

# WEIBO IOS OAUTH SDK 文档

北京新潮讯捷信息技术有限公司

编号：WEIBO\_IOS\_OAUTH\_SDK

版本：WEIBO\_IOS\_OAUTH\_SDK V1.0.0

修订记录

时间	文档版本	修订人	备注
2012/8/6	1.0.0	陈行政	初稿

## 目录

WEIBO IOS OAUTH SDK 文档 .....	1
一、 概述 .....	3
名词解释 .....	3
二、 流程 .....	4
三、 接口说明 .....	5
SinaWeibo .....	5
SinaWeiboDelegate .....	6
SinaWeiboRequestDelegate .....	7
SinaWeiboAuthorizeView .....	8
SinaWeiboRequest .....	8
SinaWeiboConstants.h .....	8
四、 集成步骤及示例分析 .....	8
1. 新建工程，引入相关文件 .....	8
2. 定义所需常量 .....	9
3. 实现 SinaWeiboDelegate 协议 .....	9
4. 实现 SinaWeiboRequestDelegate 协议 .....	10
5. AppDelegate 中初始化 SinaWeibo 对象 .....	11
6. 实现登录操作 .....	12
7. 实现请求微博 API 操作 .....	12
8. 实现退出操作 .....	12
五、 操作过程解析 .....	13
1. Oauth2.0 认证登录过程 .....	13
2. 一般请求过程 .....	14

## 一、 概述

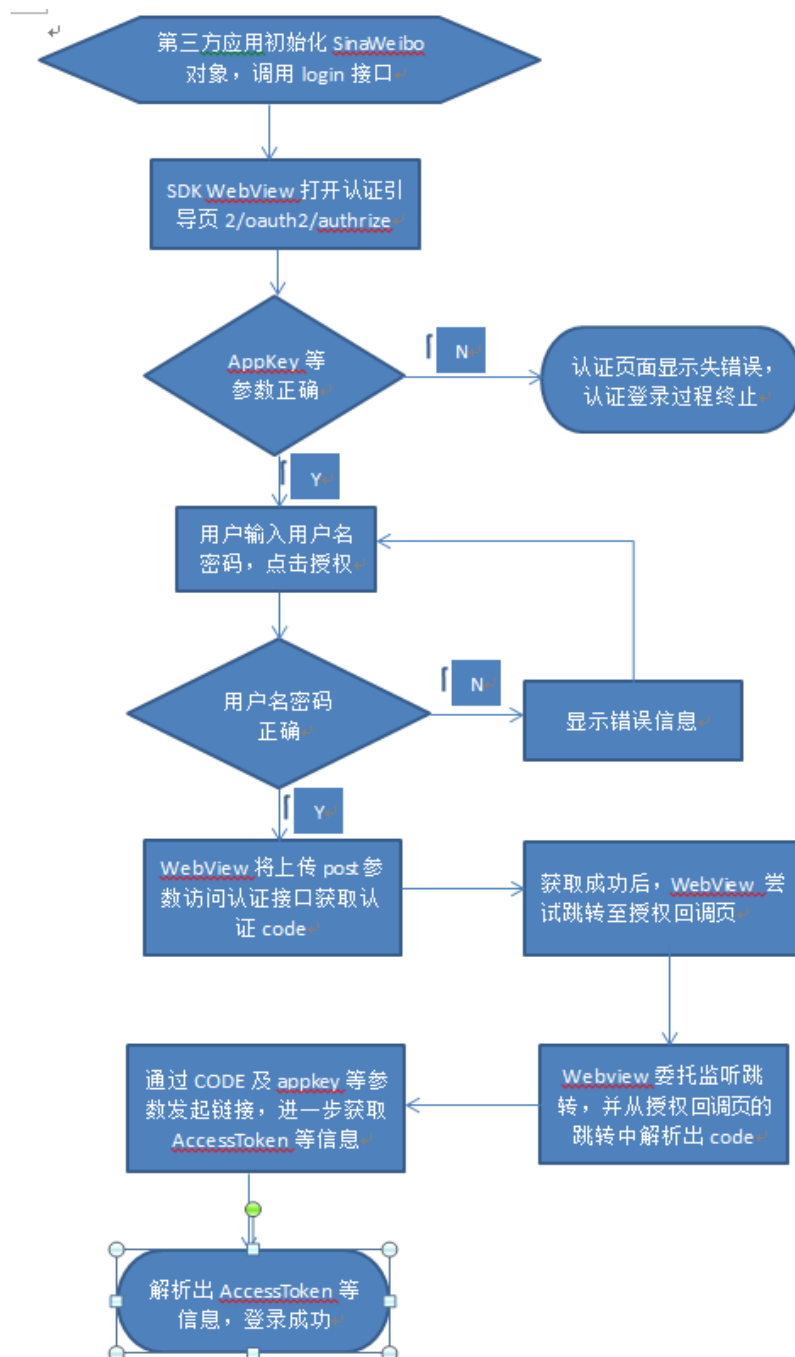
Oauth SDK 为第三方应用封装 Oauth2.0 登录认证接口，隐藏了 Oauth2.0 认证的复杂步骤。第三方应用只需配置好自己应用的相关属性（appKey/appSecret/RedirectURI），即可使用 SDK 认证登录并调用 OPEN API。本文档旨在说明 SDK 的工作流程包括相关变量，相关类的作用。SDK DEMO 的下载及使用可参照[https://github.com/mobileresearch/weibo\\_sdk\\_oauth.git](https://github.com/mobileresearch/weibo_sdk_oauth.git)。

