Top IOS Sdk 使用文档（v0.2）

Author：放翁（文初）

Email：[fangweng@taobao.com](mailto:fangweng@taobao.com)

Date：2012/11/23

旺旺群：590851148 加入消息提示（top ios sdk开发）

概要：

无线开发者在淘宝应用市场越来越多，而开发过程中往往会遇到很多授权及服务调用通信方面的问题，因此淘宝开放平台提供了IOS SDK来帮助开发者快速开发业务（无需关注授权，服务多线程调用，异步处理返回等问题）。同时由于无线端的安全问题比较突出，开放平台通过SDK配合服务端升级了授权和通信交互协议，提高了无线应用使用卖家相关的服务权限，便于无线开发者深度开发无线卖家应用。

当前开放平台上无线类应用分成两类：无线买家应用，在线订购应用（无线应用类目），后者主要是无线卖家类应用。两类应用的差别是服务使用权限不同，后者更偏向于卖家服务类应用的开发。

使用说明：

场景一：申请应用

进入<http://my.open.taobao.com> ，选择创建应用



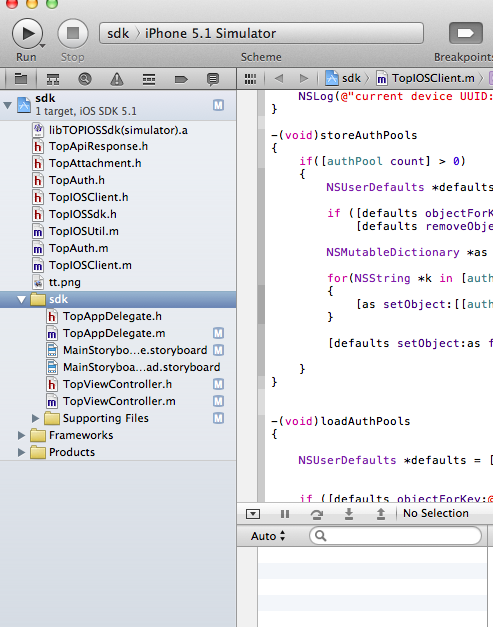
如果是无线买家类应用，这里可以直接选择无线买家应用，如果是无线卖家应用，则选择在线订购类应用。如果选择的是在线订购应用，请参照下图，将最下面的类目选择为无线应用的选项。



注意，这里有一个地址非常重要，回调页面URL（暂时可以先填写一个地址作为基础，后续修改），这个后续对无线授权有非常大的影响。

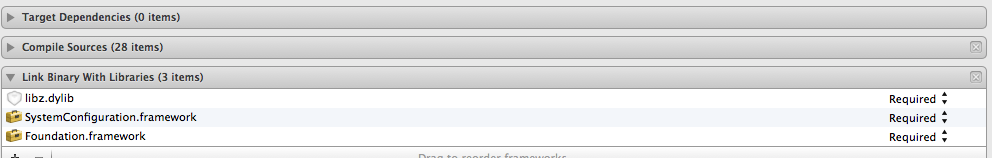
场景二 使用IOS sdk

当前TOP IOS sdk提供的是静态链接库的方式给外部开发者，提供的是一系列接口（几个h结尾的文件）和两个链接库文件（.a结尾的文件，其中名字较长的是用于模拟器开发使用）。使用静态链接库的方式，打开应用，右键选择add Files to …，将模拟器静态文件和接口文件加入到项目中，如下图（结构可以调整的更好一些）

