微博iOS平台SDK文档

编号：WEIBO_IOS_SDK
版本：WEIBO_IOS_SDK V3.1.4

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
2014/11/17	3.0.0	邱文杰	SDK升级到3.0版本
2015/02/04	3.1.0	李靖宇	新增短信注册通道
2016/05/11	3.1.4	李靖宇	新增私信分享，支持ipv6-only

目录

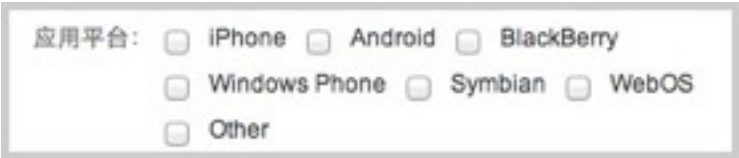
Weibo SDK

一.SDK接入设置

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。



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：

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）

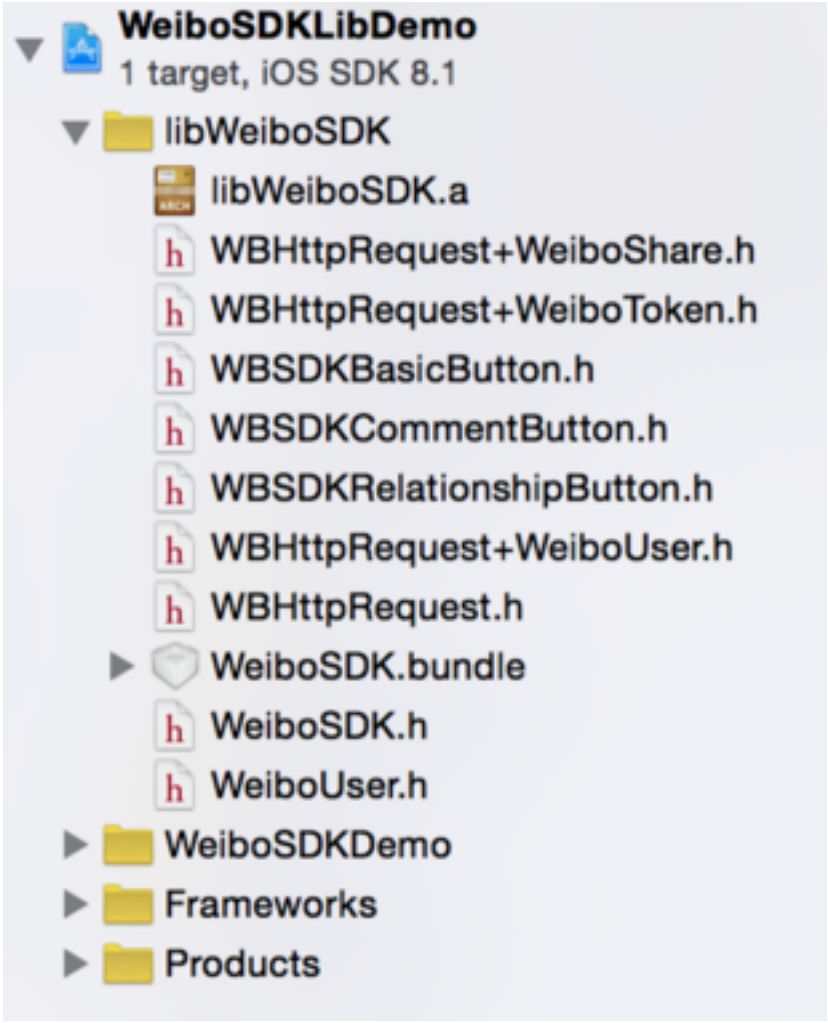
注：Apple ID如果没有的话，先随意填写，当获取了合法的Apple ID之后请马上到这个页面修改为正式版本。而Bundle ID需要和工程设置保证一致，在XCODE5下Bundle的截图如下：

