

一登 iOS 平台 SDK 文档

SDK版本： SUPERID-IOS-SDK V1.2.8(20150603)

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。

1 一登 SDK 介绍

一登 SDK 为应用提供了便捷的刷脸登录模式。通过一登账号体系，移动终端用户可快速登录开发者应用。同时，一登 SDK 提供了人脸属性高级功能，让移动开发者可以基于人脸属性，为应用做更精准的数据推送。

一登 SDK 封装集成了最新的 OAuth2.0 授权方式，并将通讯过程封装于 SDK 内部，开发者仅需根据文档说明即可快速集成 SDK。

2 集成准备

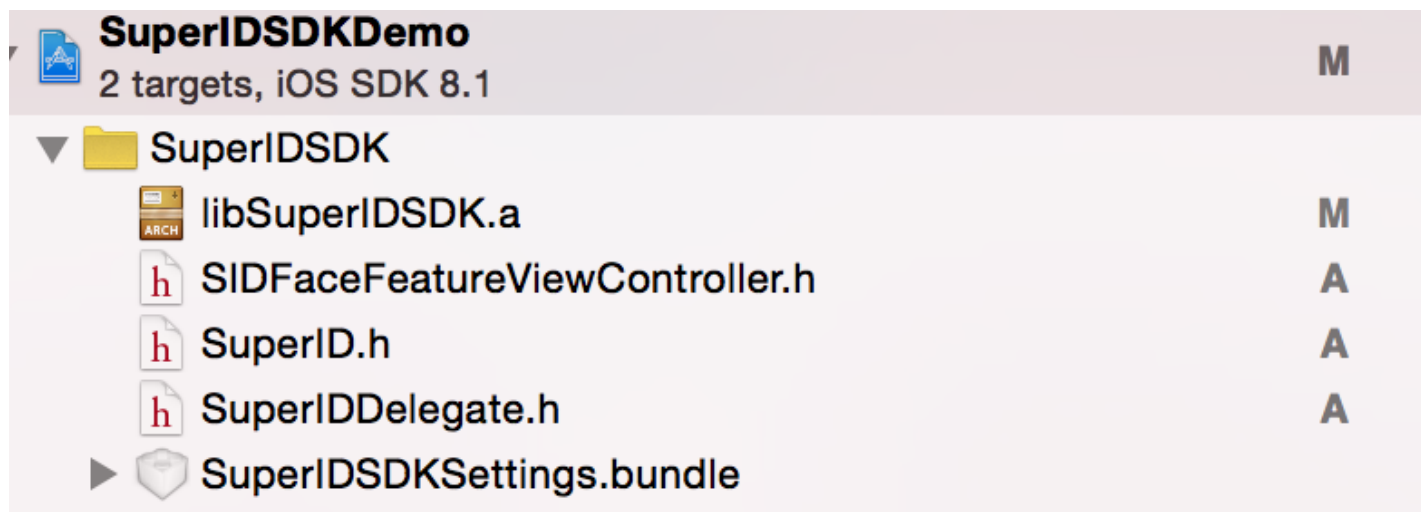
2.1 SDK 接入申请

- 1.在[官网首页](#)点击【注册】完成一登开发者注册。
- 2.登录一登开发者中心，点击【SDK 下载】，在一登 Github 上下载对应的 SDK 版本、Demo 源码和技术文档。
- 3.登录一登开发者中心，点击【创建新应用】，填写信息。
- 4.添加应用成功后，获取 APP_ID、APP_SECRET。

