

# 一登SDK使用流程

- 1.在首页点击【免费接入人脸登录】，在一登Github上下载对应的SDK版本、Demo源码和技术文档；
  - 2.回到官网，点击下一步，填写应用名及应用bundle Identifier（包含debug时的签名及打包应用后的签名）；
  - 3.申请成功后，系统将会发送APP\_ID、APP\_Secret到开发者邮箱；
  - 4.阅读技术文档并集成SDK；
  - 5.测试
- 

## 1 集成准备

### 1.1 获取App\_ID等：获取App\_ID、App\_Secret

---

获取iOS App的bundle Identifier，获取路径：在开发者的Xcode工程内，选择：项目->General->Identify->Bundle Identifier(图1.1). 您需要在 [SDK申请](#) 页面中填写此Bundle Identifier，申请SDK使用，获取App\_ID、App\_Secret。

系统版本：目前一登SDK支持iOS7.0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。

### 1.2 导入SDK

---

步骤1：下载一登SDK，将下载包中的所有资源文件拖拽到您的工程项目目录中或者在Xcode中，选择File->Add Files to "your project",选中所下载的文件。

步骤2：配置APP的Build Settings。配置方法为:在您的Xcode项目内，选择：xcodproj->Build Settings->Linking->Other Linker Flags,在Other Linker Flags选项中，双击该选项，点击弹出框左下角的+按钮，分别添加“-ObjC”字符和“-lstdc++”字符。

步骤3：添加依赖库。在您的Xcode项目中，选择：xcodproj->Build Phases->Link Binary With Libraries->点击左下角  按钮->添加 `CoreTelephony.framework` 依赖库。

---

## 2 权限声明

### 2.1 摄像头权限

---



具体工程入口的代码如下所示：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    [[SuperIDSDK sharedInstance] initWithCredentialsAppID:@"您的AppID"  
                                                AppSecret:@"您的AppSecret"];  
  
    return YES;  
}
```

## 3.2 选择语言模式

目前一登SDK支持英语、简体中文、繁体中文三种语言模式。您可以调用以下接口设置SDK的语言模式：

```
+(void)SuperIDSDKLanguageMode:(languageType)type;
```

设置参数分别为：`autoMode`（自动模式，SDK将根据系统语言进行自动切换，当系统当前语言为非中文时，显示英文）；`simplified_Chinese`（简体中文模式,采用该模式时SDK语言将一直保持简体中文，下同）；`English`（英文模式）；`tradition_Chinese`（繁体中文模式）。

注明：当该接口未被调用时，系统默认采用自动模式。

# 4 一登用户登录

## 4.1 添加登录触发事件

开始使用一登SDK登录功能时，开发者可在当前APP登录界面添加按钮，并设置其触发事件，并调用一登账号登录接口。用户在当前App首次使用一登时，一登SDK会为用户生成三方App的用户Uid,三方开发者可以使用这个Uid作为其的账号体系中的用户id,也可以自行引导当前一登用户与App的账号进行绑定，当开发者获取该用户的Uid后，调用可调用SDK的updateAppUid接口完成首次一登账号与三方App用户账号的绑定。

**API接口的如下所示（获取一登账号登录VC）**

```
-(id)obtainSuperIDLoginView;
```

具体工程入口的代码如下所示：

```

- (IBAction)LoginWithSuperID:(id)sender {
    //获取SuperIDLoginView:
    NSError *error = nil;
    id SuperIDLoginView=[SDK obtainSuperIDLoginView:&error];

    if (error) {

        NSLog(@"loginView Error =%d,%@",[error code],[error localizedDescription])
    }
    if (SuperIDLoginView) {

        //采用present的方式弹出LoginView
        [self presentViewController:SuperIDLoginView animated:YES completion:^(void){

        }]];
    }
}

```

在您调用SuperIDLoginView的当前View Controller中，您需要设置当前VC作为SDK的协议委托对象，并在头文件中声明继承一登SDK的Protocol（SuperIDSDKDelegate）。

具体工程代码如下所示：

