

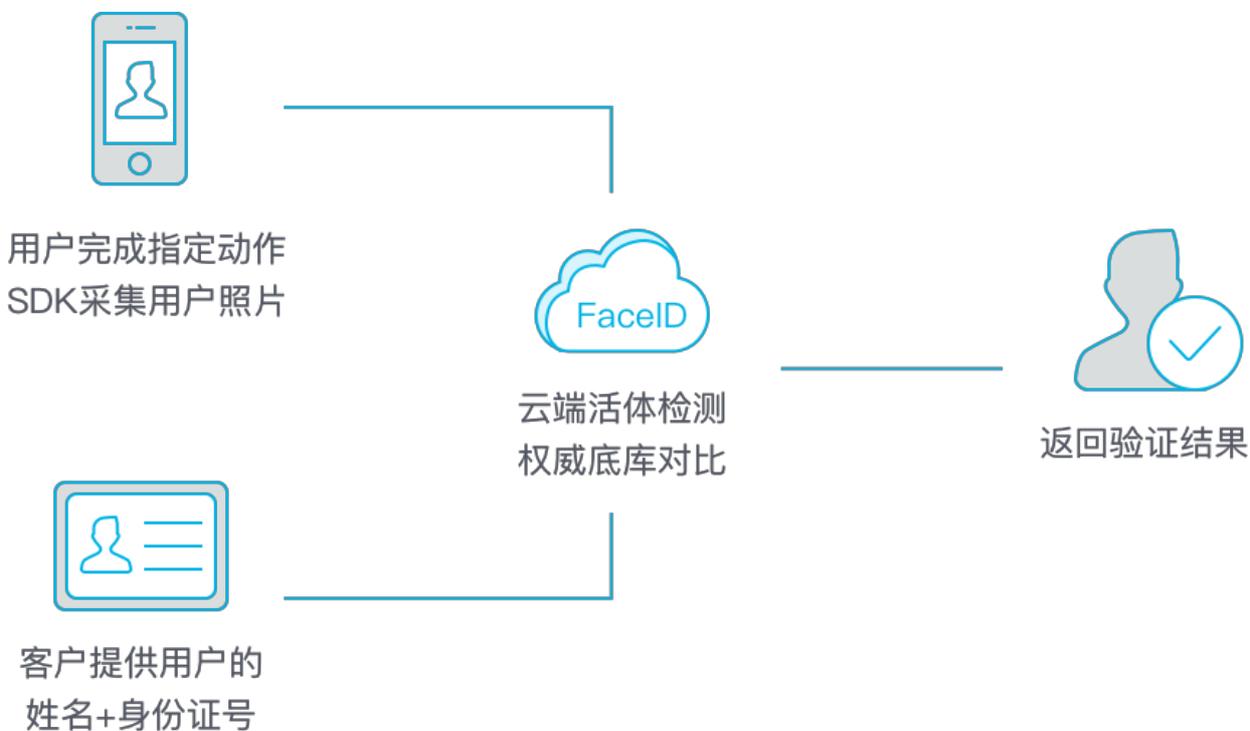
# FaceID人脸身份验证接入文档

## 简介

FaceID 人脸核身服务提供了“我是我”的真人核验能力，通过姓名、身份证号、人脸识别、动作判断等多种手段来有效的核实使用者的身份，有效防止身份冒用、欺诈等风险。

