

# WEIBO IOS SDK 文档

北京新潮讯捷信息技术有限公司

编号: WEIBO\_IOS\_SDK

版本: WEIBO\_IOS\_SDK V2.4.0

修订记录:

时间	文档版本	修订人	备注
2012/07/19	1.0.0	陈行政	初稿
2013/01/30	1.0.1	陈行政	文档整合
2013/01/31	1.1.0	陈行政	文档更新
2013/03/12	2.0.0	洪涛	SDK 升级到 2.0 版本
2013/04/16	2.1.0	唐庆杰	SDK 升级到 2.1 版本
2013/09/09	2.3.0	唐庆杰	新增登陆登 出按钮、好 友邀请功能
2013/10/20	2.3.2	洪涛	重写文档， 添加完整 SDK 指引

WEIBO IOS SDK 文档.....	1
Weibo SDK .....	3
一、 概述 .....	3
二、 SDK 接入设置（请仔细阅读本节） .....	5
三、 应用场景代码示例 .....	10

# Weibo SDK

## 一、概述

Weibo IOS 版本 SDK，是方便移动应用开发者更快捷地接入新浪微博开放平台，使用微博 API 及平台服务的一款产品。

最新版本 SDK 具备如下功能：

1、SSO 授权(单点登录功能)：接入新浪微博的第三方 IOS 应用可通过[新浪微博官方客户端](#)快速完成 OAuth2.0 授权。不需要重复输入新浪微博用户名、密码，只需要一步操作，直接点击授权按钮即可完成授权，增强了操作简便性及帐号安全性。（如用户未安装微博客户端，将唤起 SDK 自定义的 WEBVIEW 实现授权）

SSO 授权流程示例图：



2、微博原生分享（在有官方客户端的情况下）：

原生分享（Native Share），即通过调用官方微博客户端的微博发布器，来实现发送微博的功能。

当前支持两种形式：唤起分享和集成分享。

a) 唤起分享：从第三方应用选择内容分享到微博，



发布后的文案来源为第三方应用。

b) 集成分享（需要合作接入，LinkCard 和附件栏集成分享权限需要合作申请，商务对接请发邮件至：[yuqing1@staff.sina.com.cn](mailto:yuqing1@staff.sina.com.cn)）：集成分享是从官方微博客户端中唤起第三方应用，实现分享过程。

要实现集成分享首先要用户添加第三方应用到附件栏



然后由用户主动点击微博客户端附件栏图标唤起第三方应用进行分享：



发布后的文案来源为官方微博客户端。

### 3、邀请好友：

最新版本 SDK 中集成了对邀请接口的封装，通过相应调用可以实现对好友使用当前应用的邀请，好友将收到私信例如下图：



## 二、SDK 接入设置（请仔细阅读本节）

### 1) 注册成为开发者，创建移动应用



如果你还不是一名开发者, 请先注册成为开发者, 具体参考新手指南 :  
<http://open.weibo.com/wiki/%E6%96%B0%E6%89%8B%E6%8C%87%E5%8D%97>  
创建应用时, 开发者需要谨慎选择应用对应平台, 不同的平台建议使用不同 APPKEY 开发。

应用平台:	<input type="checkbox"/> iPhone	<input type="checkbox"/> Android	<input type="checkbox"/> BlackBerry
	<input type="checkbox"/> Windows Phone	<input type="checkbox"/> Symbian	<input type="checkbox"/> WebOS
	<input type="checkbox"/> Other		

本文档读者请选择 iPhone

## 2) 设定授权回调页

请在“我的应用 - 应用信息 - 高级信息”中填写您的应用回调页, 这样才能使 OAuth2.0 授权正常进行。如果您的 APPSECRET 发生泄露, 您也可以通过该页面中的重置按钮对其重置, 如下图所示:

客户端sdk测...