```

-(void)viewWillAppear{
    ...
    SDK.delegate = self;
}

```

注明：建议开发者将SDK的委托代码置于ViewWillAppear方法中。

## 4.2 登录完成事件处理

当用户完成一登账号登录事件后，SDK将执行相应的协议回调方法，开发者可在协议方法中，进行相应的登录事件完成操作。

一登用户登录成功将执行如下回调方法：

```

-(void)userDidFinishLoginWithSuperID:(NSDictionary *)userInfo andAppUID:(NSString *)appUid;

```

注明：userInfo为用户一登账号的账户信息（包括昵称、移动电话、头像、AppUid等），开发者可根据需要进行选取使用。

一登用户登录失败将执行如下回调方法：

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

注明：当出现用户主动放弃操作、网络连接失败、用户账户有误、人脸比对失败的情况以及其他原因导致登录失败时，一登SDK将执行该回调接口方法。开发者需根据实际需求进行相应的事件处理。具体error事件描述可详见下文。

## 5 一登账号绑定

### 5.1 有手机号码绑定：开发者App账号体系拥有用户手机号码

使用一登SDK绑定的API接口，执行步骤与一登账号登录方法一致。开发者需要在当前VC中声明SDK的协议继承，将自身VC作为一登SDK的委托对象。在绑定过程中，开发者需要传入App应用用户的Uid(必选)以及用户的手机号码（可选）进行绑定。目前一登SDK接口提供三方应用用户信息的存入（可选）。您的用户信息内容可包括：用户昵称（Name），用户邮箱（Email），用户头像地址（Avatar）三项内容。若您的用户信息中不存在以上项目，可将此传参设置为nil。调用此接口可在您的相应的App的按钮触发事件中添加，与一登用户登录方法一致。

**API接口的如下所示（获取一登账号绑定VC）**

```
-(id)obtainSuperIDAouthViewWithAppUid:(NSString *)uid
    MobliePhone:(NSString *)mobilePhone
    userAppInfo:(NSDictionary *)userInfo
    error:(NSError **)error;
```

具体工程示例代码如下所示：

```

- (IBAction)withPhoneBundlingSuperID:(id)sender {

    NSMutableDictionary *userInfo = [[NSMutableDictionary alloc] init];
    [userInfo setObject:@"XXXX" forKey:@"name"]; //
    [userInfo setObject:@"XXXXXX@gmail.com" forKey:@"email"];
    [userInfo setObject:@"http://XXXXXXXXXXXXXXXX" forKey:@"avatar"];
    NSError *error = nil;

    id SuperIDAouthView = [[SuperIDSDK sharedInstance] obtainSuperIDAouthViewWithApp
    pUid:@"用户在本应用中的UserID"
    MobliePhone:@"136XXXXXXXX" userAppInfo:userInfo error:&error];

    NSLog(@"loginView Error =%d,%@",[error code],[error localizedDescription]);

    if (SuperIDAouthView) {

        [self presentViewController: SuperIDAouthView animated:YES completion:nil]
    ;
    }
}

```

注明：用户信息存入的userInfo为NSDictionary对象，每项信息的字段名（KEY）必须与示例代码中的保持一致。如无，可置为nil，当开发者需传入手机号码时，传入的手机号必须为中国地区的手机号码。

## 5.2 无手机号码绑定：开发者App账号体系无用户手机号码

当三方开发者的账号体系内没有用户手机号码这项信息时，可以将以上的绑定API接口入参项：设置为nil。一登SDK在用户绑定开发者账号成功后，将执行协议方法，并传递用户一登账号的个人信息。

## 5.3 绑定完成：一登账号绑定完成事件处理

当用户完成一登账号绑定事件后，SDK将执行相应的协议回调方法，开发者可在协议方法中，进行相应的登录事件完成操作。

一登用户绑定App成功将执行如下回调方法：