用户使用流程如下：

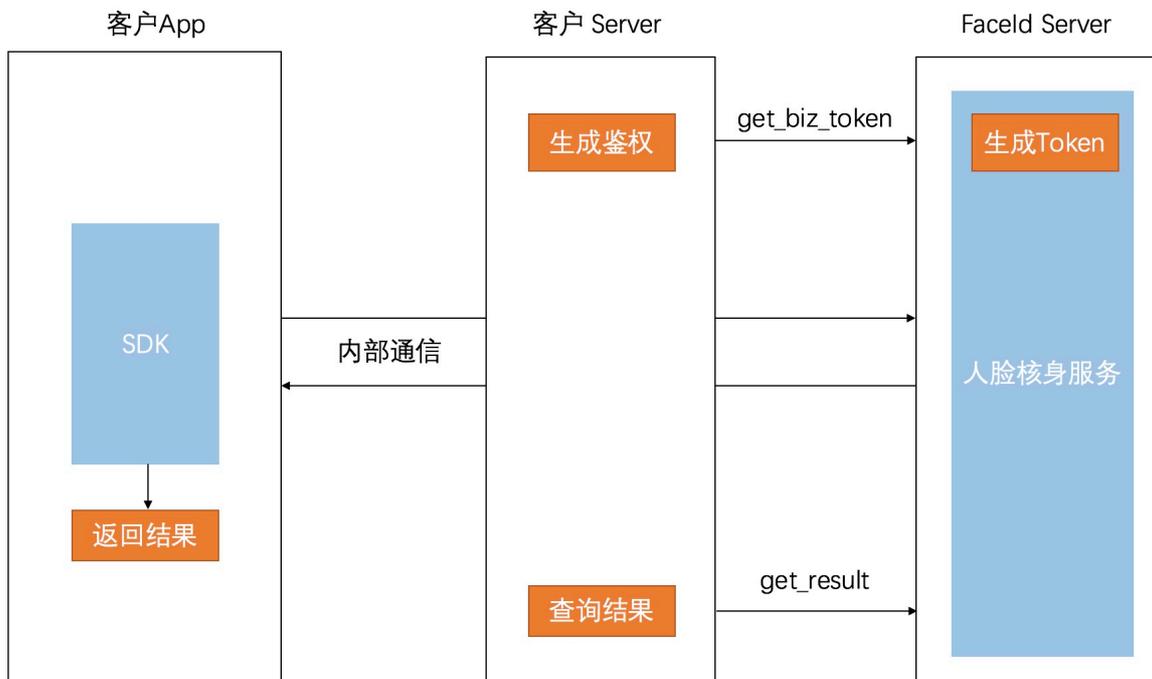
- 用户按照SDK UI页面的提示，进行点头，摇头，眨眼等动作活体验证。
- 动作通过后，SDK内部会调用FaceID服务器进行云端攻击检测，并开始人脸身份验证。
- 人脸身份验证的结果会返回给SDK，详细信息可以通过我们提供的API进行查询。



## 计费说明

- 按次收费，用户完成一次人脸身份验证才扣费(get\_result接口返回值中result\_code为1000/2000/3000时计费)，未完成的情况不扣费。

## 接入流程图



说明：

1. 通过get\_biz\_token接口获取token
2. 下载SDK，集成到客户的APP
3. (可选) 通过get\_result接口获取详情

- android SDK下载：[http://assets.faceid.com/faceidopen/sdk/FaceIDZFAC\\_1.2.1A.zip](http://assets.faceid.com/faceidopen/sdk/FaceIDZFAC_1.2.1A.zip)
- iOS SDK下载：[http://assets.faceid.com/faceidopen/sdk/FaceIDZFAC\\_1.2.1I.zip](http://assets.faceid.com/faceidopen/sdk/FaceIDZFAC_1.2.1I.zip)

下面进行详细阐述

## 第1步：使用get\_biz\_token获取biz\_token

### 调用地址

POST [https://fid.market.alicloudapi.com/face/v1.2/sdk/get\\_biz\\_token](https://fid.market.alicloudapi.com/face/v1.2/sdk/get_biz_token)

### 签名方式

参见阿里云文档：

[https://help.aliyun.com/document\\_detail/29490.html?spm=5176.730006-56956004-57124001-cmapi029634/A.content.10.O4CB1x](https://help.aliyun.com/document_detail/29490.html?spm=5176.730006-56956004-57124001-cmapi029634/A.content.10.O4CB1x)