控制台

应用信息

基本信息

高级信息

测试帐号

数据统计

接口管理

转让应用

删除应用

OAuth2.0 授权设置

授权回调页：<https://api.weibo.com/oauth2/default.html>

取消授权回调页：<https://api.weibo.com/oauth2/default.html>

安全设置

应用的服务IP地址：未填写

重置App Secret

cf95889f127786279e4a7ad286258949

重置

如果App Secret泄露，可以通过重置更换，原App Secret将作废

注意：IOS 应用推荐使用默认授权回调页！地址为：  
<https://api.weibo.com/oauth2/default.html>

### 3) 设定 Apple ID 和 Bundle ID

请在“我的应用 - 应用信息 - 基本信息”中填写您的 Apple ID 和 Bundle ID，这样您的应用才能正常使用微博 IOS SDK 授权和回调。（更改设置有延时，建议退出账号重新登录后再测试）

应用基本信息

应用类型：普通应用 - 客户端

应用名称：客户端sdk测试使用

该名称也用于来源显示，不超过10个汉字或20个字母

应用平台：手机

查看移动客户端接入指南

☒ iPhone
 ☐ Android
 ☐ BlackBerry
 ☐ Windows Phone
 ☐ Symbian
 ☐ WebOS
 ☐ Other

☒

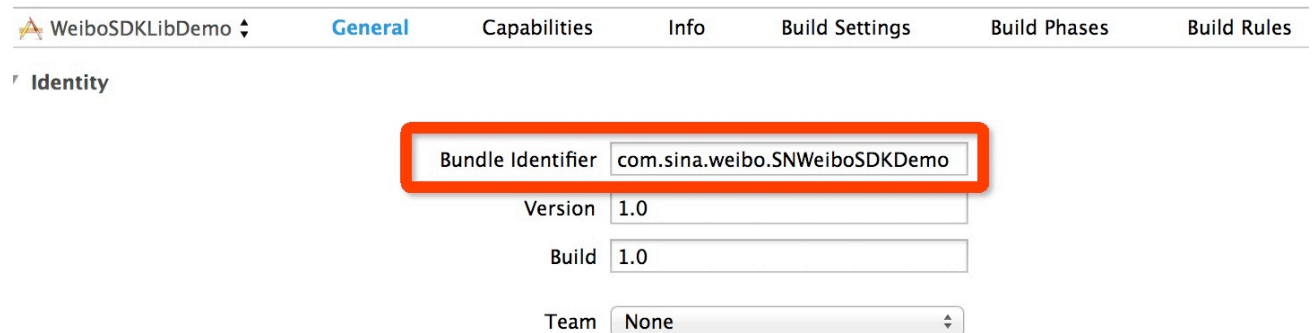
\* Apple ID：

如何查找Apple ID ?