```

-(void)userDidFinishAouthAppWithSuperID:(NSDictionary *)userInfo andAppUID:(NSString *)appUid;

```

注明：userInfo为用户一登账号的账户信息以及成功绑定的App的用户Uid，开发者可根据需要进行选取使用。

一登用户绑定App失败将执行如下回调方法：

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

注明：当出现用户主动放弃操作、网络连接失败、用户账户有误、人脸比对失败的情况以及其他原因导致账号绑定失败时，一登SDK将执行该回调接口方法。开发者可根据实际需求进行相应的事件处理。具体error事件描述可详见下文。

## 6 一登账号解除绑定

### 6.1 解除绑定：一登账号与App账号解除绑定

开发者使用一登SDK解绑的API接口，执行步骤与一登账号登录方法一致。

API接口的如下所示（获取一登账号绑定的View）

```
-(void)superIDUserCancelTheAuthorization;
```

具体工程示例的代码如下所示：

```
- (IBAction)LogoutWithSuperID:(id)sender {  
  
    [[SuperIDSDK sharedInstance] superIDUserCancelTheAuthorization];  
}
```

### 6.2 解绑完成：一登账号解绑完成事件处理

当用户主动选择解除账号绑定后，开发者可在协议方法中接受事件完成回调。

一登用户解绑App成功将执行如下回调方法：

```
-(void)userCancelAouthWithSuperIDSuccess;
```

注明：开发者可在该方法中，提醒用户解绑完成。

一登用户解绑App失败将执行如下回调方法：

```
-(void)userCancelAouthWithSuperIDFail;
```

注明：当出现用户长时间不登陆导致一登账号授权过期、当前用户尚未绑定一登账号情况或在其他设备解其一登账号与App账号的绑定时，一登用户解绑事件执行失败。

## 6.3 登出操作：不解除账号授权绑定

当开发者的App需要执行登出操作时，此刻无须解除SuperID账号与App账号的关联。开发者只需要在登出的方法中添加以下接口方法即可：

```
-(void)logoutSuperIDAccountInApp;
```

# 7 一登用户人脸信息检测

## 7.1 人脸信息检测操作

一登SDK提供一登用户的人脸信息检测功能模块，开发者可使用该模块引导用户使用，并通过使用一登SDK检测的人脸信息进行智能推荐等拓展功能。使用该功能模块的操作步骤与一登账号登录的步骤类似。

**API接口的如下所示（一登用户人脸特征检测）**

```
-(id)obtainSuperIDFaceFeatureViewWithPreview:(NSError **)error;
```

具体工程示例的代码如下所示：

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

    NSError *error = nil;
    id emotionView = [[SuperIDSDK sharedInstance]obtainSuperIDFaceFeatureViewWithP
review:nil];
    NSLog(@"loginView Error =%d,%@",[error code],[error localizedDescription]);
    if (emotionView) {

        [self presentViewController:emotionView animated:YES completion:nil];

    }
}
```

## 7.2 人脸信息检测：一登用户人脸信息检测事件完成操作

当一登用户在一登SDK完成人脸信息检测事件后，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。

检测人脸信息成功事件将执行如下回调方法：



```
-(void)getSuperIDUserFaceFeatureSuccess:(NSDictionary *) featureData;
```

注明：开发者可在该方法中，解析传参（featureData）的内容，其中包括的人脸信息有，美貌度、年龄、性别、是否带眼睛、眼睛闭合度、胡须密度、当前用户心情等。

检测人脸信息失败事件将执行如下回调方法：

```
-(void)getSuperIDUserFaceFeatureFail;
```

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。

用户已解除一登账号绑定事件将执行如下回调方法：

```
-(void)userHasCancelAouthToCurrentAccount;
```

注明：当用户解除当前App Uid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。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。

## 7.2 人脸信息检测：权限查询

目前一登SDK的人脸信息检测功能只开发给深度合作的开发者，您可以调用以下接口查询是否拥有该权限，如无，调用以上人脸信息接口时将无法获得正确返回值。查询人脸信息检测接口权限如下：

```
-(BOOL)checkCurrentAppTokenCanAccessToFaceFeatureFunction;
```