### 参数

必选/可选	参数	类型	参数说明
必选	liveness_type	string	刷脸验证流程中的活体类型，当前只支持meglive(动作活体)；
必选	idcard_name	string	姓名
必选	idcard_number	string	身份证号
可选	image_ref1	string	由客户提供的参照人脸照片
可选	image_ref2	string	由客户提供的参照人脸照片
可选	biz_no	string	客户业务流水号，建议设置为您的业务相关的流水串号并且唯一。并会在 return 时原封不动的返回给您的服务器，以帮助您确认对应业务的归属。此字段不超过128字节
可选	liveness_timeout	int	超时时间，默认值为 60，表示用户进入活体识别流程后的超时时间，若未在规定时间内完成操作，则本次活体失败
可选	liveness_action_count	int	动作活体时动作个数：默认3个动作；默认 3， $\in [1,3]$
可选	security_level	int	表示对对比结果的严格程度限制，请根据您的场景，选择安全规则，越严格，准确性要求越高，通过率也会相应下降 1：宽松(误识率为千分之一)； 2：标准(误识率为万分之一，默认值)； 3：严格(误识率为十万分之一)； 4：非常严格(误识率为百万分之一)；
可选	force_compare	int	表示云端判断为假脸后，是否依然进行比对； 0:默认值，云端判断为假脸，则直接返回结果，不执行比对，可以节省成本； 1:云端判断为假脸后，依然进行比对
可选	multi_oriented_detection	int	对于image_ref1,image_ref2 参考照片，当检测不出人脸时，是否旋转90度、180度、270度后再检测人脸。 "1": 默认值，要旋转检测； "0": 不旋转； 请注意：设置此参数为1可能会轻微增加误检人脸的概率，如果您明确您的业务场景里不存在非正向的人脸图片、或概率极低，建议勿设置此参数。

## 返回值

参数	类型	参数说明
request_id	string	用于区分每一次请求的唯一的字符串。此字符串可以用于后续数据反查。此字段必定返回。
time_used	int	整个请求所花费的时间，单位为毫秒。此字段必定返回。
biz_token	string	调用 SDK 时传入的信息。请求成功时返回。
error	string	具体返回内容见错误码列表。请求失败时返回。

## 错误码列表

HTTP 状态码	错误信息	说明
400	MISSING_ARGUMENTS:<key>	缺少某个必选参数。
400	BAD_ARGUMENTS:<key>	某个参数解析出错（比如必须是数字，但是输入的是非数字字符串；或者长度过长）
400	IMAGE_ERROR_UNSUPPORTED_FORMAT:<param>	参数<param>对应的图像无法解析，有可能不是图像文件、或有数据破损。<param>为 image_ref1、image_ref2中的一个。请注意：<param>只会有一项，即第一个满足条件的名称。
400	NO_FACE_FOUND:<param>	表示上传的 image_ref 的图像中，没有检测到人脸。param为 image_ref1、image_ref2中的一个。请注意：<param>只会有一项，即第一个满足条件的名称。
400	INVALID_IMAGE_SIZE<param>	客户上传的图像太大，具体是指像素尺寸的长或宽超过4096像素。<param>为 image_ref1、image_ref2中的一个。请注意：<param>只会有一项，即第一个满足条件的名称。
403	AUTHENTICATION_ERROR	无效签名
403	AUTHORIZATION_ERROR:<reason>	调用次数超限、没有调用此API的权限，或者没有以当前方式调用此API的权限
403	CONCURRENCY_LIMIT_EXCEEDED	并发数超过限制
404	API_NOT_FOUND	所调用的API不存在
413	Request Entity Too Large	客户发送的请求大小超过了20MB限制。该错误的返回格式为纯文本，不是json格式。
500	INTERNAL_ERROR	服务器内部错误，当此类错误发生时请再次请求，如果持续出现此类错误，请及时联系 FaceID 客服或商务