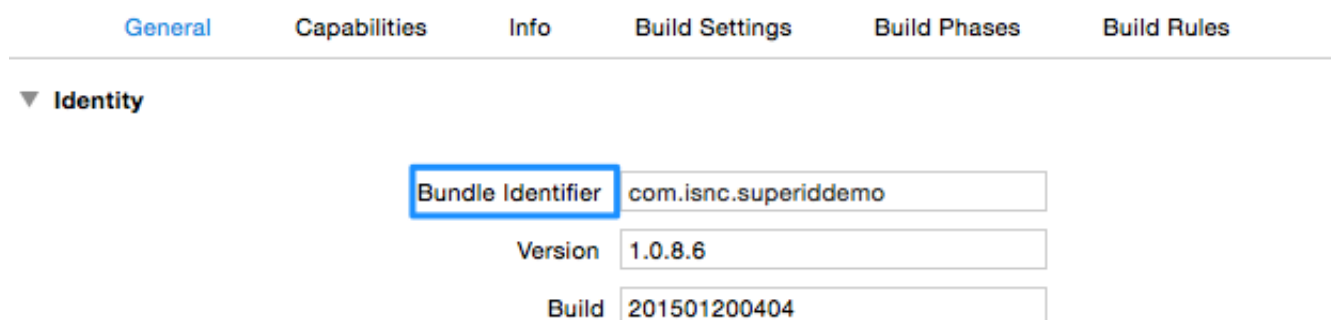
2.2 SDK 接入设置

方法1

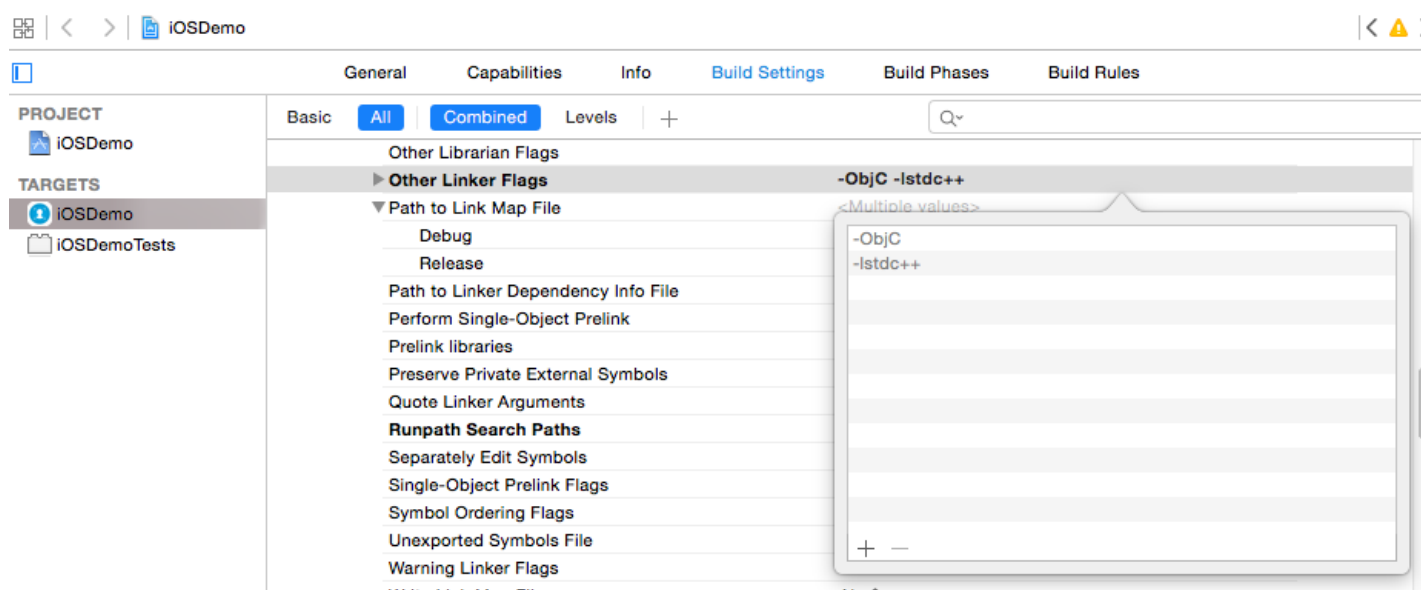
步骤1： 在一登 GitHub 下载一登 iOS SDK。将 SDK 包中的文件添加至本地工程，其中包括：SuperIDSDKSettings.bundle、libSuperIDSDK.a、SIDFaceFeatureViewController.h、SuperID.h、SuperIDDelegate.h 共5个文件。