## 8 用户绑定状态查询

### 8.1 查询绑定状态：查询当前App用户与一登账号绑定状态

开发者需要在用户选择绑定或解绑操作前进行用户绑定状态的查询，并根据查询结果引导用户完成一登账号的其他操作。

API接口的如下所示（一登用户绑定状态查询）

```
-(void)checkUserAouthrizedStateWithUid:(NSString *)uid;
```

### 8.2 查询事件处理：查询完成的事件处理：

查询完成后，一登SDK将执行如下回调方法通知开发者。

用户绑定状态查询事件完成将执行如下回调方法：

```
-(void)appUserUidAouthStateResponse:(appUserUidAouthState)aouthState;
```

注明：回调参数aouthState包含三种状态：UidHasAouth（uid已与一登账号绑定），UidNoAouth（uid与一登账号尚未绑定），CheckUserAouthFail（当出现网络问题等异常状况导致的查询失败）。

## 9 更新App用户ID

### 9.1 更新用户ID：更新当前的App用户ID

对于已用账号体系的App，开发者在用户首次使用一登账号登录后，可通过如下接口更新一登账号绑定的App用户ID，该方法便于开发者在未知用户使用一登账号登录后的后续账号绑定操作。

API接口的如下所示（一登用户绑定状态查询）

```
-(void)updateAppUserUidToSuperIDAccount:(NSString *)uid;
```

### 9.2 更新事件完成操作：更新App用户ID事件完成操作

当一登SDK完成用户ID更新到用户一登账号的事件完成后，一登将执行如下方法通知开发者。

```
-(void)updateUserAppUidResponse:(updateAppUserUidState)updateUid;
```

注明：回调参数updateAppUserInfo包含三种状态：uidUpdateSuccess（更新成功），uidHasExist（此App Uid已被一登账号绑定），userNoAouthToSuperID（用户解除绑定或尚未绑定一登账号，无法更新），userAccessExpired（用户授权已过期，无法更新）。开发者可根据相应的事件进行后续操作。

## 10 更新App用户个人信息

### 10.1 更新用户信息：更新App用户个人信息

当用户更改了App中的个人信息，开发者可通过如下接口将更改后的信息上传指一登账号进行信息更新。

```
-(void)updateAppUserInfoToSuperIDAccount:(NSDictionary *)userInfo;
```

## 10.2 更新信息事件完成操作：更新App用户个人信息事件完成操作

```
-(void)updateUserAppInfoStateResponse:(updateAppUserInfoState)updateAppUserInfo;
```

注明：回调参数updateAppUserInfo包含三种状态：updateAppUserInfoSuccess（更新成功），updateAppUserInfoFail（更新失败），userNoAouthToSuperID（用户解除绑定或尚未绑定一登账号，无法更新），userAccessExpired（用户授权已过期，无法更新）。开发者可根据相应的事件进行后续操作。

## 11 回调错误解析

在使用一登SDK过程，当出现错误反馈时，开发者可查询如下列表进行信息定位并进行相应操作：

- **error code:-1000** 表示一登用户登录操作失败
- **error code:-1001** 表示当前网络环境有问题
- **error code:-1002** 表示一登用户手机号在使用过程申请验证码超过当天限制
- **error code:-1003** 表示一登用户手机号申请验证码失败
- **error code:-1004** 表示用户不允许访问摄像头
- **error code:-1005** 表示当前一登账号已经被其他uid绑定
- **error code:-1006** 表示一登用户绑定操作失败
- **error code:-1007** 表示当前的应用授权已过期
- **error code:-1008** 表示缺乏传入参数UID
- **error code:-1009** 表示用户未授权当前APP
- **error code:-1010** 当前SDK版本过期，请开发者升级SuperIDSDK版本
- **error code:-1011** 表示SDK初始化失败
- **error code:-1012** 当前开发者账号未有权限使用人脸表情
- **error code:-1013** 表示当前Apptoken过期，SDK将直接续期
- **error code:-1014** 传入手机号码为非中国地区