## 第2步：集成SDK

### 2.1 Android SDK说明

#### 入口类

com.megvii.faceid.zzplatform.sdk.manager.FaceIdManager

## 获取单实例对象接口

```
public static synchronized FaceIdManager getInstance(Context context);
```

- 参数说明

参数	说明
Context	android.os.Context: 建议使用ApplicationContext

- 返回值说明

返回值	说明
FaceIdManager	入口类FaceIdManager的单实例对象

## 初始化接口

```
public void init( String bizToken);
```

- 调用说明  
必须在调用所有接口前，初始化成功，否则无法正常工作。
- 参数说明

参数	说明
bizToken	业务流水号

## 开始检测接口

```
public void startDetect();
```

- 参数说明

参数	说明
无	无

- 返回值说明

返回值	说明
void	无返回值

## 设置Init回调接口

```
public void setFaceIdInitListener(FaceIdInitListener listener);
```

- MegLiveDetectListener 说明

```
public interface FaceIdInitListener {  
    void onSuccess ();  
    void onFailed(int code, String msg);  
}
```

参数	说明
code	Init结果状态码, 参见SDK-错误码说明
message	Init结果信息, 参见SDK-错误码说明

## 设置Detect回调接口

```
public void setFaceIdDetectListener(FaceIdDetectListener listener);
```

- FaceIdDetectListener 说明

```
public interface FaceIdDetectListener {  
    void onSuccess(int code, String msg);  
    void onFailed(int code, String msg);  
}
```

参数	说明
code	startDetect结果状态码, 参见SDK-错误码说明
message	startDetect结果信息, 参见SDK-错误码说明

## 获取SDK版本号接口:

```
public String getSdkVersion();
```

- 参数说明

参数	说明
无	无

- 返回值说明

返回值	说明
String	SDK版本号

## 获取编译版本号接口：

```
public String getSdkBuildInfo();
```

- 参数说明

参数	说明
无	无

- 返回值说明

返回值	说明
String	SDK编译信息

## 2.2 Android 接入指南(Android Studio)

1. 把arr文件复制进入libs文件夹中
2. 在APP的gradle中加入：

```
repositories {
    flatDir{
        dirs 'libs'
    }
}
```

3. 在APP的gradle的dependencies中加入：

```
compile(name:'faceid_sdk',ext:'aar')
```

4. Rebuild Project

## 2.3 iOS SDK说明

### 类MGFaceIDDetectManager

### 函数initFaceIdManagerWithToken

函数名	<b>initFaceIdManagerWithToken</b>
方法描述	<code>- (instancetype)initFaceIdManagerWithToken:(NSString * __nonnull)token error:(FaceIDDetectError ** __nonnull)error;</code>
名称	初始化方法
说明	活体检测初始化时调此接口
变量名	说明
token	类型： NSString * 说明： 业务流水号
error	类型： FaceIDDetectError** 说明： 初始化错误类型，若没有错误，则该参数为空

## 函数startDetect

函数名	<b>startDetect</b>
方法描述	<code>- (void)startDetect:(UIViewController * __nonnull)detectVC callback:(FaceIDDetectBlock __nonnull)result;</code>
名称	启动FaceID活体检测
类型说明	函数
说明	启动FaceID活体检测
变量名	说明
detectVC	类型： UIViewController * 说明： 启动检测的VC
result	类型： FaceIDDetectBlock 说明： 检测结果的block回调。详情请参考(MGFaceIDDetectConfig.h)

- 注意：每次调用startDetect前，请先调用initFaceIdManagerWithToken接口

## 函数getSdkVersion

此函数是为非必须函数，获取的SDK版本号便于后期定位问题，建议使用；

函数名	<b>getSdkVersion</b>
方法描述	<code>+ (NSString * _Nonnull) getSdkVersion;</code>
名称	获取SDK的版本号
类型说明	函数
说明	无
返回值	NSString*, 一个字符串, 表示当前SDK的版本号。

## 函数getSdkBuildInfo

此函数为非必须函数, 获取的SDK构筑信息便于后期定位问题, 建议使用;