步骤2： 设定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。在 Xcode 下的 Bundle 截图如下：



步骤3： 在工程t引入静态库之后"需要在编译时添加 -ObjC 和 -lstdc++ 编译选项。方法：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（如下图）。



方法2

步骤1： 使用CocoaPods安装一登SDK

```
platform :ios, '7.0'
pod "SuperID"
```

执行 `pod install` 将会安装最新版本的一登SDK

步骤2： 设定Bundle Identifier，操作步骤如上所示。

2.3 添加依赖库

如果你的应用无法正常通过编译，请添加SDK所需的依赖库。主要为：

CoreTelephony.framework

CoreMedia.framework

AVFoundation.framework

libc++.dylib

添加路径为：工程->Build Phases->Link Binary With Libraries->Add->选择上述的依赖库。

3 权限声明

3.1 摄像头权限

一登（SuperID）账号体系是基于用户的人脸特征进行登录、绑定。在使用过程中，SDK 将访问 iPhone 设备的前置摄像头，在用户首次使用 SuperID 时，会向用户申请使用摄像头权限，用户需允许访问方可进行后续操作。当用户选择不允许时，一登 SDK 将无法使用，SDK 会将该事件通知开发者，开发者可根据情况进行相应的处理。建议开发者进行如下操作：当开发者收到用户不允许访问摄像头事件时，在您的应用界面上弹出 AlertView 引导用户进行权限的开启。

参考代码如下：

```
NSBundle *bundle =[NSBundle mainBundle];
NSDictionary *info =[bundle infoDictionary];
NSString *AppName =[info objectForKey:@"CFBundleDisplayName"];
NSString *MessageString=[NSString stringWithFormat:@"请手动开启相机权限：iPhone设置-隐私-相机-(%)打开",AppName];
UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"无法启动相机"
                                                message:MessageString
                                                delegate:nil
                                                cancelButtonTitle:@"我知道了"
                                                otherButtonTitles:nil, nil];

[alert show];
```

3.2 相册权限

用户在首次使用一登登录或账号绑定时，可在 SDK 的页面内进行个人头像的设置，当用户选择从“从手机相册选择”时，开发者的应用将向用户请求访问相册的权限。当用户选择允许时，可选择个人相册图片作为头像资源进行设置。当用户选择不允许时，该功能不可使用，但不影响一登 SDK 的正常登录、绑定等操作。

4 SDK 初始化

4.1 头文件声明：

在您所有引用到 SDK 的程序代码中，声明 SDK 头文件的引用：`#import "SuperID.h"`，程序启动时，在代码中向一登 SDK 注册你的 AppID 和 AppSecret（建议将一登 SDK 的头文件声明置于 pch 文件中）。

4.2 添加 SDK 的初始化代码：

API 接口的如下所示

```
- (void)registerAppWithAppID:(NSString *)appID withAppSecret:(NSString *)appSecret
;
```

具体工程入口的代码如下所示（重写 **AppDelegate.m** 中的方法）：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
  
    [[SuperID sharedInstance]registerAppWithAppID:@"应用的AppID" withAppSecret:@"应用的AppSecret"];  
  
    return YES;  
}
```

4.3 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)setLanguageMode:(SIDLanguageMode)type;
```

设置参数分别为：`SIDAutoMode`（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）；`SIDSimplifiedChineseMode`（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）；`SIDEnglishMode`（英文模式）；`SIDTraditionChineseMode`（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。

4.4 设置 SDK 的调试模式

在开发者调试过程中，可在应用启动时设置 SDK 的调试模式，在应用发布时关闭一登 SDK 的调试模式。

API接口的如下所示

```
+ (void)setDebugMode:(BOOL)flag;
```

注明：当 flag 设置为 YES 时，开启 SDK 调试模式。一登 SDK 会在应用接入 SDK 过程中，输出相应的 Log 到调试平台。当 flag 为 NO 时，关闭一登 SDK 调试模式。