### 名词解释

AppKey	分配给每个第三方应用的 app key。用于鉴权身份，显示来源等功能。
APPSecret	生成请求 request Token 的 secret，与 app key 一起分配。
AccessToken	表示用户身份的 token，用于微博 API 的调用。
ExpirationDate	过期时间，用于判断登录是否过期。
RedirectURI	应用回调页面，可在新浪微博开放平台->我的应用->应用信息->高级应用->Oauth2.0 授权设置->授权回调页中找到。注：使用 SDK 认证需定义此值，若第三方应用未配置此值，请编辑此值。

## 二、 流程

Oauth2.0 认证过程将通过 WebView 调用 <https://api.weibo.com/2/oauth2/authorize> 接口获取 code，进而用获取的 code 调用 [https://api.weibo.com/2/oauth2/access\\_token](https://api.weibo.com/2/oauth2/access_token) 获取 token。随 token 返回的还有 expiration、uid 等信息。



### 三、 接口说明

#### SinaWeibo

SinaWeibo 作为提供给第三方应用的主要接口，提供登录，登出，以及访问微博 api 的功能。用户在使用 sdk 前需实现唯一的 SinaWeibo 对象，将在之后的一系列操作中用到。

方法	- (id)initWithAppKey:(NSString *)_appKey appSecret:(NSString *)_appSecret appRedirectURI:(NSString *)_appRedirectURI andDelegate:(id<SinaWeiboDelegate>)_delegate	
简介	初始化方法，构造一个 SinaWeibo 对象。	
参数&作用	_appKey	分配给第三方应用的 app key
	_appSecret	分配给第三方应用的 app secret
	_appRedirectURI	应用回调页
	_delegate	第三方应用实现的 SinaWeiboDelegate 协议对象，用来处理登录及登出结果
返回类型	SinaWeibo:构造函数返回当前实现的对象，在第三方应用中调用此对象进行登入登出等操作	

方法	- (void)login	
简介	登录操作，直接调用。SinaWeibo 创建一个 WebView 进行浏览器登录，Oauth 认证完成后，获取 AccessToken 等信息，并通过构造时传入的 SinaWeiboDelegate 对象调用 sinaweiboDidLogin 方法将登录结果返回给应用。	
参数&作用	无	无
返回类型	无	

方法	- (void)logout	
简介	登出操作，直接调用。在接收到登出结果后解析，并通过构造时传入的 SinaWeiboDelegate 对象调用 sinaweiboDidLogout 方法将登出结果返回给应用。	
参数&作用	无	无
返回类型	无	

方法	- (BOOL)isLoggedIn	
简介	判断是否已登录	
参数&作用	无	无
返回类型	YES 为已登录；NO 为未登录	

方法	- (BOOL) isAuthorizeExpired	
简介	判断是否过期	
参数&作用	无	无
返回类型	YES 为已过期；NO 为未过期	

方法	- (BOOL) isAuthValid	
简介	判断登录状态是否有效，即已登录且未过期	
参数&作用	无	无
返回类型	YES 为登录状态有效；NO 为登录状态无效	

方法	- (SinaWeiboRequest *)requestWithURL:(NSString *)url params:(NSMutableDictionary *)params httpMethod:(NSString *)httpMethod delegate:(id<SinaWeiboRequestDelegate>)_delegate	
简介	微博 API 的请求接口，发送请求时附带 token 值。	
参数&作用	url	请求的 api 地址，如 “users/show.json” (请求用户资料)
	params	请求的参数，如 “[NSMutableDictionary dictionaryWithObject:sinaweibo.userID forKey:@"uid"]” (用户的 uid)
	httpMethod	请求类型，如 “@GET” (GET 类型)
	_delegate	SinaWeiboRequestDelegate 对象，用于处理请求结果的回调处理。应用在请求之前需实现此协议并新建一个对应的对象，请求时作为 _delegate 参数传入。
返回类型	SinaWeiboRequest 对象	

