区分顺序，Key 为：X-Ca-Signature-Headers

签名计算的详细 demo 请参照链接：

<https://github.com/aliyun/api-gateway-demo-sign-java>。

二、各个 API 的详细调用

1. API 调用时，需要的 key 需由商家提供
2. 各个 API 调用是需要的 code 及请求参数、返回参数

2.1 名片

接口名称：businessCard

Path：/rt/ws/v1/ocr/bcard/recg

Code：多语言带坐标：cf22e3bb-d41c-47e0-aa44-a92984f5829d

简体中文：91f6a58d-e418-4e58-8ec2-61b583c55ba2

繁体中文：e6a41101-e7aa-4f7a-a8c7-719bad73a564

英文: a8ce6925-d91b-461b-bf18-99ae3663b23b

请求参数示例: {"uid": "118.12.0.12", "lang": "chns", "color": "gray", "image": "您要识别的图片的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 名片图像的 base64 串

color gray: 进行了灰度处理 ; color: 原图像未做任何处理

lang 多语言带坐标: auto; 中文简体: chns; 中文繁体: chnt; 英文: en

返回值说明:

code	response 返回码
result	response 消息内容
name	response 姓名
title	response 头衔
mobile	response 手机
tel	response 电话
degree	response 学历
dept	response 部门
comp	response 公司
web	response 网址
post	response 邮编
addr	response 地址
fax	response 传真
other	response 其他文字
numOther	response 其他数字
extTel	response 分机号
mbox	response 信箱
htel	response 家庭电话
email	response 邮箱
im	response 即时通讯号

返回示例: {
 "code": "0",
 "result": "48-5B-39-7E-6B-94",
 "rotatedAngle": "0.0",
 "name":

```
[
  "张三",
  "466",
  "224",
  "796",
  "313"
],
"title":
[
  "总经理",
  "549",
  "362",
  "708",
  "402"
],
"tel":
[
  "07448288139744",
  "374",
  "460",
  "854",
  "485"
],
"mobile":
[
],
"fax":
[
  "07448281",
  "372",
  "497",
  "671",
  "522"
],
"email":
[
  "zhangsan.nfan@zjj.wly.gov.cn",
  "373",
  "537",
  "877",
  "564"
],
"comp":
[
```

```
    "汉王科技",
    "345",
    "57",
    "911",
    "101",
    "Zhangwang Touristry Information Co.,
    Ltd",
    "347",
    "113",
    "907",
    "138"
  ],
  "dept":
  [
  ],
  "degree":
  [
  ],
  "addr":
  [
    "张家界市紫舞路 738 号",
    "372",
    "572",
    "757",
    "598"
  ],
  "post":
  [
  ],
  "mbox":
  [
  ],
  "htel":
  [
  ],
  "web":
  [
  ],
  "im":
  [
  ],
  "numOther":
  [
  ],
```

```
    "other":  
    [  
    ],  
    "extTel":  
    [  
    ]  
}
```

每项后面的四位数为坐标。

2.2 云文本商务版

2.2.1 返回文本

接口名称: table_text

Path: /rt/ws/v1/ocr/table/text/recg

Code: 0d3b7d23-915a-4c6f-9886-6312440aba51

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的base64字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回示例: {"code": "0", "result": "表头 1 表头 2

	第一列	第二列
第一行	11	12
第二行	21	22

表尾

"}

返回参数说明:

code response 返回码

result response 识别出的文本内容

2.2.2 返回 json

接口名称: table_json

Path: /rt/ws/v1/ocr/table/json/recg

Code: 95738911-8f66-49af-813d-75e9fe4b1ec2

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回参数说明:

code	response	返回码
result	response	返回码所对应的信息
block_count	response	文本块的总数
table_count	response	表格的总数
block_list	response	文本块的集合
table_list	response	表格的集合

block_list 结构如下

order	block_list	在全篇文本的块顺序号
line_list	block_list	一个文本块里行的集合
line_num	line_list	一个文本块的行号
text	line_list	识别出来的文字内容, 第一候选

table_list 结构如下

order	table_list	一个表格在全篇文本的块顺序号
tr_count	table_list	一个表格行数
td_count	table_list	一个表格列数
table	table_list	表示一个表格
tr	table	一个表格里的一行
td	tr	某一行里的一列
colspan	td	单元格可横跨的列数
rowspan	td	单元格可纵跨的行数
line_list	td	一个单元格里行的集合
line_num	line_list	一个文本块的行号
text	line_list	识别出来的文字内容, 第一候选