5 一登账号登录

5.1 名词解析

完成一登 SSO 登录或者一登账号授权绑定所需的几个名词：

名词	解析
一登账号 (SuperID)	用户的人脸账号，以手机号码作为账号标识
应用 userID(AppUid)	当前应用用户的 userID, 需三方应用提供
一登账号用户 信息	包含用户一登账号的昵称、移动电话、头像等
app id 和 app secret	在应用内注册一登 SDK 需提供这两项信息
appToken	用于一登 SDK 基础通讯使用，SDK 初始化成功自动获取，由 SDK 内部维护，对开发者不可见。
accessToken	用户授权登录成功后获取，用于后续操作（查询、更新信息、获取信息）的用户身份认证使用，其信息对开发者不可见，token 状态将通过事件回调通知开发者。

5.2 一登登录

使用一登账号实现快速刷脸登录，需完成用户一登账号和应用账号的授权关联过程。以下分为两种情况进行指南：

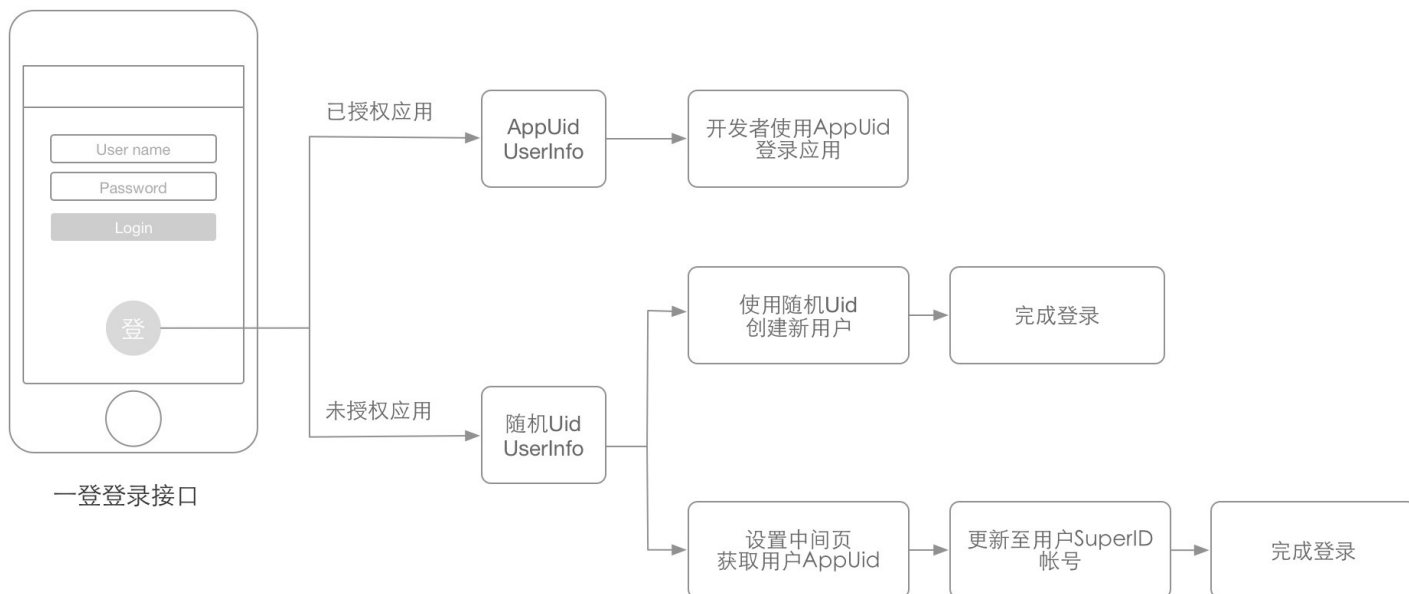
场景1：当用户尚未处于为登录应用状态（开发者未知当前用户 Uid）。此时用户点击登录界面的一登按钮(API 见下文)，开发者可根据以下情况选择合适的处理方案。

当用户直接点击一登登录按钮，如果当前一登用户已授权你的应用，直接登录成功后，我们会返回当前用户的在开发者应用中的用户 ID 和一登账号的用户信息。开发者可使用该用户 ID 直接登录成功。

