**iOS SDK 环境搭建**

目录

前言 1

1. iOS SDK 下载 2

2. iOS SDK目录结构 2

3. 将iOS SDK文件添加到工程中 2

4. 修改必要的代码 4

4.1 修改工程配置文件 4

4.2 重写AppDelegate 的handleOpenURL和openURL方法 4

4.3 在代码中实现TencentSessionDelegate协议中的方法 5

4.4 初始化iOS SDK API数据对象TencentOAuth。 5

5. 调用SDK登录 6

# 前言

QQ互联SDK（iOS） 封装了QQ登录的登录授权以及大部分OpenAPI，应用只需要修改相应参数，不需要理解验证授权流程，即可快速实现QQ登录功能。

iOS平台上（iPad，iPhone，iPod）的应用，请在[申请](http://connect.opensns.qq.com/apply)appid，appkey后，使用QQ互联提供的iOS SDK。

以前申请过 APPID 类似“QQXXXXXXXX”的开发商，建议重新[申请](http://connect.opensns.qq.com/apply)。为了兼容旧版本的手机QQ，需要增加 URL Scheme，QQ + 十六进制新AppId，不足八位在首部补0。（如 appid=222222 则 scheme=QQ0003640E）

手机QQ SDK说明文档详见iOS SDKAPI使用说明。

本SDK仅适用于移动应用，即基于[使用Implicit\_Grant方式获取Access\_Token](http://wiki.open.qq.com/wiki/website/%25E4%25BD%25BF%25E7%2594%25A8Implicit_Grant%25E6%2596%25B9%25E5%25BC%258F%25E8%258E%25B7%25E5%258F%2596Access_Token)接入的应用。

