

人人网开放平台iOS SDK使用说明

一、产品特点

- 一键图片上传：提供了仅仅只需一条语句的调用即可完成的图片上传功能，并包括了完整的流程和界面显示。
- 常用功能，简化调用：查询用户信息、好友列表等常用功能的调用过程进行了再封装，大幅简化了开发者传参和处理返回数据的工作。
- 简约接口，轻松接入：针对原版本的接口设计进行了大幅度的合并、优化和再设计，即使是最复杂的功能的调用过程都变得更简单，更明晰，学习成本更低。

二、SDK 功能介绍

接口列表

类型	功能	方法名
登录、授权与登出	授权页面方式登录	-(void)authorizationWithPermisson:(NSArray *)permissions andDelegate:(id<RenrenDelegate>)delegate
	用户名密码方式登录	-(void)passwordFlowAuthorizationWithParam:(ROPasswordFlowRequestParam *)param andDelegate:(id<RenrenDelegate>)delegate;
	登出	-(void)logout:(id<RenrenDelegate>)delegate;
一键发布	一键上传照片	-(void)publishPhotoSimplyWithImage:(UIImage *)image caption:(NSString *)caption;
常用功能	获取用户信息	-(void)getUsersInfo:(ROUserInfoRequestParam *)param andDelegate:(id<RenrenDelegate>)delegate;
	获取好友列表	-(void)getFriends:(ROGetFriendsRequestParam *)param andDelegate:(id<RenrenDelegate>)delegate;
	获取好友详细信息列表	-(void)getFriendsInfo:(ROGetFriendsInfoRequestParam *)param andDelegate:(id<RenrenDelegate>)delegate;
	获取相册信息	-(void)getAlbums:(ROAlbumsInfoRequestParam *)param andDelegate:

		(id<RenrenDelegate>)delegate;
	上传照片	-(void)uploadPhoto: (ROUploadPhotoRequestParam *)param andDelegate: (id<RenrenDelegate>)delegate;
	新建相册	-(void)createAlbum: (ROCreateAlbumRequestParam *)param andDelegate: (id<RenrenDelegate>)delegate;
通用方法	通用API接口调用	-(RORequest *)requestWithParams: (NSMutableDictionary *)params andDelegate: (id <RenrenDelegate>)delegate;
	Widget Dialog调用	-(void)dialog:(NSString *)action andParams:(NSMutableDictionary *)params andDelegate: (id <RODialogDelegate>)delegate;

代理方法列表

代理方法	说明
- (void)renren:(Renren *)renren requestDidReturnResponse: (ROResponse*)response;	API接口请求成功时调用，返回服务器给出的 请 求结果数据对象。（一一键上传和登录登出 相关 功能不适用）
- (void)renren:(Renren *)renren requestFailWithError: (ROError*)error;	API接口请求失败时调用，返回表示错误原因的 错误对象。（一一键上传和登录登出相关功 能不 适用）
- (void)renrenDialogDidCancel: (Renren *)renren;	Dialog页面被用户或Renren对象取消并关闭 时调 用。（仅 Dialog 功能适用）
- (void)renrenDidLogin: (Renren *)renren;	登录成功时调用。（仅登录相关功能适用）
- (void)renrenDidLogout: (Renren *)renren;	登出成功时调用。（仅登出功能适用）
- (void)renren:(Renren *)renren loginFailWithError: (ROError*)error;	登录失败时调用，返回 返回表示错误原因的错 误对象。（仅登录相关功能适用）

三、使用说明

1. 请将 JSON 和 Renren 两个文件夹复制到你的 Xcode 工程下
2. 在 Xcode 下右键点击你的 Xcode 工程代码 Group，从弹出的菜单中选择 Add -> Existing Files，从而加入你需要的代码。
3. 如果需要设置包含路径，请正确设置。（一般不需要）
4. 在你的代码中，需要时包含调用 SDK 的头文件：
`#import "ROConnect.h"`

使用 iOS SDK 提供的页面授权方式。

1. 首先，打开 Renren 文件夹下找到 Renren.h 文件：

```
#define kAPP_ID      @"YOUR APP ID"  
#define kAPI_Key     @"YOUR API KEY"  
#define kSecret      @"YOUR SECRET"
```

将上面宏定义的内容配置为你在人人开发平台上申请的应用信息。

2. 实例化一个 Renren 对象，你可以在应用程序委托 application:didFinishLaunchingWithOptions: 方法或是在视图加载的 viewDidLoad 方法时，
创建一个 Renren 对象。

```
Renren *renren = [Renren sharedRenren];
```

3. [renren isValidSession]

这个方法可以判断当前用户是否登录状态

4. [renren authorizeWithPermission:nil andDelegate:self];

通过这个方法，用户可以进行登录验证授权

authorizeWithPermission:[授予权限] 如 publish_feed photo_upload 等

参数默认可以不写，如有多个权限的话，可以这样

```
NSArray *permissions=[NSArray arrayWithObjects:@"photo_upload", "publish_feed", nil];  
[renren authorizeWithPermission:permissions andDelegate:self];
```

5. 你还需要在你的应用中实现 Renren 的代理 RenrenDelegate，主要实现下面两个方法：

```
/**
```

```
 * 授权登录成功时被调用，第三方开发者实现这个方法
```

```
* @param renren 传回代理授权登录接口请求的 Renren 类型对象。
*/
- (void)renrenDidLogin:(Renren *)renren;

/**
 * 授权登录失败时被调用，第三方开发者实现这个方法
 * @param renren 传回代理授权登录接口请求的 Renren 类型对象。
 */
- (void)renren:(Renren *)renren loginFailWithError:(ROError*)error;
```