函数名	<b>getSdkBuildInfo</b>
方法描述	<code>+ (NSString * _Nonnull) getSdkBuildInfo;</code>
名称	获取SDK的构筑信息
类型说明	函数
说明	无
返回值	NSString*, 一个字符串。

## FaceIDDetectBlock回调函数

函数名	<b>FaceIDDetectBlock</b>
方法描述	Detect的回调函数, 返回相应的结果
名称	<code>typedef void(^ FaceIDDetectBlock)(NSInteger Code, NSString* Message);</code>
类型说明	函数
说明	无
变量名	说明
Code	类型: NSInteger 说明: 参见SDK-错误码说明
Message	类型: NSString* 说明: 参见SDK-错误码说明

## 2.4 iOS接入指南

# SDK介绍

SDK 版本：FaceIDZFAC 1.2.0I

## SDK 集成要求：

- SDK支持的系统最低要求是 `ios 8.0`，只支持真机架构 (`arm64和armv7`)，不支持模拟器架构 (`x86_64和i386`)。

## SDK 集成说明：

- 1、添加 SDK 库：将 `sdk` 文件夹中的 `MGFaceIDLiveDetect.framework` 添加到工程中，添加后确保 `TARGETS-Build Phases-Link Binary With Libraries` 中可以查找到 `MGFaceIDZZIDCardKit.framework(Required)`。如果该 `SDK(.framework)` 在非 `PROJECT` 根目录下，请确定在 `TARGETS-Build Settings-Framework Search Paths` 选项中添加了该 SDK 所在的目录路径。
- 2、因为该 `framework` 中使用了 `category`，需要在 `Other Linker Flags` 中添加 `-ObjC`；
- 3、添加系统依赖：在 `TARGETS-Build Phases-Link Binary With Libraries` 中添加 `AVFoundation.framework`、`CoreMedia.framework`；
- 4、添加资源：将 `resource` 文件夹中的资源包 (`MGFaceIDLiveDetectResouce.bundle`) 添加到工程中，需要勾选 `(Add to targets)` 到指定的 `target`；
- 5、因为该 SDK 中使用了 C++ 编码，需要工程支持混编 C++。设置 `TARGETS-Build Settings-Compile Sources As` 选项为 `Objective-C++` 或者将调用该 SDK 的文件后缀名称改为 `mm`；
- 6、添加硬件权限：因为在 SDK 中调用了系统相机，需要在 `info.plist` 中添加 `Privacy - Camera Usage Description`；
- 7、调用 SDK 接口：通过 `#import <MGFaceIDLiveDetect/MGFaceIDLiveDetect.h>` 调用 SDK 中的 API 接口；

## 集成注意事项：

- 1、必须在网络连通的情况下使用，离线状态无法该 SDK 无法启动；
- 2、SDK 不支持 `bitcode` 设置。

## 2.5 SDK 错误码说明

result_code	result_message	含义解释
51000	SUCCESS	待比对照片与权威数据照片比对结果是同一个人
52000	PASS_LIVING_NOT_THE_SAME	待比对照片与权威数据照片比对结果不是同一个人
53000	NO_ID_CARD_NUMBER	权威数据无此身份证号
53000	ID_NUMBER_NAME_NOT_MATCH	身份证号，姓名不匹配
53000	NO_FACE_FOUND	权威数据照片中找不到人脸
53000	NO_ID_PHOTO	无法获取权威数据照片