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

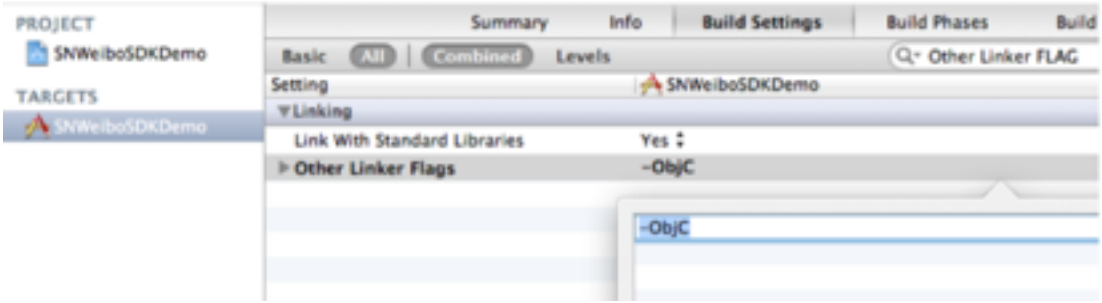
5.添加SDK文件到工程

将从GitHub上下载的libWeiboSDK文件夹添加至工程，其中包含WeiboSDK.h、WeiboUser.h 、WBHttpRequest.h、WBHttpRequest+WeiboUser.h、WBHttpRequest+WeiboShare.h、WBHttpRequest+WeiboToken.h、WBSDKBasicButton.h、WBSDKRelationshipButton.h、WBSDKCommentButton.h、WeiboSDK+Statistics.h这10个.h文件以及libWeiboSDK.a 和 WeiboSDK.bundle，统共12个文件。



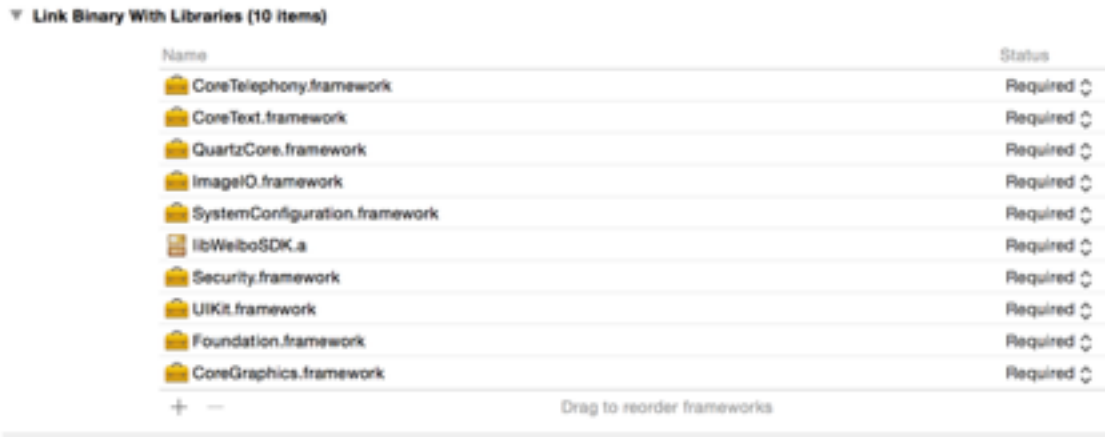
6.在工程中引入静态库之后,需要在编译时添加 –objC 编译选项

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。



7.添加FrameWork文件到工程

在工程中修改Other Linker Flags后，需要修改编译步骤的链接库设置，避免链接阶段由于库的设置错误导致程序崩溃。方法:程序 Target->Buid Phases->Link Binary With Libraries下添加以下Framework至工程中。需要添加的Frameworks为：QuartzCore.framework、ImageIO.framework、SystemConfiguration.framework、Security.framework、CoreTelephony.framework、CoreText.framework、UIKit.framework、Foundation.framework、CoreGraphics.framework 、libz.dylib、libsqlite3.dylib。



8.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"
```

9.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];
}
```