2.2.3 返回增强版 json

接口名称: table_enhancedjson

Path: /rt/ws/v1/ocr/table/enhancedjson/recg

Code: 540ab4a3-fa83-4fe2-b188-3edbac684d3f

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回参数说明:

code	response	返回码
result	response	返回码所对应的信息
block_count	response	文本块的总数
table_count	response	表格的总数
block_list	response	文本块的集合
table_list	response	表格的集合

block_list 结构如下

order	block_list	在全篇文本的块顺序号
line_list	block_list	一个文本块里行的集合
line_num	line_list	一个文本块的行号
text	line_list	识别出来的文字内容, 第一候选
confidence	line_list	每个文字的置信度, 从 A 到 E, 置信度一次降低
pos	line_list	每个文字的位置, 文字间用 分隔, 位置之间用-分隔

table_list 结构如下

order	table_list	一个表格在全篇文本的块顺序号
tr_count	table_list	一个表格行数
td_count	table_list	一个表格列数
table	table_list	表示一个表格
tr	table	一个表格里的一行
td	tr	某一行里的一列
colspan	td	单元格可横跨的列数
rowspan	td	单元格可纵跨的行数
line_list	td	一个单元格里行的集合

line_num	line_list	一个文本块的行号
text	line_list	识别出来的文字内容，第一候选
confidence	line_list	每个文字的置信度, 从 A 到 E, 置信度一次降低
pos	line_list	每个文字的位置, 文字间用 分隔, 位置之间用-分隔

2.3 银行卡

2.3.1 普通版

接口名称: bankCard

Path: /rt/ws/v1/ocr/bankcard/recg

Code: d72b071d-f407-42c9-b50c-8122bc4d5fc8

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回值示例:

```
{
  "code": "0",
  "result": "48-5B-39-7E-6B-94",
  "bankCard": {
    "bankname": "中国农业银行股份有限公司(103100000026)",
    "cardname": "借记卡(银联卡)",
    "cardtype": "借记卡",
    "cardno": "95599 8396 80856 48017",
    "expmonth": "0",
    "expyear": "0",
    "cropped_image": null
  }
}
```

返回参数说明:

code response 返回码

result	response 消息内容
bankCard	response 银行卡
bankname	bankCard 银行名称
cardname	bankCard 卡名
cardtype	bankCard 卡类型
cardno	bankCard 卡号
expmonth	bankCard 有效期截止月份
expyear	bankCard 有效期截止年份
cropped_image	bankCard 切割图的 base64 串 (仅切割版有值)

2.3.2 切图版

接口名称: bankCard_cropped

Path: /rt/ws/v1/ocr/bankcard/cropped/recg

Code: e954b425-6380-4946-83ae-9cc44f7d8331

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回值同 2.3.1。

2.4 行驶证

2.4.1 普通版

接口名称: vehicleCard

Path: /rt/ws/v1/ocr/vehiclecard/recg

Code: 6e8d31e0-f55b-4c22-8bcb-1211fd42596d

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回值示例:

```
{
  "code": "0",
  "result": "48-5B-39-7E-6B-94",
  "vehicleCard":
  {
    "plateno": "陕 K0076K",
    "vehicletype": "小型轿车",
    "veaddress": "北京市海淀区东北旺西路 8 号中关村软件园 5 号楼汉王大厦",
    "usecharacter": "非营运",
    "engineno": "166495",
    "model": "天籁牌 EQ7230BA",
    "vin": "LGBF1CM16R174349",
    "registerdate": "2006-08-01",
    "issuedate": "20120810",
    "cropped_image": null
  }
}
```