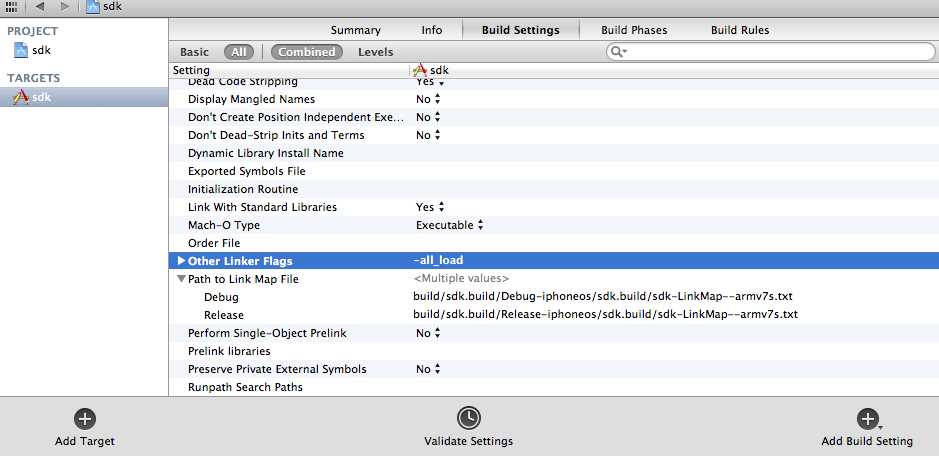


同时需要修改项目Target的几个配置：

1.增加libz.dylib和SystemConfiguration.framework。



2.修改一下Build Settings



接着大家就可以使用这个静态链接库了。

场景三 HelloWorld

很多时候跑起来HelloWorld，就能够给大家减少非常多的问题。因此这里将会通过一个应用的构建和运行来让开发者对无线SDK的集成有一个大致的概念。因此在整个压缩包里面还包含了一个sdk的demo项目，这个项目只需要少量修改就可以运行起来。

首先选择TopAppDelegate.m这个文件，里面有如下两个方法：

//这个方法是在应用启动的时候，初始化你的ios sdk（注册了一个appkey的Ios client实例，只要调用过registerIOSClient，后面就可以通过getIOSClientByAppkey来获得实例授权和调用服务），需要做的就是修改蓝色部分的内容。appkey换成你刚才申请的应用的appkey，appsecretcode替换成这个appkey对应的secretcode，callbackurl请非常小心的修改为你自己定义的一个native地址（暂时你可以不修改，但是你需要去修改app的信息中回调地址url也为appcallback://），注意这个native必须和刚才前面提到的应用回调页面url设置完全保持一致（什么叫native地址，就是应用的外部交互地址，比如你设置appcallback://作为你的交互地址，那么如果你在safari浏览器里面输入appcallback://name=cenwenchu&age=33，此时应用就会和一台服务器一样接收到请求和参数，那应用什么地方会处理这个请求呢，就请看下面一个函数）

- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions

{

TopIOSClient \*topIOSClient = [TopIOSClient registerIOSClient:@"appkey" appSecret:@"appsecretcode" callbackUrl:@"appcallback://" needAutoRefreshToken:TRUE];

[TopAppConnector registerAppConnector:@" appkey " topclient:topIOSClient];

return YES;

}

//这个函数是应用处理外部请求的函数，url就是外部的url信息，这里注意，因为当前外部请求默认都是授权处理，因此下面代码中就是对授权的处理，第一句话就是获取某一个appkey对应的IOS sdk实例（这个客户端支持多个appkey实例），因此修改那个蓝色的appkey为你的真实appkey。下面一句话不用修改，默认处理授权url返回。

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

{

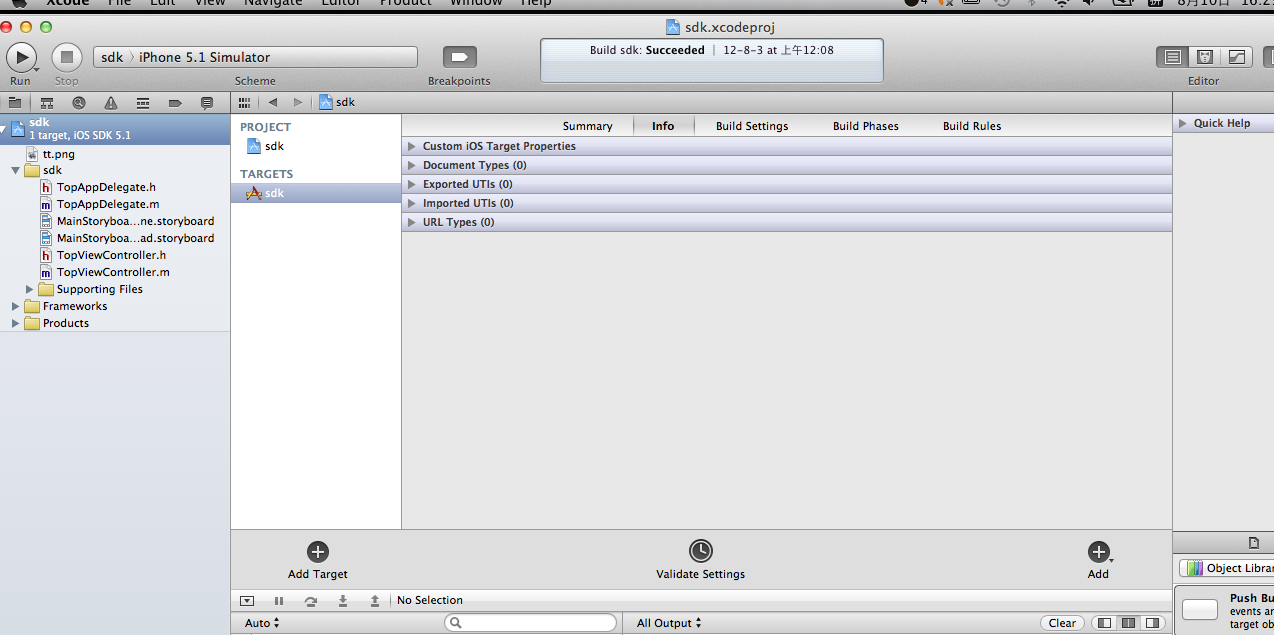
TopAppConnector \*appConnector = [TopAppConnector getAppConnectorbyAppKey:@" appkey "];