10.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 ];
}
```

二、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

第三方程序需要实现 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.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，

此外,为了方便使用,SDK 还封装了部分请求 Open API 请求,范例方法如下: +(WBHttpRequest *)requestForFriendsListOfUser:(NSString*)currentUserID withAccessToken:(NSString*)accessToken andOtherProperties:(NSDictionary*)otherProperties queue:(NSOperationQueue*)queue withCompletionHandler:(WBRequestHandler)handler;

这个请求调用Open API的friendships/friends接口,接口返回相应的关注人列表。 参数说明如下:

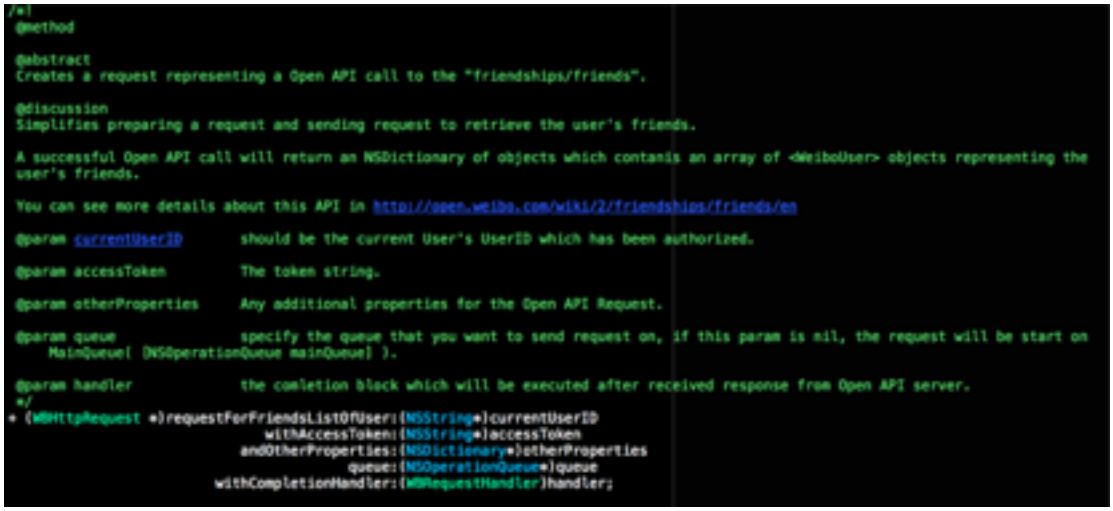
@param currentUserID 当前授权用户所对应的 UserID.

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

@param otherProperties 向接口传递的额外参数,

@param queue 发起请求的线程,默认为主线程,

@param handler 用于接收接口请求的请求的响应 Block。



```
/*!
 @method

 @abstract
 Creates a request representing a Open API call to the "friendships/friends".

 @discussion
 Simplifies preparing a request and sending request to retrieve the user's friends.

 A successful Open API call will return an NSDictionary of objects which contains an array of <WeiboUser> objects representing the
 user's friends.

 You can see more details about this API in http://open.weibo.com/wiki/2/friendships/friends/en

 @param currentUserID       should be the current User's UserID which has been authorized.

 @param accessToken         The token string.

 @param otherProperties     Any additional properties for the Open API Request.

 @param queue               specify the queue that you want to send request on, if this param is nil, the request will be start on
     MainQueue( [NSOperationQueue mainQueue] ).