53000	PHOTO_FORMAT_ERROR	权威数据照片格式错误
53000	DATA_SOURCE_ERROR	其他权威数据照片错误
54100	FAIL_LIVING_FACE_ATTACK	未经过活体判断，可能的原因：是假脸攻击
54100	REPLACED_FACE_ATTACK	发生换脸攻击，在做活体过程中出现两张不相同的人脸
54200	BIZ_TOKEN_DENIED	传入的 biz_token 不符合要求
54200	AUTHENTICATION_FAIL	鉴权失败，鉴权过程中非biz_token的错误
54200	MOBILE_PHONE_NOT_SUPPORT	手机在不支持列表里
54200	SDK_TOO_OLD	SDK版本过旧，已经不被支持
54200	MOBILE_PHONE_NO_AUTHORITY	没有权限（运动传感器、存储、相机）
54200	USER_CANCELLATION	用户活体失败，可能原因：用户取消了
54200	USER_TIMEOUT	用户活体失败，可能原因：验证过程超时
54200	VERIFICATION_FAILURE	用户活体失败，可能原因：验证失败
54200	UNDETECTED_FACE	用户活体失败，可能原因：未检测到人脸
54200	ACTION_ERROR	用户活体失败，可能原因：用户动作错误；
55000	IP_NOT_ALLOWED	不允许访问的IP
55000	NON_ENTERPRISE_CERTIFICATION	客户未进行企业认证
55000	BALANCE_NOT_ENOUGH	余额不足
55000	MORE_RETRY_TIMES	获取服务器配置时超过重试次数
55000	ACCOUNT_DISCONTINUED	用户帐号已停用
55000	EXPIRED_SIGN	签名过期
55000	INVALID_SIGN	无效的签名
55000	REPLAY_ATTACK	重放攻击，单次有效的签名被多次使用
56000	USER_CANCEL	用户取消
56000	NO_CAMERA_PERMISSION	没有打开相机的权限，请开启权限后重试
56000	ILLEGAL_PARAMETER	传入参数不合法
56000	DEVICE_NOT_SUPPORT	无法启动相机，请确认摄像头功能完好
56000	INVALID_BUNDLE_ID	信息验证失败，请重启程序或设备后重试
56000	NETWORK_ERROR	连不上互联网，请连接上互联网后重试
56000	FACE_INIT_FAIL	无法启动人脸识别，请稍后重试
56000	LIVENESS_DETECT_FAILED	活体检测不通过
56000	NO_SENSOR_PERMISSION	无法读取运动数据的权限，请开启权限后重试
56000	NO_WRITE_EXTERNAL_STORAGE_PERMISSION	没有读取写SD卡的权限
56000	INIT_FAILED	初始化失败
59000	LIVING_NOT_START	活体验证没有开始
59000	LIVING_IN_PROGRESS	正在进行验证
59000	LIVING_OVERTIME	操作超时，由于用户在长时间没有进行操作

50000	UNKOWN_ERROR	未知错误
50400	IMAGE_ERROR_UNSUPPORTED_FORMAT: <param>	参数<param>对应的图像无法解析，有可能不是图像文件、或有数据破损。<param>为 image_ref1、image_ref2中的一个。请注意：<param>只会有一项，即第一个满足条件的名称。
50400	MISSING_ARGUMENTS:<key>	缺少某个必选参数。
50400	BAD_ARGUMENTS:<key>	某个参数解析出错（比如必须是数字，但是输入的是非数字字符串；或者长度过长）
50400	NO_FACE_FOUND:<param>	表示上传的 image_ref 的图像中，没有检测到人脸。
50400	INVALID_IMAGE_SIZE<param>	客户上传的图像太大，具体是指像素尺寸的长或宽超过4096像素。<param>为 image_ref1、image_ref2中的一个。请注意：<param>只会有一项，即第一个满足条件的名称。
50403	AUTHORIZATION_ERROR:<reason>	调用次数超限、没有调用此API的权限，或者没有以当前方式调用此API的权限。
50429	CONCURRENCY_LIMIT_EXCEEDED	并发数超过限制
50404	API_NOT_FOUND	所调用的API不存在。
50413	Request Entity Too Large	客户发送的请求大小超过了20MB限制或单张照片大小超过了2MB。该错误的返回格式为纯文本，不是json格式。
50500	INTERNAL_ERROR	服务器内部错误，当此类错误发生时请再次请求，如果持续出现此类错误，请及时联系 FaceID 客服或商务