## SinaWeiboDelegate

应用要完成登录的功能，需实现此协议，用来处理登入登出的回调结果。用户登录之前，初始化

方法	- (void)sinaweiboDidLogIn:(SinaWeibo *)sinaweibo	
简介	SinaWeibo 对象登录操作完成后，调用此方法。应用根据登录结果可进行自己的相关处理。	
参数&作用	sinaweibo	SinaWeibo 对象，应用可在方法中通过此对象判断登录结果，以进行不同的处理。
返回类型	无	

方法	- (void)sinaweiboDidLogOut:(SinaWeibo *)sinaweibo	
简介	SinaWeibo 对象登出操作完成后，调用此方法。应用根据登出结果可进行自己的相关处理。	
参数&作用	sinaweibo	SinaWeibo 对象，应用可在方法中通过此对象判断登出结果，以进行不同的处理。
返回类型	无	

方法	- (void)sinaweiboLogInDidCancel:(SinaWeibo *)sinaweibo	
简介	监听取消登录操作。	
参数&作用	sinaweibo	SinaWeibo 对象。
返回类型	无	

方法	- (void)sinaweibo:(SinaWeibo *)sinaweibo loginDidFailWithError:(NSError *)error;	
简介	登录失败后，将错误信息回调给此方法。	
参数&作用	sinaweibo	SinaWeibo 对象。
	error	失败信息。
返回类型	无	

方法	- (void)sinaweibo:(SinaWeibo *)sinaweibo accessTokenInvalidOrExpired:(NSError *)error;	
简介	操作过程中出错时 sinaweibo 对象将调用此方法，如 accessToken 无效或者过期。用户接收到此错误信息后应重新登录。	
参数&作用	sinaweibo	SinaWeibo 对象。
	error	失败信息。
返回类型	无	

## SinaWeiboRequestDelegate

应用进行请求前，需实现此协议，并将对象传入，用以监听请求结果。

方法	- (void)request:(SinaWeiboRequest *)request didFailWithError:(NSError *)error	
简介	请求失败时回调此方法。	
参数&作用	request	SinaWeiboRequest 对象，SDK 中具体用来实现请求的类，应用可根据 request.url 来判断具体哪个请求失败。
	error	失败信息。
返回类型	无	

方法	- (void)request:(SinaWeiboRequest *)request didFinishLoadingWithResult:(id)result	
简介	请求完成后回调此方法。	
参数&作用	request	SinaWeiboRequest 对象，SDK 中具体用来实现请求的类，应用可根据 request.url 来判断具体哪个请求。
	result	返回的结果，一般为请求的内容。
返回类型	无	

方法	- (void)request:(SinaWeiboRequest *)request didReceiveResponse:(NSURLResponse *)response;	
简介	请求接收到响应（应用不需要实现此方法）。	
参数&作用	request	SinaWeiboRequest 对象。
	response	响应信息。
返回类型	无	

方法	- (void)request:(SinaWeiboRequest *)request didReceiveRawData:(NSData *)data;	
简介	请求接收到数据（应用不需要实现此方法）。	
参数&作用	request	SinaWeiboRequest 对象。
	data	响应数据，最后需要对数据解析。
返回类型	无	

## SinaWeiboAuthorizeView

微博认证组件，进行 OAuth2.0 认证，此页面将打开内置浏览器，并请求用户授权，授权成功后将授权结果通过 SinaWeiboDelegated 对象(构造 SinaWeibo 对象时传入的 delegate) 返回给应用。第三方应用不需要直接使用此类，其功能已集成到 SinaWeibo 的登录方法中了。