 @param handler             the completion block which will be executed after received response from Open API server.
 */
+ (WBHttpRequest *)requestForFriendsListOfUser:(NSString*)currentUserID
                                withAccessToken:(NSString*)accessToken
                             andOtherProperties:(NSDictionary*)otherProperties
                                          queue:(NSOperationQueue*)queue
                          withCompletionHandler:(WBRequestHandler)handler;
```

比如:

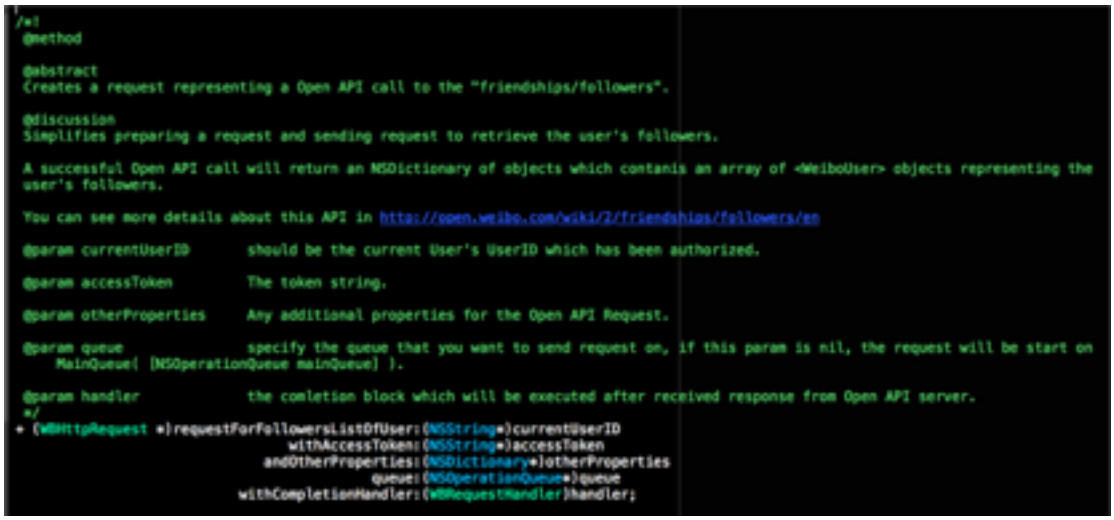
```
+(WBHttpRequest*)requestForFollowersListOfUser:(NSString*)currentUserID
                                withAccessToken:(NSString*)accessToken
  andOtherProperties:(NSDictionary*)otherProperties
        queue:(NSOperationQueue*)queue
```

withCompletionHandler:(WBRequestHandler)handler;
这个请求调用Open API的friendships/followers接口,接口返回相应的粉丝列表。
参数说明如下:

@param currentUserID 当前授权用户所对应的 UserID.
@param accessToken 应用获取到的 accessToken,用于身份验证
@param otherProperties 向接口传递的额外参数,

@param queue 发起请求的线程,默认为主线程,

@param handler 用于接收接口请求的请求的响应 Block。
你可以在WBHttpRequest+WeiboUser.h、WBHttpRequest+WeiboShare.h、WBHttpRequest+WeiboToken.h里找到类似上述两个例子的接口请求详细说明和使用方法。



6.aid 相关 aid 是设备唯一的移动设备指纹
如果您的应用需要使用一个设备唯一标识符来区分设备，可以使用aid值。
使用方法见以下方法:
+ (NSString*)getWeiboAid;
方法声明在 WeiboSDK.h 中,说明如下:

1)该方法返回当前设备的aid值。返回的aid值可能为nil,当值为nil 时会尝试向服务器获取aid值。
2）当获取成功时（aid值变为有效值）时，SDK会发出名为WeiboSDKGetAidSucessNotification的通知,通知中带有aid值。
3)当获取失败时,SDK会发出名为WeiboSDKGetAidFailNotification的通知，通知中带有NSError 对象。

7.社会化评论组件
WBSDKCommentButton.h中有如下方法：
- (id)initWithFrame:(CGRect)frame
accessToken:(NSString*)accessToken
keyword:(NSString*)keyWord

urlString:(NSString*)urlString
category:(NSString*)category
completionHandler:(WBSDKButtonHandler)handler;

这个方法的作用是初始化一个社会化评论按钮。

参数说明如下

@param frame 按钮的frame值

@param accessToken 用户授权后获取的Token

@param keyWord 社会化评论的热点词

@param urlString 社会化评论链接，可传空

@param category 领域ID, 此参数为必选参数。

@param handler 回调函数，当用户点击按钮，进行完与社会化评论组件相关的交互之后，回调的函数。

```
@interface WBSDKCommentButton : WBSDKBasicButton

/**
 初始化一个社会化评论按钮
 @param frame 按钮的frame值
 @param accessToken 用户授权后获取的Token
 @param keyWord 社会化评论的热点词
 @param urlString 社会化评论链接，可传空
 @param category  领域ID，此参数为必选参数。
 @param handler   回调函数，当用户点击按钮，进行完与社会化评论组件相关的交互之后，回调的函数。
 */
- (id)initWithFrame:(CGRect)frame
        accessToken:(NSString*)accessToken
            keyword:(NSString*)keyWord
          urlString:(NSString*)urlString
           category:(NSString*)category
  completionHandler:(WBSDKButtonHandler)handler;