## 第3步(可选)：使用get\_result获取详细信息

### 接口说明

查询比对的结果

### 调用地址

GET [https://fid.market.alicloudapi.com/face/v1.2/sdk/get\\_result](https://fid.market.alicloudapi.com/face/v1.2/sdk/get_result)

### 签名方式

参见阿里云文档: [https://help.aliyun.com/document\\_detail/29490.html?spm=5176.730006-56956004-57124001-cmapi029634/A.content.10.O4CB1x](https://help.aliyun.com/document_detail/29490.html?spm=5176.730006-56956004-57124001-cmapi029634/A.content.10.O4CB1x)

### 参数

必选/ 可选	参数	类型	参数说明
必选	biz_token	String	FaceID回调给客户的biz_token信息
可选	verbose	Int	表示返回数据的详细程度，取值如下： 0：默认值，仅返回结论； 1：返回结论与摘要信息

## 返回值

参数	类型	说明
request_id	string	用于区分每一次请求的唯一的字符串。此字符串可以用于后续数据反查。此字段必定返回。
biz_no	string	传入的业务流水号，原封不动地返回。
time_used	int	整个请求所花费的时间，单位为毫秒。此字段必定返回。
biz_token	string	字符串，通过get_biz_token获取的唯一串码
result_code	int	表示本次验证的结果状态码，参见下面详述
result_message	string	开发者可通过此字段信息知晓具体的原因，参见下面详述
verification	object	人脸比对的详细结果，可能的字段如下： - idcard: Json类型，表示活体照片和权威数据比对的结果，具体包括以下字段： -- confidence: 比对结果的置信度，Float类型，取值 [0, 100] ，数字越大表示两张照片越可能是同一个人。 -- thresholds: 一组用于参考的置信度阈值，Json类型，包含三个字段，均为Float类型、取值 [0, 100] ： --- 1e-3: 误识率为千分之一的置信度阈值； --- 1e-4: 误识率为万分之一的置信度阈值； --- 1e-5: 误识率为十万分之一的置信度阈值； --- 1e-6: 误识率为百万分之一的置信度阈值； - ref1: Json类型，表示活体照片和image_ref1的比对结果，字段和上面的相同。 - ref2: Json类型，表示活体照片和image_ref2的比对结果，字段和上面的相同。
images	object	活体照片，字段如下： image_best: 活体检测获得的最优照片，base64编码。该照片会和权威照片或者用户上传的参考照片进行比对。
error	string	HTTP status code非200时返回，参见下面详述(HTTP状态码和对应的ERROR字段说明)

## Result\_code&result\_message说明

result_code	result_message	含义解释
1000	SUCCESS	待比照片与权威数据照片比对结果是同一个人
2000	PASS_LIVING_NOT_THE_SAME	待比照片与权威数据照片比对结果不是同一个人
3000	NO_ID_CARD_NUMBER	权威数据无此身份证号
3000	ID_NUMBER_NAME_NOT_MATCH	身份证号，姓名不匹配
3000	NO_FACE_FOUND	权威数据照片中找不到人脸
3000	NO_ID_PHOTO	无法获取权威数据照片
3000	PHOTO_FORMAT_ERROR	权威数据照片格式错误
3000	DATA_SOURCE_ERROR	其他权威数据照片错误
4100	FAIL_LIVING_FACE_ATTACK	未过活体判断，可能的原因：是假脸攻击
4100	REPLACED_FACE_ATTACK	发生换脸攻击，在做活体过程中出现两张不相同的人脸
4200	BIZ_TOKEN_DENIED	传入的 biz_token 不符合要求
4200	AUTHENTICATION_FAIL	鉴权失败
4200	MOBILE_PHONE_NOT_SUPPORT	手机在不支持列表里
4200	SDK_TOO_OLD	SDK版本过旧，已经不被支持
4200	MOBILE_PHONE_NO_AUTHORITY	没有权限（运动传感器、存储、相机）
4200	USER_CANCELLATION	用户活体失败，可能原因：用户取消了
4200	USER_TIMEOUT	用户活体失败，可能原因：验证过程超时
4200	VERIFICATION_FAILURE	用户活体失败，可能原因：验证失败
4200	UNDETECTED_FACE	用户活体失败，可能原因：未检测到人脸
4200	ACTION_ERROR	用户活体失败，可能原因：用户动作错误；
5000	IP_NOT_ALLOWED	不允许访问的IP
5000	NON_ENTERPRISE_CERTIFICATION	客户未进行企业认证
5000	BALANCE_NOT_ENOUGH	余额不足
5000	MORE_RETRY_TIMES	获取服务器配置时超过重试次数
5000	ACCOUNT_DISCONTINUED	用户帐号已停用
5000	EXPIRED_SIGN	签名过期
5000	INVALID_SIGN	无效的签名