[appConnector receiveMessageFromApp:[url absoluteString]];

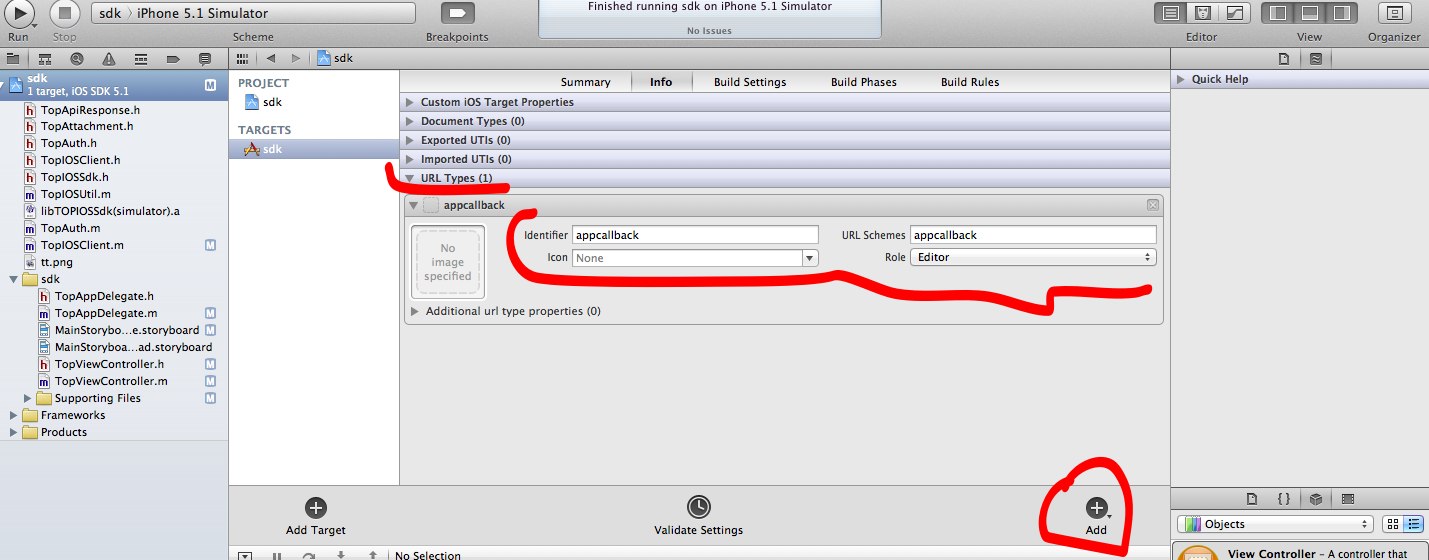
return YES;

}

接着，就要设置native地址如何被识别的关键步骤，选择sdk项目本身，然后选择target，再选择info这个tab，就可以看到下图



选择URL types，然后再选择右下角的＋，填入刚才我们的地址协议部分：appcallback，然后保存。至此如果你修改完open.taobao.com中的应用回调地址后，授权设置就完毕了。（总结一下，注册appclient的时候的回调地址和app再open.taobao.com保持一致，然后将协议部分设置到应用的url types）