返回值说明:

code	response	返回码
result	response	消息内容
vehicleCard	response	行驶证
plateno	vehicleCard	车牌号码
vehicletype	vehicleCard	车辆类型
owner	vehicleCard	所有人
veaddress	vehicleCard	住址
usecharacter	vehicleCard	使用性质
engineno	vehicleCard	发动机号码
model	vehicleCard	品牌型号
vin	vehicleCard	车辆识别代码
registerdate	vehicleCard	注册日期
issuedate	vehicleCard	发证日期
cropped_image	vehicleCard	切割图(仅切割版有值)

2.4.2 切图版

接口名称: vehicleCard_cropped

Path: /rt/ws/v1/ocr/vehiclecard/cropped/recg

Code: 2bdab410-71b7-4f6d-9b91-8db27c9197e4

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回值及返回值说明同 2.4.1。

2.5 驾驶证

2.5.1 普通版

接口名称: driverCard

Path: /rt/ws/v1/ocr/drivercard/recg

Code: 8d45c31f-59d2-4004-907e-7eff134017df

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回值示例:

```
{
  "code": "0",
  "result": "48-5B-39-7E-6B-94",
  "driverCard":
  {
    "name": "汉王",
    "gender": "男",
    "nation": "中国",
    "cardno": "430602198503064527",
    "address": "北京市海淀区东北旺西路 8 号中关村软件园 5 号楼汉王大厦",
    "birthdate": "1985-03-06",
    "issuedate": "2007-09-25",
    "driverclass": "C1",
    "validdate": "2007-09-26 年",
    "cropped_image": ""
  }
}
```

```
}  
}
```

返回值说明:

code	response	返回码
result	response	消息内容
driverCard	response	驾驶证
name	driverCard	姓名
gender	driverCard	性别
nation	driverCard	国籍
cardno	driverCard	证号
address	driverCard	住址
birthdate	driverCard	出生日期
issuedate	driverCard	初次领证日期
driverclass	driverCard	准驾车型
validdate	driverCard	有效期限
cropped_image	driverCard	切割图(仅切割版有值)

2.5.2 切图版

接口名称: driverCard_cropped

Path: /rt/ws/v1/ocr/drivercard/cropped/recg

Code: ea22962a-47c9-4cb6-9566-286a926736bc

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串

返回值示例及返回值说明同 2.5.1。

2.6 身份证

2.6.1 普通版

接口名称: idCard

Path: /rt/ws/v1/ocr/idcard/recg

Code: 8d497db3-7341-4f1f-875a-2f5444884515

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串, 图片要求为真彩色, 不要进行灰度处理

返回值示例:

```
{
  "code": "0",
  "result": "",
  "name": "韦小宝",
  "gender": "男",
  "people": "土",
  "byear": "1651",
  "bmonth": "12",
  "bday": "1",
  "address": "北京市东召山",
  "idnumber": "112814165112202111",
  "authority": "",
  "validdate": "",
  "left": 211,
  "top": 25,
  "right": 312,
  "bottom": 169,
  "croppedImage": "",
  "headImage": ""
}
```

返回值说明:

code	response code
result	response result
name	response 姓名
gender	response 性别
people	response 民族
byear	response 出生年
bmonth	response 出生月
bday	response 出生日
address	response 住址

idnumber response 公民身份证号
authority response 签发机关
validdate response 有效期限
left response 身份证头像左边距
top response 身份证头像上边距
right response 身份证头像右边距
bottom response 身份证头像下边距
croppedImage response 身份证切割图的 base64 串 (仅切图版有值)
headImage response 身份证头像图片的 base64 串 (仅切图版有值)

2.6.2 切图版

接口名称: idCard_cropped

Path: /rt/ws/v1/ocr/idcard/cropped/recg

Code: 56bc30ea-2229-4891-9c92-8a022f10889f

请求参数示例: {"uid": "118.12.0.12", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串, 图片要求为真彩色, 不要进行灰度处理

返回值示例及返回值说明同 2.6.1。

2.7 题目

接口名称: title

Path: /rt/ws/v1/ocr/formula/recg

Code: fb177e48-fc6b-4d43-9a60-e4487db4cd20

请求参数示例: {"uid": "118.12.0.12", "color": "gray", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

color gray: 进行了灰度处理 ; original: 原图像未做任何处理 上传图像的类型, 建议进行灰度处理

image 题目图像的 base64 串，要求上传前进行灰度处理

返回值说明：

code	response	返回码
result	response	消息内容
blocks	response	返回结果列表
type	blocks	类型 1：公式 2：文字
formulaResult	blocks	与类型对应的识别结果
rect	blocks	识别结果所对应的矩形区域
left	rect	矩形左上角 x 轴坐标
top	rect	矩形左上角 y 轴坐标
right	rect	矩形右下角 x 轴坐标
bottom	rect	矩形右下角 y 轴坐标

