

人脸对比接口 guahao. face. facematch

更新时间：2021-02-23 18:32:34

简介

[平台历史数据迁移]

请求地址

环境	HTTPS
测试环境	https://openapi.guahao-test.com/openapi
生产地址	https://openapi.guahao.com/openapi

访问规则

规则	描述
请求方式	POST
请求参数加密	无
响应数据解密	无

公共参数

Header

参数名称	参数类型	是否必填	描述
appkey	String	是	用户秘钥管理中的AccessKey ID，具体参见： 访问控制
method	String	是	平台接口名称，此接口为：guahao. face. facematch
timestamp	String	是	UNIX时间戳，单位毫秒，需要与平台服务器时间差值在2.5分钟内，如：1605868401000
version	String	是	版本，此接口为：2.0
product-code	String	是	产品编码，此接口为：1V1HYV30f
message-id	String	是	由调用方生成，请求唯一标识，长度不超过36位；相同message-id 2.5 分钟内只能被使用一次
sign	String	是	签名，详情见 签名介绍
content-type	String	是	application/json
content-md5	String	是	请求体为JSON且报文不为空时，对JSON报文体数据进行md5并转化为32为大写的字符串，用

			于加签时使用
--	--	--	--------

请求参数

参数描述

参数名称	参数类型	是否必填	描述
faceMatchRequestDTO	object	否	
- appVersion	string	否	应用版本
- business	string	否	业务
- appId	string	否	应用ID
- appKey	string	否	应用账号
- imageList	array	否	图片列表
faceMatchRequestDTO2	object	否	
- appVersion	string	否	应用版本
-- faceMatchRequestDTO2	object	否	
--- appVersion	string	否	应用版本
--- business	string	否	业务
--- appId	string	否	应用ID
--- appKey	string	否	应用账号
--- imageList	array	否	图片列表
- business	string	否	业务
- appId	string	否	应用ID
- appKey	string	否	应用账号
- imageList	array	否	图片列表

请求示例

```
curl --request POST 'http://openapi.guahao-test.com/openapi' \
--header 'Content-Type: application/json' \
--header 'appkey: AccessKey ID' \
--header 'method: guahao.face.facematch' \
--header 'sign: ' \
--header 'timestamp: 1617246004732' \
--header 'version: 2.0' \
--header 'product-code: 1V1HYV30f' \
--header 'message-id: f526b131-875e-46c8-9df9-e726ac13c60f' \
--header 'version: 2.0' \
--header 'content-md5: JSONJSONJSONmd532' \
--data-raw '{
  "faceMatchRequestDTO": {
    "appVersion": null,
    "business": null,
    "appId": null,
    "appKey": null,
```

```
        "imageList":null
    },
}
```

响应参数

参数描述

参数名称	参数类型	是否必填	描述
code	string	是	响应码
data	object	是	响应数据
- score	string	否	对比相似度
- redirectUrl	string	否	业务方回调地址
- livenessTicket	string	否	人脸比对票据，值与identityTicket值相同，有效期1小时
- livenessKanoId	string	否	活体图片
- authResult	int	否	认证结果，成功：0，失败：1，认证标准是大于等于分数阈值认证成功，小于分数阈值认真失败
- faceList	array	否	人脸比对结果
- faceKanoId	string	否	静态图片包括身份证图片，拍照头像
message	string	是	响应信息

响应示例

```
{
    "code": "0",
    "data": {
        "score": null,
        "redirectUrl": null,
        "livenessTicket": null,
        "livenessKanoId": null,
        "authResult": null,
        "faceList": null,
        "faceKanoId": null
    },
    "message": null
}
```

API工具

[java-demo代码](#) [php-demo代码](#)

签名介绍

加签主体

仅请求体加签，响应体不涉及加签。

加签步骤

1 构造原串

1.1 将公共参数（除sign外的非空参数） + HTTP Query 参数（即，URL后面拼接的参数或FORM表单字段）按照字母先后顺序排序得到：keyvaluekeyvalue...keyvalue即得到拼接字符串originalSignStr。

A. JSON报文数据（若JSON报文为空，则content-md5不参与加签构造）

- 首先对JSON请求体数据（JSON整体，若需要加密则为加密后的JSON报文）进行md5并转化为32位大写字符串
- 放置于请求头content-md5字段的value值中
- 以content-md5作为键参与键值对排序