使 iOS SDK 提供的用户名密码授权方式

1.请参照《使用 iOS SDK 提供的页面授权方式》的 1，2，3 步

2.创建 ROPasswordFlowRequestParam 的对象并初始化

```
ROPasswordFlowRequestParam *requestParam = [[[ROPasswordFlowRequestParam alloc] init] autorelease];
```

设置 requestParam 中的 userName 和 passWord 属性，如下

```
requestParam.userName = @"user";
requestParam.passWord = @"123456";
```

设置权限的种类(只获得默认权限可以不用设置此属性)

```
requestParam.scope = @"operate_like";
```

调用下面的方法完成授权

```
[renren passwordFlowAuthorizationWithParam:requestParam andDelegate:self];
```

3.实现 RenrenDelegate 中代理方法，完成授权成功和失败的处理，可以参照《使用 iOS SDK 提供的页面授权方式》的第 5 步

使用 iOS SDK 调用人人 API 接口 （以取得用户信息为例）

1.请参照《使用 iOS SDK 提供的页面授权方式》的 1，2。

2.完成授权后，创建 RUserInfoRequestParam 的对象并初始化

```
RUserInfoRequestParam *requestParam = [[[RUserInfoRequestParam alloc] init] autorelease];
```

3.设置 requestParam 中的 fields(返回用户那些字段信息)属性，如下

```
requestParam.fields = @"uid,name,sex";
```

调用下面的方法完成授权

```
[renren getUserInfo:requestParam andDelegate:self];
```

4.需要实现 RenrenDelegate 中代理方法

```
/**
 * 接口请求成功，第三方开发者实现这个方法
 * @param renren 传回代理服务器接口请求的 Renren 类型对象。
 * @param response 传回接口请求的响应。
 */
- (void)renren:(Renren *)renren requestDidReturnResponse:(ROResponse*)response;

/**
 * 接口请求失败，第三方开发者实现这个方法
 * @param renren 传回代理服务器接口请求的 Renren 类型对象。
 * @param response 传回接口请求的错误对象。
 */
- (void)renren:(Renren *)renren requestFailWithError:(ROError*)error;
```

使用 iOS SDK 完成一键上传照片

1.请参照《使用 iOS SDK 提供的页面授权方式》的 1，2。

2.完成授权后，创建 Image 的对象和描述，

```
UIImage* image = [UIImage imageNamed:@"test.png"];
NSString *caption = @"这是一张测试图片";
```

3.[renren publishPhotoSimplyWithImage:image caption:caption];
调用上面的方法即可完成上传照片的功能。

4.需要实现 RenrenDelegate 中代理方法

```
/**
 * 接口请求成功，第三方开发者实现这个方法
 * @param renren 传回代理服务器接口请求的 Renren 类型对象。
 * @param response 传回接口请求的响应。
 */
- (void)renren:(Renren *)renren requestDidReturnResponse:(ROResponse*)response;

/**
 * 接口请求失败，第三方开发者实现这个方法
 * @param renren 传回代理服务器接口请求的 Renren 类型对象。
 * @param response 传回接口请求的错误对象。
 */
```

```
- (void)renren:(Renren *)renren requestFailWithError:(ROError*)error;
```

使用 iOS SDK 提供的 widget dialog （以发布状态为例）

1.请参照《使用 iOS SDK 提供的页面授权方式》的 1， 2。

2.完成授权后，创建请求 widget dialog 的字典

```
NSMutableDictionary *params=[NSMutableDictionary dictionaryWithObjectsAndKeys:
    @"http://dev.renren.com/",@"url",
    @"人人网开放平台",@"name",
    @"访问我们",@"action_name",
    @"http://dev.renren.com/",@"action_link",
    @"来自人人网 iOS SDK",@"description",
    @"欢迎使用人人网 SDK!",@"caption",
    @"http://hdn.xnimg.cn/photos/hdn421/20090923/1935/head_1Wmz_19242g019116.jpg",@"image",
    nil];
```

3.[renren dialog:@"feed" andParam:params andDelegate:self];

调用上面的方法即可完成发布状态的功能。

4.需要实现 RenrenDelegate 中代理方法

```
/**
 * 接口请求成功，第三方开发者实现这个方法
 * @param renren 传回代理服务器接口请求的 Renren 类型对象。
 * @param response 传回接口请求的响应。
 */
- (void)renren:(Renren *)renren requestDidReturnResponse:(ROResponse*)response;

/**
 * 接口请求失败，第三方开发者实现这个方法
 * @param renren 传回代理服务器接口请求的 Renren 类型对象。
 * @param response 传回接口请求的错误对象。
 */
- (void)renren:(Renren *)renren requestFailWithError:(ROError*)error;
```

使用 iOS SDK 退出授权状态

1 在授权以后，通过调用下面的方法退出授权状态。

```
[renren logout:self];
```

2. 需要实现 RenrenDelegate 中代理方法

```
/**
 * 用户登出成功后被调用 第三方开发者实现这个方法
 * @param renren 传回代理登出接口请求的 Renren 类型对象。
 */
- (void)renrenDidLogout:(Renren *)renren;
```

五、功能示范

• 一键上传图片

实现代码见Renren SDK Demo中的“一键上传图片”功能。



• 使用常用功能接口查询当前登录用户信息

实现代码见 Renren SDK Demo 中的“当前用户信息”功能。



• 使用通用API接口发布状态

实现代码见Renren SDK Demo中的“发布状态”功能。



- 使用Widget Dialog发布自定义新鲜事
- 实现代码见Renren SDK Demo中的“Feed Dialog”功能



人人网 iOS SDK 负责人：夏文海
E-mail:wenhai.xia@renren-inc.com
QQ:86852962