

一登 iOS 平台 SDK 高级功能文档

SDK版本： SUPERID-IOS-SDK V 1.2.5(20150513)

系统支持： 目前一登 SDK 支持 iOS7.0 及以上版本，不兼容 iOS6 及以下版本。

设备支持： 目前一登 SDK 支持 iPhone 设备。

商务合作： 除了下文提及功能模块外，如果您需要进一步需求，可邮件联系一登商务：**contact@superid.me**

功能1 人脸属性功能

一登人脸属性功能是基于一登账号。当用户的一登账号授权开发者的应用以后，即提供了分析人脸属性高级功能。人脸属性高级功能通过开启摄像头识别用户人脸，返回当前用户的人脸属性，包括（表情、脸部坐标、微笑值、是否带眼镜、胡须密度等）。开发者可以使用这些参数为用户做更精准的数据推送、图片美化等自定义操作。

1.1 人脸属性快速集成

在您调用 SIDEemotionViewController 的当前 View Controller 中，您需要设置当前 VC 作为 SDK 的协议委托对象，并在当前 VC 中声明继承一登 SDK 的 Protocol（SuperIDDelegate）并声明 SDK 单例对象，如下代码所示。

```
@interface ViewController()<SuperIDDelegate>{

    SuperID      *superIdSdk;
}
//在ViewWillAppear方法中添加SDK的委托声明：
-(void)viewWillAppear:(BOOL)animated{

    [super viewWillAppear:animated];

    superIdSdk = [SuperID sharedInstance]; //获取SDK单例；
    superIdSdk.delegate = self;           //设置SDK的委托对象

}
```

API 接口的如下所示（一登用户人脸属性检测）

```
- (instancetype)obtainFaceFeatureViewControllerWithError:(NSError **)error;
```

具体工程示例的代码如下所示：

```
- (IBAction)getFaceFeature:(id)sender {

    //采用present的方式弹出人脸情绪的功能：
    NSError *error = nil;
    id SIDEemotionViewController = [[SuperID sharedInstance]obtainFaceFeatureViewC
ontrollerWithError:&error];

    if (SIDEemotionViewController) {

        [self presentViewController:SIDEemotionViewController animated:YES completi
on:nil];

    }else{

        NSLog(@"Error =%ld,%@",(long)[error code],[error localizedDescription]);

    }

}
```

人脸属性检测：一登用户人脸属性检测事件完成操作

当一登用户在一登 SDK 完成人脸属性检测事件后，SDK 将执行协议中的相应方法，开发者可在对应的方法中进行根据需要相应事件处理。

检测人脸属性操作完成事件将执行如下回调方法：

```
- (void)superID:(SuperID *)sender userDidFinishGetFaceFeatureWithFeatureInfo:(NSDi
ctionary *) featureInfo error:(NSError *)error;
```

参数解析：开发者可在该方法中，解析传参（featureInfo）的内容，其中包括的人脸属性有，颜值、年龄、性别、是否带眼睛、眼睛闭合度、胡须密度、表情等。当操作成功，error 为 nil，当操作失败，error 不为 nil。

注明：当出现网络错误、用户一登账号授权过期、当前环境无法识别人脸、用户主动放弃操作等事件时，一登 SDK 将执行该回调方法通知开发者。当用户解除当前应用的用户 ID 与一登账号绑定或尚未绑定的情况下，人脸信息检测将无法完成（如在其他设备登录同一应用并解除绑定授权）。一登 SDK 将通过此方法通知开发者。建议开发者在接收到该事件后，引导用户与一登账号进行绑定。对应的错误代码可参考快速集成文档中的错误代码解析。

1.2 人脸属性深度集成(自定义界面)

针对人脸属性功能模块，开发者可根据需要进行深度整合。

步骤1 头文件导入

在深度继承的 VC 中，声明头文件：`#import "SIDFaceFeatureViewController.h"`。新建 view controller 文件并将该 VC 的父类修改为：`SIDFaceFeatureViewController`（如下所示）。

```
@interface AdvancedFaceFeatureViewController : SIDFaceFeatureViewController
```

步骤2 初始化 VC

初始化父类的方法，开发者需要在 `viewDidLoad` 的方法中初始化继承父类 `View` 中的方法。

初始化API如下：

```
- (BOOL)configureFaceFeatureViewWithDuration:(float)duration withFrameWidth:(float)width high:(float)high;
```

参数解析：`duration` 为单次识别人脸属性的持续时长；`width` 为获取摄像头实时视频流视窗的宽度（不大于 480）；`high` 为获取摄像头实时视频流视窗的高度（不大于 640）。当`width`和`high`设置为 0, 采用 SDK 默认输出视窗大小。