2.8 数字

接口名称：number

Path：/rt/ws/v1/ocr/number/recg

Code：3d901313-5b21-4c7c-9fc2-5dbcddef0d42

请求参数示例：{"uid": "118.12.0.12", "color": "gray", "image": "您要识别的图像的 base64 字符串"}

请求参数说明：

uid 您的服务器 IP 地址。

color gray：进行了灰度处理；original：原图像未做任何处理 上传图像的类型，建议进行灰度处理

image 图像的 base64 串，要求上传前进行灰度处理

返回值说明：

code	response	返回码
result	response	消息内容
textResult	response	数字内容

返回值示例：

```
{
  "code": "0",
  "result": "48-5B-39-7E-6B-94",
  "textResult": "12345678"
}
```

2.9 文本

接口名称: text

Path: /rt/ws/v1/ocr/text/recg

Code: 简体中文: 74e51a88-41ec-413e-b162-bd031fe0407e

繁体中文: 4060d49a-acf8-4897-9f61-5540bda01ed9

英文: fe8b94d8-f34a-4804-abf0-4d1d072797cf

请求参数示例: {"uid": "118.12.0.12", "lang": "chns", "color": "gray", "image": "您要识别的图片的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

image 图像的 base64 串, 要求上传前进行灰度处理

color gray: 进行了灰度处理 ; color: 原图像未做任何处理 上传图像的类型, 建议进行灰度处理

lang 中文简体: chns; 中文繁体: chnt; 英文: en

返回值说明:

code response 返回码

result response 消息内容

textResult response 文本内容

返回值示例:

```
{
  "code": "0",
  "result": "48-5B-39-7E-6B-94",
  "textResult": "相同的产品各地都有。了解这一点, 才会明白为什么早些年各地会同时引进 100 多条电视机生产线"
}
```

2.10 手写单字

接口名称: handWord

Path: /rt/ws/v1/hand/single

Code: 简体中文: 83b798e7-cd10-4ce3-bd56-7b9e66ace93d

繁体中文: 961e5736-ccb-437b-8cd2-d793b0a8f016

请求参数示例: {"uid": "118.12.0.12", "type": "1", "data": "76.55, 79.55, 51.7, 119.35, 43.75, 129.3, -1, 0"}

请求参数说明:

uid 您的服务器 IP 地址。

type 请求类型区分: 1. 手写轨迹识别 2. 获取联想字

当 type=1 时, 请传入手写轨迹串: x1, y1, x2, y2, x3, y3, x4, y4, -1, 0 轨迹以:
data -1, 0 结尾。

当 type=2 时, 请传入一个 unicode 码, 将返回这个 unicode 码对应联想的字符。

返回值说明:

code response 0, 表示成功。其它表示失败。

result response 如果成功, 则返回识别后的 unicode 码, 多个码之间用", " 分隔。

返回值示例:

```
{
  "code": "0",
  "result": "9968, 20032, 20843, 20154, 24515, 21340, 19979, 20033, 20837, 24028,"
}
```

2.11 手写行叠写

接口名称: handLine

Path: /rt/ws/v1/hand/line

Code: 简体中文: d4b92957-78ed-4c52-a004-ac3928b054b5

繁体中文: 05a7d172-ad21-4749-be0f-bfa4166d4da0

英文: f01d64a2-bd96-4554-8bcc-81d221f314a4

请求参数示例: {"uid": "118.12.0.12", "lang": "chns", "data": "x1, y1, x2, y2, -1, 0, x3, y3, x4, y4, -1, 0, -1, -1"}