然后开始修改业务逻辑代码比较重的TopViewController.m文件，这个文件主要负责响应前台界面指令，做以下几件事情：发起授权，调用普通API，调用TQL，调用图片上传类的服务。下面按照这个顺序去介绍：

授权：(这个方法只需要替换掉appkey为正式的appkey即可)

- (IBAction)authAction:(id)sender {

TopIOSClient \*iosClient = [TopIOSClient getIOSClientByAppKey:@"appkey"];

[iosClient auth:self cb:@selector(authCallback:)];

}

上面这个函数响应授权按钮发起的指令，首先还是通过标准的方式获取客户端，注意将appkey替换成你前面合法设置的appkey，然后调用授权（iosclient的auth方法），方法有两个参数，self作为第一个参数，@selector(authcallback：)作为第二个参数，其实就是授权成功或者失败的回调函数，这里是通过SEL的方式来回调，因此需要给出归属对象和对象的方法，这里给的是当前TopViewController这个对象（self关键字来替代），方法就是TopViewController中的authCallback这个方法，可以搜索这个类看到具体的代码。（授权方法详细描述参考api接口文件中的注释）

普通服务调用：(这个方法只需要替换掉appkey为正式的appkey即可)

- (IBAction)sendRequestAction:(id)sender {

//……(前面都是准备参数的代码)

TopIOSClient \*iosClient =[TopIOSClient getIOSClientByAppKey:@"appkey"];

NSString\* sessionkey = [iosClient api:@"GET" params:params target:self cb:@selector(showApiResponse:) userId:uid needMainThreadCallBack:true];

//如果想要取消此请求的结果返回，则使用[iosClient cancel:sessionkey];

……

api的调用方式很简单，获取出appkey对应的client实例，然后调用它的api方法，第一个参数是http请求方式（淘宝开放平台对与更新操作要求必须用post方式），第二个参数是业务参数集合，第三，第四个参数是服务返回后回调应用处理的函数，第五个参数表明是对那一个用户做服务调用（因为api调用过程中如果操作用户数据必须有授权，当你授权完成以后，sdk内部会保存授权信息，此时你只需要通过传入用户id，sdk就会在请求的时候带上用户的授权调用服务），最后一个参数用来告诉sdk是否要用主线程阻塞模式回调，因为部分情况下返回结果直接要渲染到view上，那么必须通过主线程阻塞模式来做，如果不是渲染view，可以选择非主线程模式。至此看以下这个类的showApiResponse你就能只到返回怎么处理了，接口细节参看api接口文件中的注释。

TQL调用：

TQL是另一种方式的服务调用，可以支持服务并联和串联调用，具体参看：<http://open.taobao.com/doc/detail.htm?id=806> ,调用方式和普通api调用非常类似，唯一的差异是它调用的是client的tql方法，前者是api方法。

下面代码主要用了同步调用模式，注释掉的是异步模式，异步模式和上面的普通调用一样，开发者根据自己需求来选择异步还是同步模式。

- (IBAction)tqlRequest:(id)sender {

TopIOSClient \*iosClient = [TopIOSClient getIOSClientByAppKey:@"appkey"];

NSMutableDictionary \*params = [[NSMutableDictionary alloc] init];

NSString \*uid = [userId text];

[params setValue:@"select uid,nick,sex from user" forKey:@"ql"];

//[params setValue:@"select num\_iid,title,nick,pic\_url,cid,price,type,delist\_time,post\_fee from item where num\_iid=15004213634" forKey:@"ql"];

//[iosClient tql:@"GET" params:params target:self cb:@selector(showApiResponse:) userId:uid needMainThreadCallBack:true];

TopApiResponse \* response = [iosClient tql:@"GET" params:params userId:uid];

[responseContentView setText:[response content]];

}

Schedule调用：

Schedule提供的是一种异步定时任务服务，具体使用场景和使用方法参看：<http://open.taobao.com/doc/detail.htm?spm=0.0.0.21.da03e0&id=922> ，当前SDK只提供同步接口（因为Schedule本身就是异步，因此无需提供异步模式）

* (IBAction)scheduleRequest:(id)sender {

TopIOSClient \*iosClient = [TopIOSClient getIOSClientByAppKey:@"appkey"];

NSMutableDictionary \*params = [[NSMutableDictionary alloc] init];