初始化方法如下：

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    [self configureFaceFeatureViewWithDuration:10.0f withFrameWidth:320 high:320];  
}
```

步骤3 启动/结束检测

启动人脸属性检测功能：深度合作模块的人脸属性检测功能在界面设计和部分功能设计由开发者自行设计。当需要启动人脸属性检测功能时，在您的对应的事件响应处理中，调用如下接口：

启动人脸信息检测：

```
- (void)startDetectFaceFeature;
```

该接口调用后，人脸属性检测开启，开发者可自行引导用户当前正在使用人脸识别。该功能模块最大执行事件为 30 秒，当人脸属性检测完成后，SDK 将通过方法调用通知开发者。当操作超时，该人脸检测功能将自动关闭，并通知开发者检测失败。

结束人脸属性检测：

```
- (void)stopDetectFaceFeature;
```

步骤4 人脸检测完成

人脸属性检测功能模块在执行过程分为两部分，第一部分为人脸检测，第二部分为人脸属性分析。当SDK完成人脸检测后，我们将通知开发者已经检测到用户的脸部，开始进行人脸属性分析。该方法便于开发者设计 UX.

人脸检测事件完成的方法执行：

```
- (void)didFinishDetectFace;
```

人脸检测事件完成示例代码：

```
- (void)didFinishDetectFace{
    //do your operation here;
}
```

步骤5 事件完成回调

启动人脸信息检测事件完成：当 SDK 方法执行人脸属性检测完成后，开发者可通过以下方法分别处理不同响应状态：

人脸检测事件执行成功：

```
- (void)getFaceFeatureSucceedWithFeatureInfo:(NSDictionary *) featureInfo;
```

人脸检测事件执行失败：

```
- (void)getFaceFeatureFail:(NSError *)error;
```

注明：该情况发生在用户没有开启摄像头权限、用户操作不当致 SDK 人脸检测器没有检测到人脸，网络连接问题，用户尚未绑定一登账号，或长时间不登陆导致授权 Token 过期等。当用户尚未绑定一登账号或用户解

除绑定时，SDK 将执行该方法。开发者可在继承的子类内执行该方法，引导用户先进行人脸绑定。当用户长时间不登录导致授权过期时，SDK 将执行该方法。开发者可在继承的子类内执行该方法，引导用户先进行人脸绑定。

1.3 人脸属性接口返回数据说明

人脸属性属性如下所示：

```
{
  "smiling": {
    "result": 0,
    "score": 0.001059
  },
  "male": {
    "result": 1,
    "score": 0.999986
  },
  "eyeglasses": {
    "result": 0,
    "score": 0.001633
  },
  "sunglasses": {
    "result": 0,
    "score": 0.005324
  },
  "age": 26.85,
  "mustache": {
    "result": 0,
    "score": 0.00102
  },
  "emotions": {
    "angry": 0.007319,
    "calm": 0.792951,
    "confused": 0.160389,
    "disgust": 0.004511,
    "happy": 0.016296,
    "sad": 0.015238,
    "surprised": 0.003295
  },
  "attractive": {
    "result": 0,
    "score": 0.117668
  },
  "orientation": {
    "yaw": 3.852539,
    "pitch": -0.84375,
    "roll": -4.250381
  }
}
```

```
{
  "blink": {
    "result": 1,
    "score": 0.515972
  },
  "mouth_open": {
    "result": 0,
    "score": 0.409592
  },
  "components": {
    "left_eye": [
      170.37178,
      327.428619
    ],
    "right_eye": [
      295.026978,
      332.592041
    ],
    "nose": [
      222.983948,
      385.265198
    ],
    "mouth_left": [
      176.645065,
      466.788452
    ],
    "mouth_right": [
      276.454803,
      468.780487
    ]
  },
  "position": {
    "top": 264,
    "left": 134,
    "right": 408,
    "bottom": 538
  },
}
```

人脸属性数据解析：

人脸属性	解析
smiling	【微笑值】 result=1表示微笑，score表示微笑的程度
male	【性别】 result=1表示男性，score表示为男性的可信度，result=0表示女性
eyeglasses	【是否戴眼镜】 result=1表示有戴，0表示没戴
sunglasses	【是否戴太阳眼镜】 result=1表示有戴，0表示没戴
age	【年龄】
mustache	【胡须密度】 result=1表示有胡须，0表示没胡须，score表示有胡须的程度
emotions	【表情】 包含生气、平静、困惑、愤怒、快乐、悲伤、惊喜7种情绪所占的分值，可以取最大值作为当前人脸的表情
attractive	【颜值】 取值范围为0~1
orientation	【人脸旋转角】， 是人脸眼中间和嘴中间的连线，与垂线的夹角
blink	【闭眼】 result=1表示闭眼，score表示闭眼的程度
mouth_open	【嘴巴张开度】 result=1表示嘴巴张开，score表示嘴巴张开的程度
components	【人脸关键点】
position	【人脸框】