请求参数说明:

uid 您的服务器 IP 地址。

lang 中文简体: chns; 中文繁体: chnt; 英文: en

手写行轨迹串。格式是: $x_1, y_1, x_2, y_2, -1, 0, x_3, y_3, x_4, y_4, -1, 0, -1, -1$

data 每一笔写完以: $-1, 0$ 结尾。

全部写完以: $-1, -1$ 结尾。

返回值说明:

code response 0, 表示成功。其它表示失败。

result response 如果成功, 则返回识别后的 unicode 码, 多个码之间用", " 分隔。

返回值示例:

```
{
  "code": "0",
  "result":
"25105, 123, 105, 0, 25105, 104, 58, 0, 25105, 104, 63, 0, 25105, 105, 58, 0, 25105, 123, 58, 0, 25105, 105, 63, 0, 25105, 123, 63, 0, "
}
```

2.12 手写公式

接口名称: handFormula

Path: /rt/ws/v1/formula/hand/recg

Code: 14b05be6-a9a5-45bc-980b-afc2e9847742

请求参数示例: {"uid": "118.12.0.12", "data": "x1, y1, x2, y2, -1, 0, x3, y3, x4, y4, -1, 0, -1, -1"}

请求参数说明:

uid 您的服务器 IP 地址。

手写轨迹串: $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, -1, 0, -1, -1$

data 每一笔写完以: $-1, 0$ 结尾。

全部写完以: $-1, -1$ 结尾。

返回值说明:

code response 0, 表示成功。其它表示失败。

result response 消息内容

formulas response 识别结果的数组集合 (最多六个结果)

返回值示例:

```
{  
  "code": "0",  
  "result": "",  
  "formulas":  
  [  
    "f ( x ) = \\sqrt{3 + x}",  
    "\\int ( x ) = \\sqrt{3 + x}",  
    "f ( x ) = \\sqrt{3 + \\times }",  
    "f ( x ) = \\sqrt{3 + x}",  
    "f ( x ) = \\sqrt{3 + x}",  
    "f ( x ) = \\sqrt{3 + x}"  
  ]  
}
```

2.13 人脸关键点定位

接口名称: face_kpl

Path: /rt/ws/v1/face/kpl/recg

Code: 7815ff0e-aac8-4d74-8b7e-b190398cc649

请求参数示例: {"uid": "118.12.0.12", "lang": "chns", "color": "gray", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

data 人脸图像 base64 串, 要求上传前进行灰度处理

返回值说明:

code	response	返回码
result	response	消息内容
faces	response	人脸定位信息
keyPoints81	faces	81 点人脸定位信息
col	keyPoints81	定位点的横坐标
row	keyPoints81	定位点的纵坐标
keyPoints14	faces	14 点人脸定位信息
face	keyPoints14	脸的中心坐标
leftEye	keyPoints14	左眼中心坐标

rightEye keyPoints14 右眼中心坐标
leftUpperEye keyPoints14 左上眼坐标
rightUpperEye keyPoints14 右上眼坐标
leftleftEye keyPoints14 左眼左侧坐标
leftrightEye keyPoints14 左眼右侧坐标
rightleftEye keyPoints14 右眼左侧坐标
rightrightEye keyPoints14 右眼右侧坐标
leftNostril keyPoints14 左鼻孔
rightNostril keyPoints14 右鼻孔
nosePoint keyPoints14 鼻中心
leftMouth keyPoints14 左嘴角
rightMouth keyPoints14 右嘴角

2.14 人脸年龄性别表情

接口名称: face_age

Path: /rt/ws/v1/face/age/recg

Code: 2f873139-c6d0-462c-81d9-3623f96ecaa8

请求参数示例: {"uid": "118.12.0.12", "lang": "chns", "color": "gray", "image": "您要识别的图像的 base64 字符串"}

请求参数说明:

uid 您的服务器 IP 地址。

data 人脸图像 base64 串, 要求上传前进行灰度处理

返回值说明:

code response 返回码
result response 消息内容
age response 年龄
ageResult response 年龄性别结果
exp response 表情
expResult response 表情结果
gender response 性别
expscore response 表情可信度

返回值示例:

```
{  
  "code": "0",  
  "result": "123adikd",  
  "age": "25",  
  "ageResult": "0",  
  "expResult": "0",  
  "gender": "1",  
  "exp": "1",  
  "expscore": "79"  
}
```