@property (nonatomic, retain)NSString* keyWord;
@property (nonatomic, retain)NSString* accessToken;
@property (nonatomic, retain)NSString* urlString;
@property (nonatomic, retain)NSString* category;

@end
```

8.关注组件

WBSDKRelationshipButton.h中有如下方法：

- (id)initWithFrame:(CGRect)frame
accessToken:(NSString*)accessToken
currentUser:(NSString*)currentUserID
followUser:(NSString*)followerUserID
completionHandler:(WBSDKButtonHandler)handler;

这个方法的作用是初始化一个关注组件按钮

参数说明如下

@param frame 按钮的frame值

@param accessToken 用户授权后获取的Token

@param currentUserID 当前用户的uid值

@param followerUserID 希望当前用户加关注的用户uid值

@param handler 回调函数，当用户点击按钮，进行完关注组件相关的交互之后，回调的函数。

```objc
@interface WBSDKRelationshipButton : WBSDKBasicButton

/**
 初始化一个关注组件按钮
 @param frame 按钮的frame值
 @param accessToken 用户授权后获取的Token
 @param currentUserID 当前用户的uid值
 @param followerUserID 希望当前用户加关注的用户uid值
 @param handler    回调函数，当用户点击按钮，进行完关注组件相关的交互之后，回调的函数。
 */
- (id)initWithFrame:(CGRect)frame
        accessToken:(NSString*)accessToken
        currentUser:(NSString*)currentUserID
         followUser:(NSString*)followerUserID
  completionHandler:(WBSDKButtonHandler)handler;

@property (nonatomic, retain)NSString* accessToken;
@property (nonatomic, retain)NSString* currentUserID;
@property (nonatomic, retain)NSString* followUserID;


@property (nonatomic, assign)WBSDKRelationshipButtonState currentRelationShip;


/**
 获取最新的关注状态
 该方法会调用OpenApi，获取当前用户与目标用户之间的关注状态，并将按钮的状态改变为正确的状态。
 */
- (void)checkCurrentRelationship;

@end
```

9.支付组件

微博支付需要单独申请，开通请联系：wb_pay_kf@vip.sina.com

9.1代码示例，设置OrderString就可以

```objc
WBOrderObject *order = [[[WBOrderObject alloc] init] autorelease];
[order setOrderString:@"out_trade_no=1395132477225&price=0.01&sign_type=md5&lockCheckBind=1&sign=70091c901e3d756a14fb2583ad59a24f"];

