**iOS SDK 环境搭建**

目录

[前言 2](#_Toc370737224)

[变更历史 3](#_Toc370737225)

[1. iOS SDK 下载 4](#_Toc370737226)

[2. iOS SDK目录结构 4](#_Toc370737227)

[3. 将iOS SDK文件添加到工程中 4](#_Toc370737228)

[4. 修改必要的代码 12](#_Toc370737229)

[4.1 修改工程配置文件 12](#_Toc370737230)

[4.2 重写AppDelegate 的handleOpenURL和openURL方法 13](#_Toc370737231)

[4.3 在代码中实现TencentSessionDelegate协议中的方法 14](#_Toc370737232)

[4.4 初始化iOS SDK API数据对象TencentOAuth。 14](#_Toc370737233)

[5. 调用SDK登录 15](#_Toc370737234)

# 前言

QQ登录iOS SDK 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。

1.5版本增加 QQAPI 对象，以支持手机QQ的调用。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/%E4%BD%BF%E7%94%A8Implicit_Grant%E6%96%B9%E5%BC%8F%E8%8E%B7%E5%8F%96Access_Token)接入的应用。

# 变更历史

V2.3 修改V2.2版本bug 调整SDK结构

V2.2 支持拉起手机QQ QQ空间结合版进行QZone分享

V2.0.1 增加全新的分享到QZone的接口

V2.0 新增API判断手机QQ版本

新增WPA临时会话

支持分享音频消息

Demo 实现简单调整

分享消息结构化改造

新增系统库Security.framework

v1.8 新增了手机QZone开放平台能力支持

新增了调用腾讯微云的SDK接口

新增了1个发送腾讯微博的接口

v1.5 新增了手机QQ SDK的支持

新增手机QQ SSO 登录实现

v1.4 新增[SendStory空间定向分享接口（sendStory:friendList:）](file:///C:\Users\minren\Documents\My%20RTX%20Files\minren\iOS%20SDK%201.8使用说明.doc#SendStory接口调用示例)

新增[增量授权机制](file:///C:\Users\minren\Documents\My%20RTX%20Files\minren\iOS%20SDK%201.8使用说明.doc#使用增量授权)

v1.3 新版SDK以framework形式发布

新增1个设置QQ头像接口（setUserHeadpic）

新增2个微博相关接口（matchNickTips，getIntimateFriends）

新增2个会员相关接口（getVipInfo，getVipRichInfo）

# 1. iOS SDK 下载

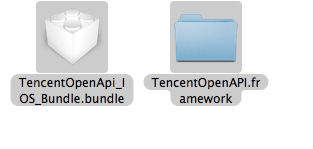
请到[SDK下载](http://wiki.connect.qq.com/sdk%E4%B8%8B%E8%BD%BD)页面下载最新版本QQ登录iOS SDK。

# 2. iOS SDK目录结构

iOS SDK包中带有两个文件：

1. TencentOpenAPI.framework打包了iOS SDK的头文件定义和具体实现。

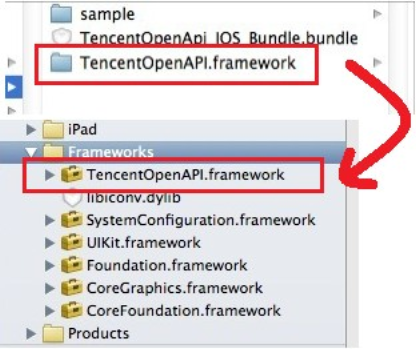
2. TencentOpenApi\_iOS\_Bundle.bundle 打包了iOS SDK需要的资源文件。



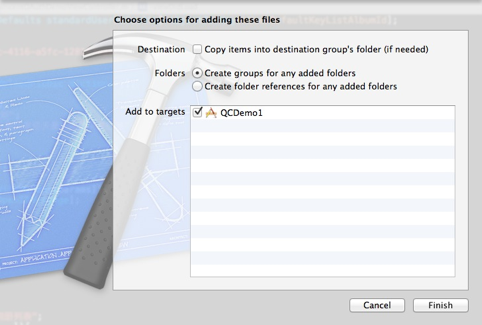
# 3. 将iOS SDK文件添加到工程中

1. 将iOS SDK中的TencentOpenAPI.framework和TencentOpenApi\_IOS\_Bundle.bundle文件拷贝到应用开发的目录下。

然后将TencentOpenAPI.framework从SDK的保存目录拖拽到工程导航视图（project navigator）中的Frameworks虚拟目录下。

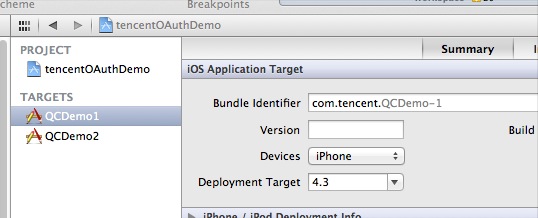


2. 在弹出的对话框中勾选“Create groups for any added folders”，去掉“copy items into destination group’s folder（if needed）”，在Add to targets中选择要加入SDK的target之后点击finish。完成之后就将iOS SDK的framework文件加入了开发工程中。

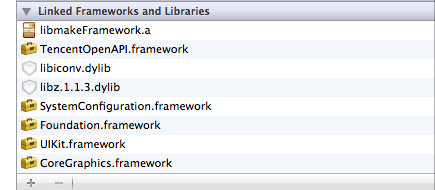


3. 添加SDK依赖的系统库文件。分别是”Security.framework”, “libiconv.dylib”，“SystemConfiguration.framework”，“CoreGraphics.Framework”、“libsqlite3.dylib”、“CoreTelephony.framework”、“libstdc++.dylib”、“libz.dylib”。