如：请求头为 method="guahao.hospdept.search", appkey="123456"
JSON原始报文为 jsonBody={"name":"test","age":25,"department":"门诊","hospital":"同济医院"}
首先：将jsonBody进行md5并转化为32位大写字符串获得contentMd5Value: BEC73AAA277077898BCEA487365782FE
其次：将contentMd5Value置于请求头content-md5字段的value值中：content-md5="BEC73AAA277077898BCEA487365782FE"
此时请求头为：method="guahao.hospdept.search", appkey="123456", content-md5="BEC73AAA277077898BCEA487365782FE"
最后：以content-md5作为键参与键值对排序拼接，得到字符串源：
originalSignStr = appkey123456content-md5BEC73AAA277077898BCEA487365782FEmethodguahao.hospdept.search

B. FORM表单数据

如：请求头为 method="guahao.hospdept.search", appkey="123456"
FORM表单数据为 query="同济医院"
拼接得到字符串源为：
originalSignStr = appkey123456methodguahao.hospdept.searchquery同济医院

1.2 将appsecret字符串名称和秘钥appsecret值分别连接到拼装好的字符串源originalSignStr头部和尾部，获得待加签字符串targetSignStr（此步骤JSON报文数据和FORM表单数据均相同）

例如：appsecret="36488SDSXXXXXXXXXXXXXXXXXXXX"
待加签字符串即为：

A. JSON报文数据：
targetSignStr = appsecretappkey123456content-md5BEC73AAA277077898BCEA487365782FEmethodguahao.hospdept.search36488SDSXXXXXXXXXXXXXXXXXXXX

B. FORM表单数据：
targetSignStr = appsecretappkey123456methodguahao.hospdept.searchquery同济医院36488SDSXXXXXXXXXXXXXXXXXXXX

2 生成签名

将待加密字符串targetSignStr进行md5并转化为32位大写，即得到签名signStr

```
signStr = md5(targetSignStr).toUpperCase();
```

3 将signStr的值置于请求头的sign字段中，最终请求信息如下：

如：signStr=1D3363C18A543E0247AD8FF83E189C0C

A. JSON报文数据

请求头：method="guahao.hospdept.search", appkey="123456", content-md5="BEC73AAA277077898BCEA487365782FE", sign="1D3363C18A543E0247AD8FF83E189C0C"
JSON报文：{"name":"test","age":25,"department":"门诊","hospital":"同济医院"}

B. FORM表单数据

请求头：method="guahao.hospdept.search", appkey="123456", sign="1D3363C18A543E0247AD8FF83E189C0C"
FORM表单数据：query="同济医院"

平台异常码

网关级异常码示例

code	描述	备注
0	成功	
-1	失败	
-14	XXX:具体method:用户签约产品不包含该api	
202101	参数[XX]不能为空	
202116	JSON报文的content-md5校验失败	
202117	时间戳timestamp格式错误	
202118	消息标识message-id已失效	
202119	消息标识message-id格式错误，长度超过36位	
202120	服务路径无效或错误	
200002	应用凭证ID[XXX]无效或错误	
202104	服务接口名称[XX]无效或错误	
202106	签名版本[XXX]无效或不支持	
202110	请求体Body数据格式类型，Content-Type[XX]不支持！	
202112	请求失效，时间戳已经过期	
200051	无效签名	
200052	签名算法,digest[XX]不支持	
400001	请求源的ip未授权	
400002	系统配置错误	

开放平台级异常码示例

code	描述	备注
OPEN_202100_SYS	method为空	
OPEN_202101_SYS	method无效，请开发者注意检查！	

OPEN_402001_API	服务API参数名[XX]不能为空！	
OPEN_402002_API	服务API参数[XX=XX]数据格式错误！	
OPEN_402003_API	参数JSON解析异常！	
OPEN_403000_API	服务API配置出错，接口协议[XX]不支持！	
OPEN_403200_API	接口URL参数非法！	
OPEN_404000_ENV	API运行环境配置失效，请开发者注意检查！	
OPEN_600000_API	Return result is empty	
OPEN_601000_API	Remote connect timeout	
OPEN_602000_API	Remote http call failed:XXX	