## 12 网络错误通知

当开发者执行查询、更新uid/user info的方法时，由于网络错误造成的操作无法完成，一登SDK将在如下的接口通知三方开发者。

```
-(void)receieveNetWorkError;
```

开发者可在此回到方法中提醒用户检测当前网络。

## 13 深度整合：人脸信息功能模块深度整合

针对人脸信息功能模块，开发者可更具需要进行深度整合。

### 步骤1 头文件导入

导入头文件：`#import "SuperIDFaceFeatureViewClass.h"`。新建view controller文件并将该VC的父类修改为：`SuperIDFaceFeatureViewClass`。

```
@interface FaceFeatureViewController:SuperIDFaceFeatureViewClass
```

### 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。

初始化方法如下：

```
- (void)viewDidLoad {
    [super viewDidLoad];
    ....
    [self initFaceFeatureViewWithoutPreviewInDuration];
}
```

### 步骤3 开启检测

启动人脸信息检测功能：深度合作模块的人脸信息检测功能在界面设计和部分功能设计由开发者自行设计。当需要启动人脸信息检测功能时，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，人脸信息检测开启，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，当人脸信息检测完成后，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。

### 步骤4 人脸检测完成

人脸信息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，第二部分为人脸信息分析。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，开始进行人脸信息分析。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：
```

```
[self 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方法执行人脸信息检测完成后，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)getFaceFeatureNOPreviewSuccessWithfeatureData:(NSDictionary *)featureData;
```

人脸检测事件执行失败：

```
-(void)getFaceFeatureNoPreviewFail;
```

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，网络连接问题等。

用户尚未绑定一登账号或解除绑定：

```
-(void)userHasCancelAouthWithCurrentAccount;
```

注明：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。

用户授权已过期：

```
-(void)userAouthrizehasExpired;
```

注明：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。

# 14 人脸信息： 参数列表

| 关键词           | 解析       |
|---------------|----------|
| age           | 年龄       |
| beard         | 美貌值      |
| boundingbox   | 人脸框      |
| confidence    | 是否人脸     |
| eye_closed    | 眼睛闭合度    |
| eye_left      | 左眼坐标     |
| eye_right     | 右眼坐标     |
| glasses       | 眼镜       |
| mouthopenwide | 嘴巴张开度    |
| mustache      | 胡须密度（胡子） |
| sex           | 性别       |
| smile         | 微笑值      |
| sunglasses    | 太阳镜      |
| beard         | 美貌值      |

| emotion   | 情绪 |
|-----------|----|
| calm      | 平静 |
| confused  | 困惑 |
| happy     | 快乐 |
| angry     | 愤怒 |
| surprised | 惊讶 |
| sad       | 悲伤 |
| disgust   | 厌恶 |

人脸情绪 是人脸算法提取当前人脸信息与人类表情库上的六种（另加中性表情： calm）基本情绪做比对后，

根据获取的人脸给每一个情绪参数赋值（范围：0-1），给出有效的三个情绪的值。目前demo中的做法是把返回的三个情绪的值中最大的一个作为当前人脸的代表情绪。

例：

```
"emotion" : {
  "confused" : 0.7,
  "calm" : 0.49,
  "surprised" : 0.05
}
```

以上表示当前人脸的情绪为confused概率值为 0.7 ； calm值 0.49 ； surprised值 0.05 。也就是当前人脸的情绪最接近confused，但也有calm、surprised的相似度。

在具体应用上，可以根据算法给出值最大的情绪参数作为需求判断；也可以根据第一、第二、第三个情绪值作为辅助参数，推算出新的需求判断。

## 关于人类表情库的理论参考：

保罗·艾克曼（Paul Ekman) Emotion in the Human

Face JAFFE人脸表情数据库