如果当前用户尚未授权，我们会引导其授权你的应用，并返回随机的应用用户 ID 和一登账号的用户信息。此时，你有以下处理方案：

- 1.使用该随机用户 ID 直接为当前一登用户创建新的应用用户；
- 2.为当前一登用户创建新的应用账号，将你账号体系的用户 ID 更新到当前的一登用户账号中（更新接口见下文），登录。
- 3.开发者自行设置中间绑定页，引导用户输入在你应用中的账号密码，获取该用户 ID 后，更新用户 ID 到当前的一登用户账号。完成登录。

以上未授权的处理事项只在用户授权时发生，当用户完成授权关联，以后即可直接刷脸登录。对于形成账号体系内已有大量老用户的开发者，可使用方法3 完成账号关联。



注明：您可以直接使用一登账号的用户信息作为应用用户的基础信息，简化用户编辑信息流程。

5.3 登录接口说明

在您调用 SuperIDLoginView 的当前 View Controller 中，您需要声明一登 SDK 头文件的引用

`#import "SuperID.h"`（下同），并设置当前 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（SuperIDDelegate）以及定义 SDK 对象和 SuperIDLoginView 对象。如下代码所示：

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

    SuperID      *superIdSdk;                //SuperID 实例对象

}

@end
```

在 ViewWillAppear 获取 SDK 单例和并设置当前 VC 为 SDK 的委托对象。

具体工程代码如下所示：

```
-(void)viewWillAppear{
    ...
    [super viewWillAppear:animated];

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

注明：开发者应在 viewWillAppear 中设置 SDK 的委托对象，当应用从其他 VC 回来回到当前 VC 后，确保 SDK 的委托对象为当前 VC。

开始使用一登 SDK 刷脸登录功能时，开发者可在当前应用登录界面添加按钮，并设置其触发事件，并调用一登账号登录接口。用户在当前应用首次使用一登时，一登 SDK 会为用户生成三方应用的用户 ID，三方开发者可以使用这个 ID 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应用的账号进行绑定，当开发者获取该用户的 ID 后，调用 SDK 的用户 ID 更新接口完成首次一登账号与三方应用用户账号的绑定。

如果您的应用采用手机号码作为唯一用户 ID 标识，可调用一登 SDK 有 phone 登录接口，直接完成授权绑定过程。当然，如果你的账号体系不依托于手机号码或用户使用 SuperID 登录时尚未获取用户手机号，直接调用一登 SDK 无 Phone 登录接口。

API接口（获取一登账号无 phone 登录 VC）

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

API接口（获取一登账号有 phone 登录 VC）

```
- (instancetype)obtainLoginViewControllerWithPhoneNumber:(NSString *)phoneNumber error:(NSError **)error;
```

具体工程入口的代码如下所示：

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

    NSError *error = nil;

    //获取一登登录VC，如调用有phone接口，在参数phoneNumber中传入手机号码即可。
    id SIDLoginViewController = [superIdSdk obtainLoginViewControllerWithError:&error];

    if (SIDLoginViewController) {
        //开发者须采用present方式弹出一登登录VC
        [self presentViewController:SIDLoginViewController animated:YES completion:nil];
    }else{