## SinaWeiboRequest

请求微博 API 的类，其调用封装在 SinaWeibo 类中。

## SinaWeiboConstants.h

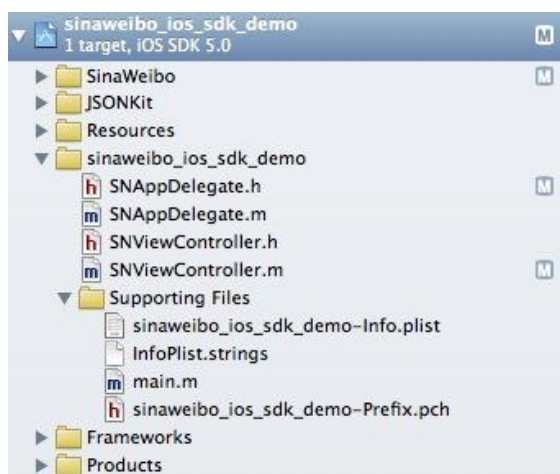
定义常量的头文件。

# 四、 集成步骤及示例分析

以实际例子分析第三方应用引入 WeiboSDK 的过程。具体代码参考 sinaweibo\_ios\_sdk\_demo 工程。

## 1. 新建工程，引入相关文件

创建新的工程，将 SinaWeibo 和 JSONKit 两个包拷贝并添加至工程中。





## 2. 定义所需常量

定义应用进行 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;
- **AppSecret:** 与 appkey 一起分配给第三方应用的 appsecret, 用来生成 request token;
- **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:



## 3. 实现 SinaWeiboDelegate 协议

第三方应用需要实现自己的 SinaWeiboDelegate, 来处理用户登录、退出以及取消登录的回调。方法详解见接口说明一节。需要实现如下方法:

```
#pragma mark - SinaWeibo Delegate

// 用户成功登陆
// 注意需要将登陆成功的用户信息缓存下来以便下次使用
- (void)sinaWeiboDidLogIn:(SinaWeibo *)sinaweibo
{
    NSLog(@"sinaweiboDidLogIn userID = %@ accesstoken = %@ expirationDate = %@", sinaweibo.userID, sinaweibo.accessToken, sinaweibo.expirationDate);
    [self storeAuthData];
}

// 用户退出登陆
// 注意需要清除已经缓存的用户信息
- (void)sinaWeiboDidLogOut:(SinaWeibo *)sinaweibo
{
}
```

```

        NSLog(@"sinaweiboDidLogout");

        [self removeAuthData];
    }

    // 用户取消登陆过程
    - (void)sinaweiboLoginDidCancel:(SinaWeibo *)sinaweibo
    {
        NSLog(@"sinaweiboLoginDidCancel");
    }

    // 登陆失败
    - (void)sinaweibo:(SinaWeibo *)sinaweibo loginDidFailWithError:(NSError *)error
    {
        NSLog(@"sinaweibo loginDidFailWithError %@", error);
    }

    // 缓存的 token 无效或者已经过期
    // 注意需要清除缓存的用户信息
    - (void)sinaweibo:(SinaWeibo *)sinaweibo accessTokenInvalidOrExpired:(NSError *)error
    {
        NSLog(@"sinaweiboAccessTokenInvalidOrExpired %@", error);

        [self removeAuthData];
    }
}

```

## 4. 实现 SinaWeiboRequestDelegate 协议

第三方应用实现自己的 SinaWeiboRequestDelegate，来处理微博 api 请求的结果。方法详见接口说明一节。需要实现以下方法：

```

#pragma mark - SinaWeiboRequest Delegate

// 请求失败
- (void)request:(SinaWeiboRequest *)request didFailWithError:(NSError *)error
{
    // 当一个 delegate 需要处理多个请求回调时，可以通过 url 来判断当前的 request
    if ([request.url hasSuffix:@"users/show.json"])
    {
    }

    else if ([request.url hasSuffix:@"statuses/upload.json"])
    {
    }
}

```

```

        [self resetButtons];
    }

    // 请求成功
    - (void)request:(SinaWeiboRequest *)request didFinishLoadingWithResult:(id)result
    {
        // 当一个 delegate 需要处理多个请求回调时，可以通过 url 来判断当前的 request
        if ([request.url hasSuffix:@"users/show.json"])
        {
        }

        else if ([request.url hasSuffix:@"statuses/upload.json"])
        {
        }

        [self resetButtons];
    }

```

## 5. AppDelegate 中初始化 SinaWeibo 对象

- 在 AppDelegate 中定义 sinaweibo 属性并使用自己 APP 的相关属性初始化 sinaweibo 对象
  - ◆ AppKey / AppSecret / AppRedirectURI 是在新浪微博开放平台上定义的 APP 相关属性
  - ◆ delegate 是你自己的程序中用于处理微博帐户登陆、退出相关的对象（SinaWeiboDelegate）
- 读取已缓存的相关用户信息，并传递给 sinaweibo 对象

```

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[[UIWindow alloc] initWithFrame:[UIScreen mainScreen] bounds]] autorelease;

    // Override point for customization after application launch.

    self.viewController = [[[SNViewController alloc] initWithNibName:nil bundle:nil] autorelease];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];

    sinaweibo = [[SinaWeibo alloc] initWithAppKey:kAppKey appSecret:kAppSecret appRedirectURI:kAppRedirectURI
    andDelegate:your_delegate];

    NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
    NSDictionary *sinaweiboInfo = [defaults objectForKey:@"SinaWeiboAuthData"];
    if ([sinaweiboInfo objectForKey:@"AccessTokenKey"] && [sinaweiboInfo objectForKey:@"ExpirationDateKey"] &&
    [sinaweiboInfo objectForKey:@"UserIDKey"])
    {
    }
}