\* Bundle ID：

注：Apple ID 如果没有的话，先随意填写，当获取了合法的 Apple ID 之后请马

上到这个页面修改为正式版本。而 Bundle ID 需要和工程设置保证一致，在 XCODE5 下 Bundle 的截图如下：



## 4) 设置工程回调 URL Scheme

修改 info.plist 文件 URL types 项为自己的 sso 回调地址，”WB[你的应用程序的 Appkey]”，例如:wb204543436852

如图所示：

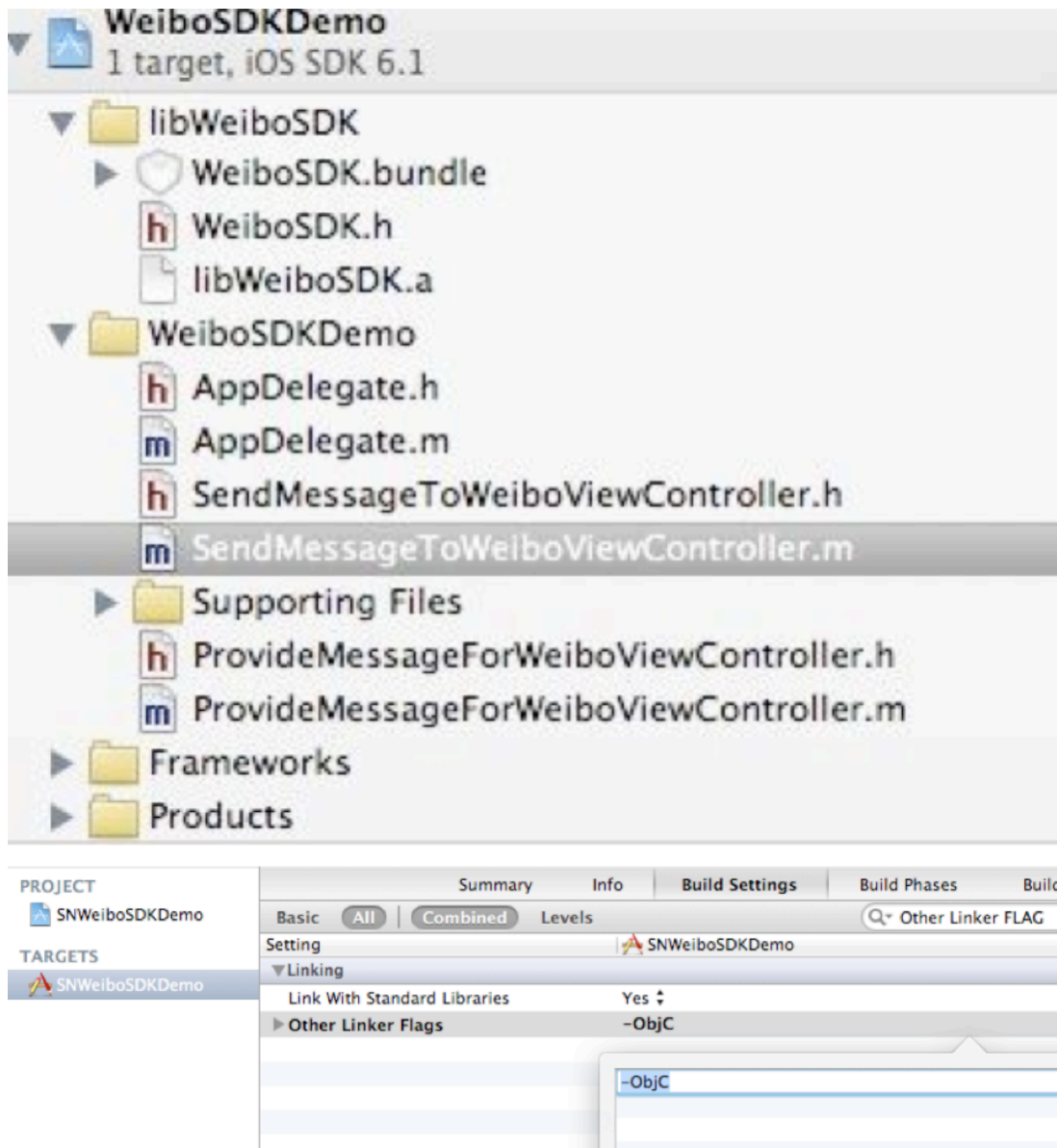


## 5) 添加 SDK 文件到工程

将从 GitHub 上下载的 libWeiboSDK 文件夹添加至工程，其中包含 WeiboSDK.h libWeiboSDK.a 和 WeiboSDK.bundle 三个文件。

在工程中引入静态库之后,需要在编译时添加 - objc 编译选项,避免静态库中类加载 不全造成程序崩溃。方法:程序 Target->Buid Settings->Linking 下 Other Linker Flags 项添加-ObjC。





## 6) 定义应用 SSO 登录或者 OAuth2.0 认证所需的几个常量

- AppKey:第三方应用申请的 appkey,用来身份鉴证、显示来源等;
- AppRedirectURL:应用回调页,在进行 OAuth2.0 登录认证时所用。对于 Mobile 客户端应用来说,是不存在 Server 的,故此处的应用回调页地址只要与新浪微博开放平台->我的应用->应用信