5000	REPLAY_ATTACK	重放攻击，单次有效的签名被多次使用
6000	USER_CANCEL	用户取消
6000	NO_CAMERA_PERMISSION	没有打开相机的权限，请开启权限后重试
6000	ILLEGAL_PARAMETER	传入参数不合法
6000	DEVICE_NOT_SUPPORT	无法启动相机，请确认摄像头功能完好
6000	INVALID_BUNDLE_ID	信息验证失败，请重启程序或设备后重试
6000	NETWORK_ERROR	连不上互联网，请连接上互联网后重试
6000	FACE_INIT_FAIL	无法启动人脸识别，请稍后重试
6000	LIVENESS_DETECT_FAILED	活体检测不通过
6000	NO_SENSOR_PERMISSION	无法读取运动数据的权限，请开启权限后重试
6000	INIT_FAILED	初始化失败
9000	LIVING_NOT_START	活体验证没有开始
9000	LIVING_IN_PROGRESS	正在进行验证
9000	LIVING_OVERTIME	操作超时，由于用户在长时间没有进行操作

## HTTP状态码和对应的ERROR字段说明

HTTP 状态码	ERROR字段	说明
403	INVALID_SIGN	无效签名
403	AUTHORIZATION_ERROR:reason>	调用次数超限、没有调用此API的权限，或者没有以当前方式调用此API的权限。
429	CONCURRENCY_LIMIT_EXCEEDED	并发数超过限制
400	MISSING_ARGUMENTS:key>	缺少某个必选参数。
400	BAD_ARGUMENTS:key>	某个参数解析出错（比如必须是数字，但是输入的是非数字字符串；或者长度过长）
404	API_NOT_FOUND	所调用的API不存在。
413	Request Entity Too Large	客户发送的请求大小超过了20MB限制或单张照片大小超过了2MB。该错误的返回格式为纯文本，不是json格式。
500	INTERNAL_ERROR	服务器内部错误，当此类错误发生时请再次请求，如果持续出现此类错误，请及时联系 FaceID 客服或商务

## 例子

```
{
  "biz_token": "1532485677,fb241f74-e4e0-4396-ad74-eb8c1764aa1b",
  "images": {
    "image_best": "xxxx" # base64编码的活体照片
  },
  "verification": {
    "idcard": {
      "confidence": 83.23666,
      "thresholds": {
        "1e-3": 62.168713,
        "1e-5": 74.39926,
        "1e-4": 69.31534,
        "1e-6": 78.038055
      }
    }
  },
  "ref1": {
    "confidence": 82.93322,
    "thresholds": {
      "1e-3": 62.168713,
      "1e-5": 74.39926,
      "1e-4": 69.31534,
      "1e-6": 78.038055
    }
  }
}
```

```
    }  
  }  
},  
"request_id": "1532487162,7fb9b4af-4c6e-47a0-bb53-9de273368bc9",  
"time_used": 108,  
"biz_no": "1234567890",  
"result_message": "SUCCESS",  
"result_code": 1000  
}
```