WBPaymentRequest *request = [WBPaymentRequest requestWithOrder:order];
[WeiboSDK sendRequest:request];
```

注：以上场景均可在 Demo 源码中找到相应代码片段

9.2支付参数说明(2.0)

参数	参数名称	类型	是否必填	参数说明
sign_type	签名类型	string	是	目前仅支持RSA
sign	签名内容	string	是	
appkey	商户的appkey	string	是	区分商户不同的业务
seller_id	商户微博id	int	是	

out_pay_id	商户订单号	string	是	商户网站唯一编号，6-64位
notify_url	异步通知地址	string	否	
return_url	支付完成回调页面地址	string	否	
subject	商品名称	string	是	
body	商品描述	string	否	
total_amount	商品总价	int	是	以分为单位
expire	超时时间	int	否	单位s，最小超时时间60s，默认值1天，即86400
extra	附加信息	string	否	同步和异步回调时会带上此参数
from_third_app	第三方app	int	是	默认为1

9.3支付接口错误返回码说明

返回错误码（Error_code）	含义
100000	操作成功
100001	操作失败
100002	非法请求
100003	无效uid
100004	验证签名失败
100005	非法ua

100006	用户未绑定支付宝
100010	没有权限进行此操作
100011	无效商户
100015	账户余额不足
100018	支付宝授权失败
100019	微博客户端版本过低
100020	访问IP非法
220005	参数错误
240001	订单已支付
240002	订单已关闭
240003	下单的微博账号和支付账号不一致
240004	wb_action参数不正确
240005	非法的微博支付id
240006	订单未支付

9.4签名机制

9.4.1 生成待签名的字符串

9.4.1.1 需要参与签名的参数

在请求参数列表中，除去sign ， sign_type俩个参数外，其他需要使用到的参数皆是要签名的参数。（个别接口中参数sign_type也需要参与签名）

在通知返回参数列表中，除去sign ， sign_type俩个参数外，凡是通知返回回来的参数皆是要签名的参数。

9.4.1.2 需生成待签名字符串

对数组里的每一个值从a到z的顺序排序，若遇到相同首字母，则看第二个字母，以此类推。排序完成之后，再把所有数组值以“&”字符连接起来，组成的字符串便是待签名的字符串。

9.4.1.3 注意

没有值的参数无需传递，也无需包含到待签名数据中。

根据HTTP协议要求，传递参数中的值中如果存在特殊字符（如：&，=），那么该值需要做URL Encoding，这样请求接收方才能接收到正确的参数值。这种情况下，待签名数据应该是encoding之后的值，而不是原生值。例如：调用支付接口需要对请求参数notify_url或者return_url进行数字签名，那么待签名数据应该是notify_url（return_url）=test.com%3fa%3d1%26b%3d2，而不是notify_url（return_url）=test.com?a=1&b=2。

9.4.2 签名

9.4.2.1 RSA签名

RSA签名由公钥和私钥两部分组成。公钥和私钥都是商户通过openssl工具生成得出的。公钥由商户上传至微博支付平台公开（请连同公钥首尾内容一并上传），用于加密和验证签名。私钥是商户自己用来进行解密和签名。

请求时签名

当拿到请求时的待签名字符串后，把待签名字符串与商户的私钥一同放入RSA的签名函数中进行签名运算，从而得到签名结果字符串。

通知返回时验证签名

当得到通知返回时的待签名字符串后，把待签名字符串、微博支付提供的公钥、微博支付通知返回参数中的参数sign的值三者一同放入RSA的签名函数中进行非对称的签名运算，来判断签名是否验证通过。

10.私信邀请

WBSDK.h中有如下类和初始化方法：

```
@interface WBSDKAppRecommendRequest : WBBaseRequest
+ (id)requestWithUIDs:(NSArray*)uids access_token:(NSString *)ace
ss_token;
```

@param uids 为推荐的好友列表，为空时跳转到用户自选的页面。

@param access_token 第三方应用之前申请的Token,当此值不为空并且无法通过客户端分享的时候,会使用此token私信。

11. 短信注册

```
/*
 第三方调用微博短信注册或者登陆
 @param navTitle 为登陆页navigationBar的title，如果为空的话，默认为“验证码登陆”
*/
+ (void)messageRegister:(NSString *)navTitle;
```

免费使用用短信注册功能,为开发者降低投入入成本,为用用户降低注册/登录门门槛。
WeiboSDK.h中有如下方方法:
/*

第三方方调用用微博短信注册或者登陆
@param navTitle 为登陆页页navigationBar的title,如果为空的话,默认为“验证码登陆” */

+ (void)messageRegister:(NSString *)navTitle;

12. 私信分享

```
WBMessageObject *message = [WBMessageObject message];
WBWebpageObject *webpage = [WBWebpageObject object];
webpage.objectID = @"";
webpage.title = NSLocalizedString(@"分享网页标题", nil);
webpage.description = [NSString stringWithFormat:NSLocalizedString(@"分享网页内容简介-%.0f", nil), [[NSDate date
    ] timeIntervalSince1970]];
webpage.thumbnailData = [NSData dataWithContentsOfFile:[[NSBundle mainBundle] pathForResource:@"image_2"
    ofType:@"jpg"]];
webpage.webpageUrl = @"http://tech.sina.com.cn/i/2015-11-19/doc-ifxkwuxx1517374.shtml";
message.mediaObject = webpage;

WBShareMessageToContactRequest *request = [WBShareMessageToContactRequest requestWithMessage:message];
request.userInfo = @{@"SendMessageFrom": @"SendMessageToWeiboViewController"};
[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。

注：以上场景均可在Demo源码中找到相应代码片段