息->高级应用->授权设置->应用回调页中的 url 地址保持一致就可以了,如图所示:

```
#define kAppKey @"2045436852"
#define kRedirectURI @"http://www.sina.com"
```

## 7) 注册 appkey(clientid)

程序启动时,在代码中向微博终端注册你的 Appkey,如果首次集成微博SDK,建议打开调试选项以便输出调试信息。

```
- (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [WeiboSDK enableDebugMode:YES];
    [WeiboSDK registerApp:kAppKey];
}
```

## 8) 重写 AppDelegate 的 handleOpenURL 和 openURL 方法

```
- (BOOL)application:(UIApplication *)application
    openURL:(NSURL *)url
    sourceApplication:(NSString *)sourceApplication
    annotation:(id)annotation
{
    return [WeiboSDK handleOpenURL:url delegate:self];
}
```

```
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url
{
    return [WeiboSDK handleOpenURL:url delegate:self];
}
```

## 三、应用场景代码示例

### 1、SSO 微博客户端授权认证

```

{
    WBAuthorizeRequest *request = [WBAuthorizeRequest request];
    request.redirectURI = kRedirectURI;
    request.scope = @"all";
    request.userInfo = @{@"SSO_From": @"SendMessageToWeiboViewController",
                        @"Other_Info_1": [NSNumber numberWithInt:123],
                        @"Other_Info_2": @[@"obj1", @"obj2"],
                        @"Other_Info_3": @{@"key1": @"obj1", @"key2": @"obj2"}};
    [WeiboSDK sendRequest:request];
}

```

调用 `SendRequest` 的方法后会跳转到微博程序。如果当前微博客户端没有账号,则进入登录界面;如果当前微博客户端已经有账户,则进入账户管理界面,选择要向第三方授权的账户。当授权完成后会回调给第三方应用程序,第三方实现 `WeiboSDKDelegate` 的 `didReceiveWeiboResponse` 方式监听此次请求的 `response`。

此中 `UserInfo` 内容为用户自定义(可不填写),微博回调 `Response` 中会通过 `requestUserInfo` 包含原 `request.userInfo` 中的所有数据,用于第三方自定义操作或 `request` 区分。

## 2、从第三方应用向微博发送请求

代码示例如:

```

WBSendMessageToWeiboRequest *request = [WBSendMessageToWeiboRequest requestWithMessage:[self
messageToShare]];
request.userInfo = @{@"ShareMessageFrom": @"SendMessageToWeiboViewController",
                    @"Other_Info_1": [NSNumber numberWithInt:123],
                    @"Other_Info_2": @[@"obj1", @"obj2"],
                    @"Other_Info_3": @{@"key1": @"obj1", @"key2": @"obj2"}};
[WeiboSDK sendRequest:request];

```

同上此中 `UserInfo` 内容为用户自定义(可不填写),微博回调 `Response` 中会通过 `requestUserInfo` 包含原 `request.userInfo` 中的所有数据,用于第三方自定义操作或 `request` 区分。

## 3、从微博的发布界面调起第三方,向第三方请求数据

第三方案需要实现 `WeiboSDKDelegate` 中的 `didReceiveWeiboRequest` 的方法,当用户选择了第三方提供的数据以后,将数据封装成 `WBMessageObject`

```

- (WBMessageObject *)messageToShare
{
    WBMessageObject *message = [WBMessageObject message];

    message.text = @"测试使用";

    return message;
}

```

构造 `WBProvideMessageForWeiboResponse` 对象如:

```

WBProvideMessageForWeiboResponse *response = [WBProvideMessageForWeiboResponse responseWithMessage:
[self messageToShare]];
[WeiboSDK sendResponse:response];