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

用户登录操作完成事件处理

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)superID:(SuperID *)sender userDidFinishLoginWithUserInfo:(NSDictionary *)userInfo withAppUid:(NSString *)uid error:(NSError *)error;
```

参数解析：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。如果您的账号体系采用手机号码作为用户唯一标识 ID，可解析返回 Json 中的手机号码作为用户 ID 使用。uid 为当前用户的授权绑定一登账号的应用 ID，如用户是第一次在应用中使用一登登录，将返回随机用户 ID。当用户登录成功，error 为 nil。当用户登录失败，error 不为 nil。

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，error 参数会返回相应的错误代码，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。

5.4 应用用户 ID 更新接口说明

API 接口的如下所示

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

更新事件完成操作：更新应用用户 ID 事件完成操作

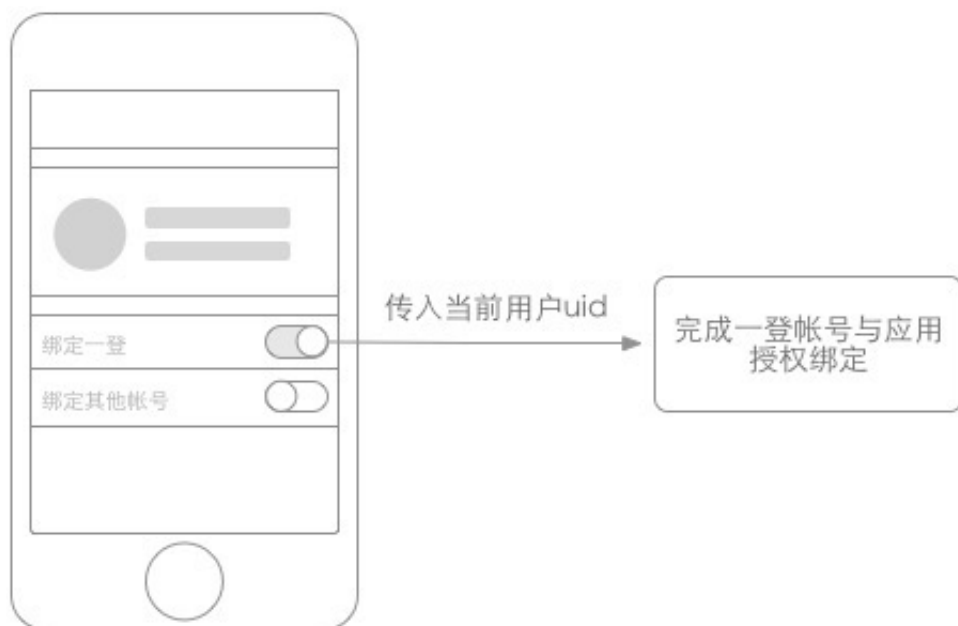
当一登 SDK 完成用户 ID 更新到用户一登账号的事件完成后，一登将执行如下方法通知开发者。

```
- (void)superID:(SuperID *)sender updateAppUidResponse:(SIDUserUpdateResponseState)state;
```

参数解析：回调参数 state 包含在该接口有四种状态：SIDUpdateAppUidSucceed（更新成功），SIDUidHasExist（此 App Uid 已被其他一登账号绑定），SIDUserNoAuthToSuperID（用户解除绑定或尚未绑定一登账号，无法更新），SIDUserAccessExpired（用户授权已过期，无法更新，出现此情况如用户长时间不使用应用导致 Token 过期）。

注明：开发者可根据相应的事件进行后续操作。建议开发者在更新 Uid 时，可添加前缀已区分随机 AppUid 和你账号体系的 AppUid。如：SuperID_XXXXX

场景2：当用户处于为登录应用状态（开发者已知当前 AppUid）。此时直接调用一登授权绑定接口，直接完成账号关联，用户再次登录应用时即可直接返回你账号体系的 AppUid。



5.5 账号授权绑定接口

有手机号码绑定：开发者应用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，开发者需要传入开发者应用用户的 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）三项内容以及其他内容。若您的用户信息中不存在以上项目，可将此传参设置为 nil。调用此接口可在开发者应用的按钮触发事件中添加，与一登用户登录方法一致。

在您调用 SIDAAuthViewController 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（SuperIDDelegate），并在当前 VC 声明 SDK 单例对象和设置委托对象，方法与一登登录接口相同。

API 接口的如下所示（获取一登账号绑定 VC）

```
- (instancetype)obtainAuthViewControllerWithUid:(NSString *)uid phoneNumber:(NSString *)phoneNumber appUserInfo:(NSDictionary *)userInfo error:(NSError **)error;
```

具体工程示例代码如下所示：