# 1. iOS SDK 下载

请到[SDK下载](http://wiki.connect.qq.com)页面下载最新版本QQ登录iOS SDK。

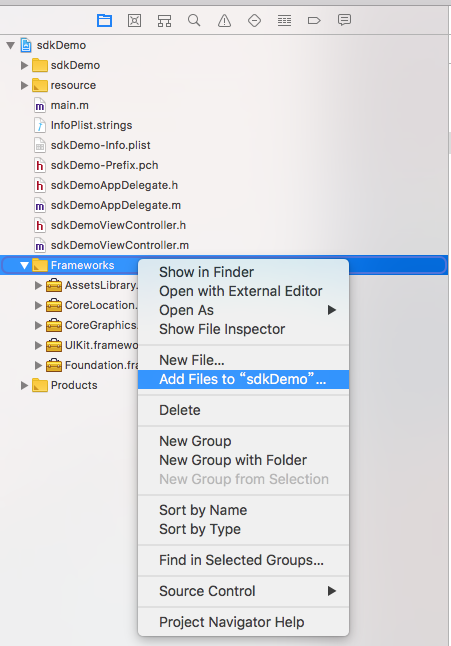
# 2. iOS SDK目录结构

iOS SDK包中打包了iOS SDK的头文件定义和具体实现。

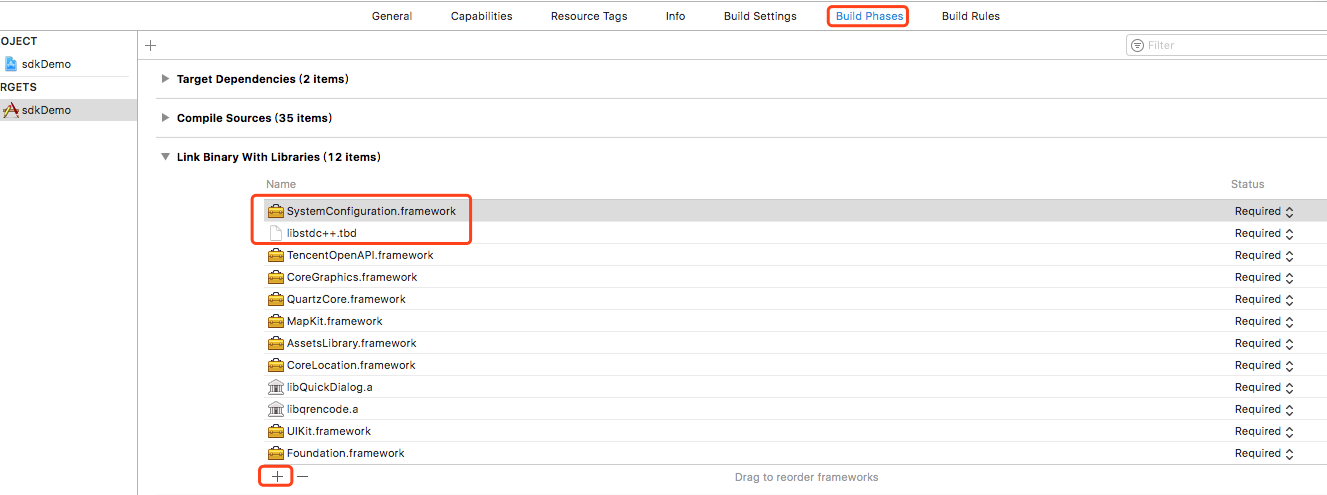
# 3. 将iOS SDK文件添加到工程中

1. 将iOS SDK中的TencentOpenAPI.framework文件拷贝到应用开发的目录下。

然后按下图所示添加到工程中。



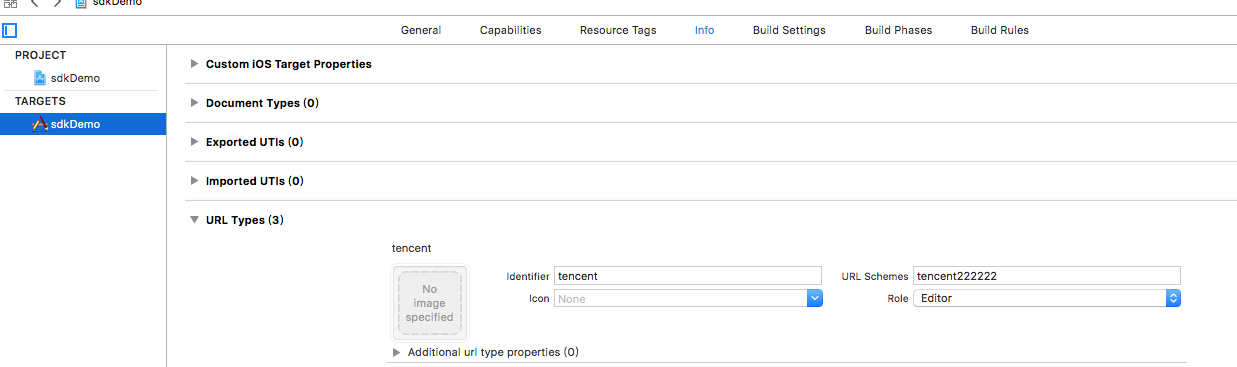
2. 添加SDK依赖的系统库文件。分别是 “SystemConfiguration.framework”.

在Xcode中打开工程配置文件，添加依赖库。

# 4. 修改必要的代码

## 4.1 修改工程配置文件

在XCode中，选择你的工程设置项，选中“TARGETS”一栏，在“info”标签栏的“URL type”添加一条新的“URL scheme”，新的scheme = tencent + appid。如果您使用的是XCode3或者更低的版本，则需要在plist文件中添加。Demo中我们注册的appid是222222。如下图



## 4.2 重写AppDelegate 的handleOpenURL和openURL方法

openURL:

- (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation{

return [TencentOAuth HandleOpenURL:url];

}

handleOpenURL:

- (BOOL)application:(UIApplication \*)application handleOpenURL:(NSURL \*)url{

return [TencentOAuth HandleOpenURL:url];