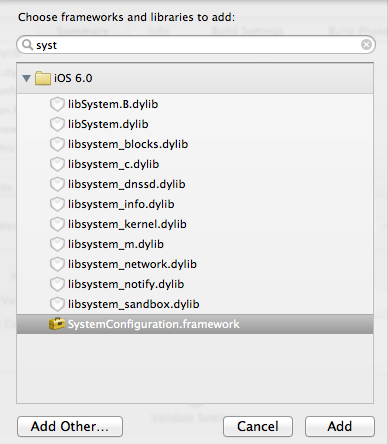
在Xcode中打开工程配置文件，选择“summary”一栏。

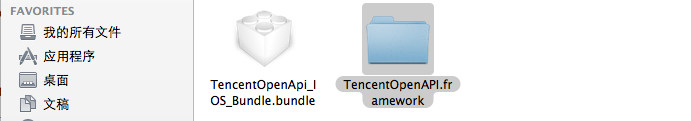


4. 在“summary”中选择“Linked Frameworks and Libraries”一栏，点击“+”图标。

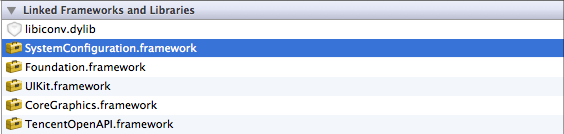


5. 直接在默认库文件中选择后点击“Add”，下图以添加“SystemConfiguration.framework”为例：

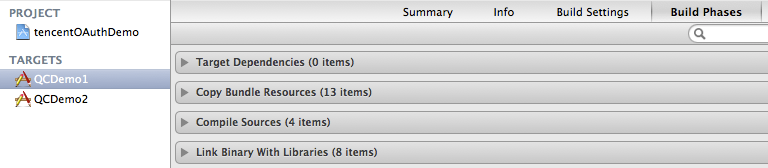




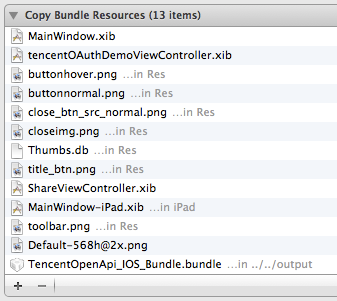
6. 返回后看到“SystemConfiguration.framework”已经在“Linked Frameworks and Libraries”中出现。



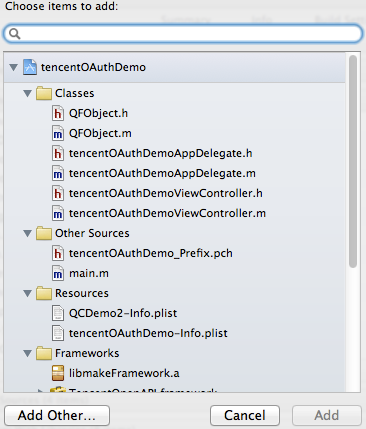
7. 在Xcode中打开工程配置文件，选择“Build Phases”一栏。

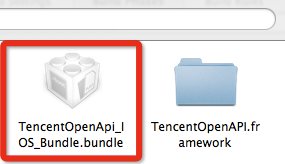


8.在“Build Phases”中选择展开“Copy Bundle Resources”一栏，并点击“+”图标

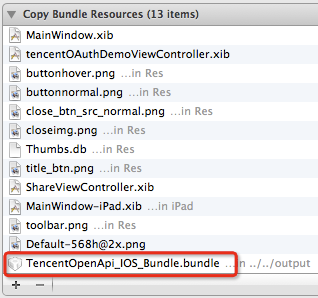


9. 选择“Add Other...”，进入iOS SDK文件所在目录，选择TencentOpenApi\_IOS\_Bundle.bundle，点击回车或者点击“Open”。



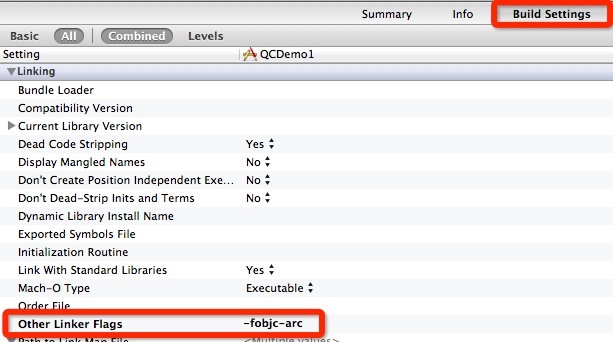


10. 返回后看到TencentOpenApi\_IOS\_Bundle.bundle已经在“Copy Bundle Resources”中出现。



11. 修改必要的工程配置属性。

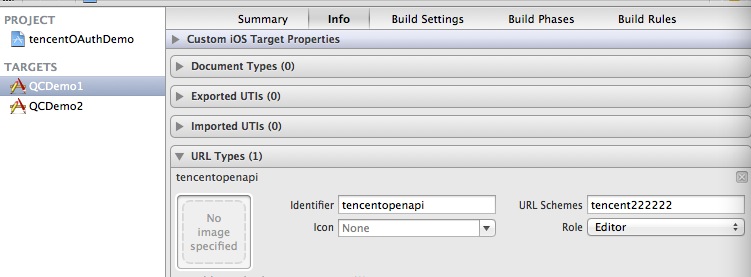
在工程配置中的“Build Settings”一栏中找到“Linking”配置区，给“Other Linker Flags”配置项添加属性值“-fobjc-arc”。



# 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。可以通过相应的属性方法直接获得。

[\_tencentOAuth accessToken] ;

[\_tencentOAuth openId] ;

**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。