```

```
sinaweibo.accessToken = [sinaweiboInfo objectForKey:@"AccessTokenKey"];
sinaweibo.expirationDate = [sinaweiboInfo objectForKey:@"ExpirationDateKey"];
sinaweibo.userID = [sinaweiboInfo objectForKey:@"UserIDKey"];
}
return YES;
}
```

## 6. 实现登录操作

应用创建 SinaWeibo 对象之后，只需调用[SinaWeibo login]方法，即可实现登录，之后的操作交给 SDK 完成，最后在自己实现的 SinaWeiboDelegate 类中接收登录结果即可。

## 7. 实现请求微博 API 操作

应用登陆成功后，通过创建的 SinaWeibo 对象调用 request 方法实现微博 API 的请求。

```
- (void)userInfoButtonPressed
{
    SinaWeibo *sinaweibo = [self sinaweibo];
    [sinaweibo requestWithURL:@"users/show.json"
                      params:[NSMutableDictionary dictionaryWithObject:sinaweibo.userID forKey:@"uid"]
                httpMethod:@"GET"
                delegate:self];
}
```

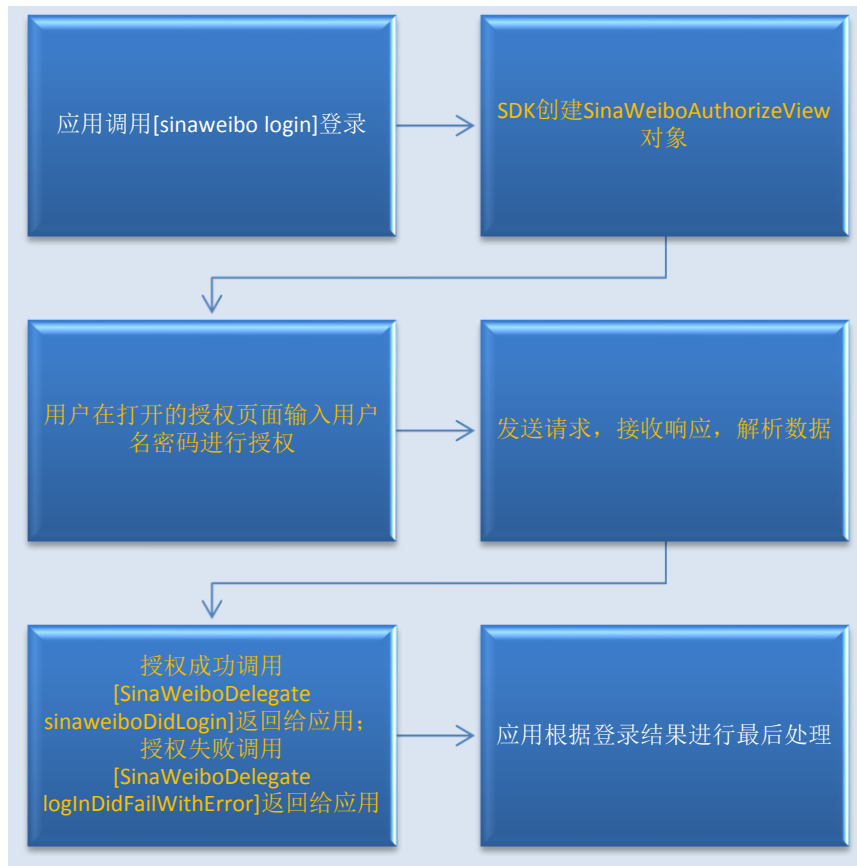
## 8. 实现退出操作

通过调用[SinaWeibo logOut]方法退出，并在 SinaWeiboDelegate 处理退出回调结果即可。

## 五、 操作过程解析

### 1. OAuth2.0 认证登录过程

若没有 SSO 登录的环境(未安装微博官方客户端或版本不支持)，则进行 OAuth2.0 认证登录。



## 2. 一般请求过程