```

- (IBAction)userAuthSuperID:(id)sender {

    //生成用户个人账户信息的字典：其中key: name, email, avatar这三项需跟demo所示一致：
    NSMutableDictionary *userInfo = [[NSMutableDictionary alloc] initWithObjectsAnd
    Keys:@"xxx",@"name",@"xxxx@gmail.com",@"email",@"http://xxxxxx",@"avatar", nil];

    //调用授权绑定的VC需传入用户的账号Uid，由于Demo无独立的三方账号体系，采用随机的时间戳作为Uid
    做Demo示例用
    NSError *error = nil;
    id SIDAAuthViewController = [[SuperID sharedInstance]obtainAuthViewControllerWith
    hUid:@"用户的AppUid" phoneNumber:@"用户的手机号码" appUserInfo:@"用户的应用账户信息" error
    r:&error];

    if (SIDAuthViewController) {

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

    }else{

        NSLog(@"SuperIDAuthView Error =%ld,%@",(long)[error code],[error localized
        Description]);
    }
}

```

注明：用户信息存入的 `userinfo` 为 `NSDictionary` 对象，头像、昵称、邮箱信息的字段名（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。

无手机号码绑定：开发者应用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。

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。

用户授权绑定一登账号操作完成将执行如下回调方法：

```

- (void)superID:(SuperID *)sender userDidFinishAuthAppWithUserInfo:(NSDictionary *
)userInfo withAppUid:(NSString *)uid error:(NSError *)error;

```

参数解析：`userInfo`为用户一登账号的账户信息以及成功绑定应用的用户 `uid`，开发者可根据需要进行选取使

用。当用户授权成功，error 为 nil。当用户授权不成功，error 不为 nil。

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。

5.6 用户授权状态查询

查询绑定状态：查询当前应用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或更改UI界面的显示。

API接口的如下所示（一登用户绑定状态查询）

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

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。需在主调方 VC 中设置一登 SDK 的委托对象，方法同上。

用户绑定状态查询事件完成将执行如下回调方法：

```
- (void)superID:(SuperID *)sender queryCurrentUserAuthorizationStateResponse:(SIDUserAuthorizationState)state;
```

参数解析：回调参数 state 包含三种状态：SIDUserHasAuth（uid已与一登账号绑定），SIDUserNoAuth（uid与一登账号尚未绑定），SIDQueryUserAuthorizationFail（当出现网络问题等异常状况导致的查询失败）。

6 用户应用账户信息更新

更新用户信息：更新应用用户个人信息

当用户在应用中的更改了其个人信息，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)updateAppUserInfo:(NSDictionary *)userInfo;
```

更新信息事件完成操作：更新应用用户个人信息事件完成操作

```
- (void)superID:(SuperID *)sender updateAppUserInfoStateResponse:(SIDUserUpdateResponseState)state;
```

参数解析：回调参数 `state` 包含三种状态：`SIDUpdateUserInfoSucceed`（更新成功），`SIDUpdateAppUserInfoFail`（更新失败,如出现网络问题等），`SIDUserNoAuthToSuperID`（用户解除绑定或尚未绑定一登账号，无法更新），`SIDUserAccessExpired`（用户授权已过期，无法更新）。开发者可根据相应的事件对用户进行后续引导操作。

7 用户解除授权

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,需要在对应的 VC 内设置一登 SDK 的委托和声明对象。

API 接口的如下所示（获取一登账号绑定的 View）

```
- (void)userCancelAuthorization;
```

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。

一登用户解绑事件操作完成将执行如下回调方法：

```
- (void)superID:(SuperID *)sender userDidFinishCancelAuthorization:(NSError *)error;
```

参数解析：当用户解绑操作成功，`error` 为 `nil`。当用户解绑操作不成功，`error` 不为 `nil`。

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除一登账号与开发者应用的用户账号的绑定时，一登用户解绑事件执行失败。

8 登出操作：不解除账号授权绑定

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

```
- (void)appUserLogoutCurrentAccount;
```

9 失败事件通知

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应用
- **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** 无法使用表情功能，用户需要先绑定一登账号（权限限制）
- **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
- **error code:-1015** 表示当前用户一登账号已处于冻结状态，无法进行操作