}

## 4.3 在代码中实现TencentSessionDelegate协议中的方法

具体协议可以参照TencentOpenAPI.framework /Headers中的TencentOAuth.h文件

## 4.4 初始化iOS SDK API数据对象TencentOAuth。

（1） 创建TencentOAuth并初始化其appid，demo为222222。delegate为实现TencentSessionDelegate的对象：

\_tencentOAuth = [[TencentOAuth alloc] initWithAppId:@"222222", andDelegate:self];

这里delegate不能为空  
（2） 初始化redirectURI（这里需要填写注册APP时填写的域名。默认可以不用填写。建议不用填写。demo中注册时的地址是“www.qq.com”）：

\_tencentOAuth.redirectURI = @"www.qq.com";

（3)设置应用需要用户授权的API列表。

(建议如果授权过多的话，可能会造成用户不愿意授权。这里最好只授权应用需要用户赋予的授权。)：

\_permissions = [[NSArray arrayWithObjects:@"get\_user\_info", @"get\_simple\_userinfo", @"add\_t", nil] retain];

# 5. 调用SDK登录

1.登录时，调用TencetnOAuth对象的authorize方法：

[\_tencentOAuth authorize:\_permissions inSafari:NO]；

2. 登录完成后，会调用TencentSessionDelegate中关于登录的协议方法。  
登录成功：

@protocol TencentSessionDelegate <NSObject>

- (void)tencentDidLogin

{

\_labelTitle.text = @"登录完成";

if (\_tencentOAuth.accessToken && 0 != [\_tencentOAuth.accessToken length])

{

//  记录登录用户的OpenID、Token以及过期时间

\_labelAccessToken.text = \_tencentOAuth.accessToken;

}

else

{

\_labelAccessToken.text = @"登录不成功 没有获取accesstoken";

}

}

非网络错误导致登录失败：

@protocol TencentSessionDelegate <NSObject>

-(void)tencentDidNotLogin:(BOOL)cancelled

{

if (cancelled)

{

\_labelTitle.text = @"用户取消登录";

}

else

{

\_labelTitle.text = @"登录失败";

}

}

网络错误导致登录失败：

@protocol TencentSessionDelegate <NSObject>

-(void)tencentDidNotNetWork

{

\_labelTitle.text=@"无网络连接，请设置网络";

}

3. 登录成功后，即可获取到access token和openid。accessToken和 openid保存在TencentOAuth对象中，并且已经本地化存储。可以通过相应的属性方法直接获得。

[\_tencentOAuth accessToken] ;

[\_tencentOAuth openId] ;

[\_tencentOAuth getCachedOpenID] ;

[\_tencentOAuth getCachedToken] ;

**特别提示：**

1.由于登录是异步过程，这里可能会由于用户的行为导致整个登录的的流程无法正常走完，即有可能由于用户行为导致登录完成后不会有任何登录回调被调用。开发者在使用SDK进行开发的时候需要考虑到这点，防止由于一直在同步等待登录的回调而造成应用的卡死，建议在登录的时候将这个实现做成一个异步过程。  
2.获取到的access token具有3个月有效期，过期后提示用户重新登录授权。

3. 第三方网站可存储access token信息，以便后续调用OpenAPI访问和修改用户信息时使用。如果需要保存授权信息，需要保存登录完成后返回的accessToken，openid 和 expirationDate三个数据，下次登录的时候直接将这三个数据是设置到TencentOAuth对象中即可。

获得：

[\_tencentOAuth accessToken] ;

[\_tencentOAuth openId] ;

[\_tencentOAuth expirationDate] ;

设置：

[\_tencentOAuth setAccessToken:accessToken] ;

[\_tencentOAuth setOpenId:openId] ;

[\_tencentOAuth setExpirationDate:expirationDate] ;

4. 建议应用在用户登录后，即调用getUserInfo接口获得该用户的头像、昵称并显示在界面上，使用户体验统一。