```

通过 `sendResponse` 方法回传数据给微博客户端完成此次任务。

## 4、用户取消对应用的授权

调用[WeiboSDK logOutWithToken: delegate:];方法即可  
调用此接口后，微博 SDK 会发起网络请求使 token 失效

@param token 第三方应用之前申请的 Token

@param delegate WBHttpRequestDelegate 对象，用于接收微博 SDK 对于发起的接口请求的响应

应用可实现 WBHttpRequestDelegate 中的

-(void)request:(WBHttpRequest\*)request didReceiveResponse:(NSURLResponse \*)response;

-(void)request:(WBHttpRequest \*)request didFailWithError:(NSError \*)error;

-(void)request:(WBHttpRequest \*)request didFinishLoadingWithResult:(NSString \*)result;

方法用于监听此次 Http 请求，如：

```
-(void)request:(WBHttpRequest *)request didFinishLoadingWithResult:(NSString *)result
{
    NSString *title = nil;
    UIAlertView *alert = nil;

    title = @"收到网络回调";
    alert = [[UIAlertView alloc] initWithTitle:title
                                     message:[NSString stringWithFormat:@"%@",result]
                                     delegate:nil
                                     cancelButtonTitle:@"确定"
                                     otherButtonTitles:nil];

    [alert show];
    [alert release];
}

-(void)request:(WBHttpRequest *)request didFailWithError:(NSError *)error;
{
    NSString *title = nil;
    UIAlertView *alert = nil;

    title = @"请求异常";
    alert = [[UIAlertView alloc] initWithTitle:title
                                     message:[NSString stringWithFormat:@"%@",error]
                                     delegate:nil
                                     cancelButtonTitle:@"确定"
                                     otherButtonTitles:nil];

    [alert show];
    [alert release];
}
```

注：向好友发送应用邀请、以及请求 OpenApi 为同样使用方法，具体源码见 Demo 源代码。

## 5、向相互关注的好友发送邀请

调用

+(void)inviteFriend:(NSString\* )data withUid:(NSString \*)uid withToken:(NSString \*)access\_token delegate:(id<WBHttpRequestDelegate>)delegate;

方法即可

@param data 邀请数据。必须为 json 字串的形式，必须做 URLEncode，采用 UTF-8 编码。

data 参数支持的参数：

{

text 要回复的私信文本内容。文本大小必须小于 300 个汉字。

url 邀请点击后跳转链接。默认为当前应用地址。

invite\_logo 邀请 Card 展示时的图标地址，大小必须为 80px X 80px，仅支持 PNG、

JPG 格式。默认为当前应用 logo 地址。

}

@param uid 被邀请人，需为当前用户互粉好友。

@param access\_token 第三方应用之前申请的 Token

@param delegate WBHttpRequestDelegate 对象，用于接收微博 SDK 对于发起的接口请求的响应

#### 6、请求 openAPI

调用如下两方法均可：

```
+ (void)requestWithURL:(NSString *)url
        httpMethod:(NSString *)httpMethod
        params:(NSDictionary *)params
        delegate:(id<WBHttpRequestDelegate>)delegate;
```

```
+ (void)requestWithAccessToken:(NSString *)accessToken
        url:(NSString *)url
        httpMethod:(NSString *)httpMethod
        params:(NSDictionary *)params
        delegate:(id<WBHttpRequestDelegate>)delegate;
```

参数说明如下：

@param accessToken 应用获取到的 accessToken，用于身份验证

@param url 请求 url 地址

@param httpMethod 支持"GET" "POST"

@param params 向接口传递的参数结构

@param delegate WBHttpRequestDelegate 对象，用于接收微博 SDK 对于发起的接口请求的响应

应用同样可实现 WBHttpRequestDelegate 中的

```
-(void)request:(WBHttpRequest*)request didReceiveResponse:(NSURLResponse*)response;
```

```
- (void)request:(WBHttpRequest *)request didFailWithError:(NSError *)error;
```

```
- (void)request:(WBHttpRequest *)request didFinishLoadingWithResult:(NSString*)result;
```

方法用于监听此次 Http 请求，

**注：以上场景均可在 Demo 源码中找到相应代码片段**