NSString \*uid = [userId text];

[params setValue:@"select num\_iid,title,nick,pic\_url,cid,price,type,delist\_time,post\_fee from item where num\_iid=17795332215" forKey:@"ql"];

[params setValue:@"2012-10-10 13:00:20" forKey:@"schedule"];

[params setValue:@"http://10.13.201.118:7777" forKey:@"callbackurl"];

response = [iosClient schedule:@"GET" params:params userId:uid];

if ([response error])

NSLog(@"schedule api call result : %@",[response error]);

else

NSLog(@"schedule api call result : %@",[response content]);

}

图片上传类服务：

这类服务就是一种api的调用服务，只是因为内部包含了图片等二进制内容，需要采用multipart方式发起请求。这里要求将图片之类的二进制文件封装到一个TopAttachment的类中，然后作为参数放入params里面发起请求。其他没有什么差别，注意将appkey替换成自己的appkey。

注意如果是文件类型必须初始化为Attachment放入params，这样sdk可以自动识别，如果是图片注意他的名字后缀需要符合规范，gif,jpg,jpeg,png,bmp，具体参见文档。

- (IBAction)uploadPicAction:(id)sender {

NSString \*uid = [userId text];

NSMutableDictionary \*params = [[NSMutableDictionary alloc]init];

[params setObject:@"taobao.item.img.upload" forKey:@"method"];

[params setObject:@"17795332215" forKey:@"num\_iid"];

NSURL \*url = [[NSURL alloc] initWithString:@"http://upload.wikimedia.org/wikipedia/commons/d/de/POL\_apple.jpg"];

NSData \*image\_data = [NSData dataWithContentsOfURL:url];

TopAttachment \*image = [[TopAttachment alloc]init];

[image setData:image\_data];

[image setName:@"apple.jpg"];

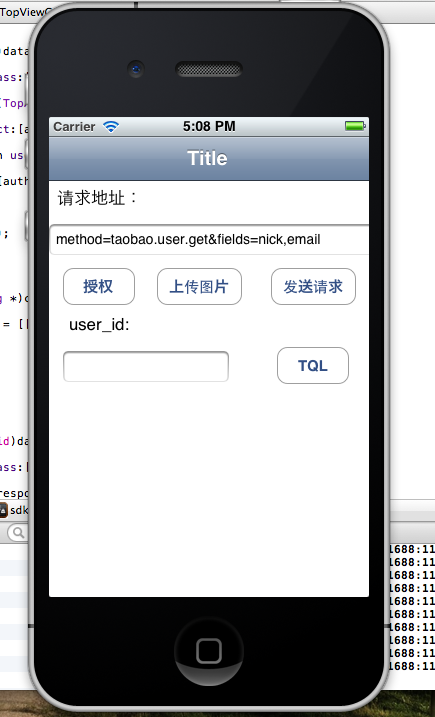
[params setObject:image forKey:@"image"];

TopIOSClient \*iosClient =[TopIOSClient getIOSClientByAppKey:@"appkey"];

[iosClient api:@"POST" params:params target:self cb:@selector(showApiResponse:) userId:uid needMainThreadCallBack:true];

}

至此，如果你确认修改完了所有appkey和上文说道的地方，然后将静态类库（模拟器版本）和接口引入到项目，你就可以点击运行了。



首先你可以点击授权那个按钮，会弹出浏览器，然后授权，请填入自己的淘宝id（注意，如果你是买家类应用没有上线的时候会默认绑定你app注册时候的淘宝帐号作为测试帐号，其他帐号不能使用，如果要增加更多帐号测试，去应用页面选择管理授权，就可以增加了），然后授权成功会返回到上一个页面。（此时授权已经保存在客户端内部了，会帮持久化到客户端中，同时如果在创建应用客户端实例的时候可以设置是否自动刷新授权），此时你可一看到user\_id已经有内容了，是你授权的用户的user\_id。（不要修改，后续请求时候要用）



这个时候就可以点击tql，发送请求，上传图片（这个代码需要修改以下，可以自己去发一个二手商品，然后修改商品图片，因为测试代码中是我的一个商品，可能无法操作）。

最后如果遇到任何服务端的错误，可以直接到支持中心搜索：<http://open.taobao.com/support/index.htm> 很多问题就可以被得到解答。当如果无法得到解答的，请加入旺旺群来交流（590851148）。