

微博iOS平台SDK文档

编号: WEIBO_IOS_SDK

版本: WEIBO_IOS_SDK V3.1.2

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

Weibo SDK	2
一.SDK接入设置	2
二.应用场景代码示例	6
9.4.1 生成待签名的字符串	15
9.4.1.1 需要参与签名的参数	15
9.4.2 签名	15

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。

应用平台：☐ iPhone ☐ Android ☐ BlackBerry
☐ Windows Phone ☐ Symbian ☐ WebOS
☐ 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：



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：



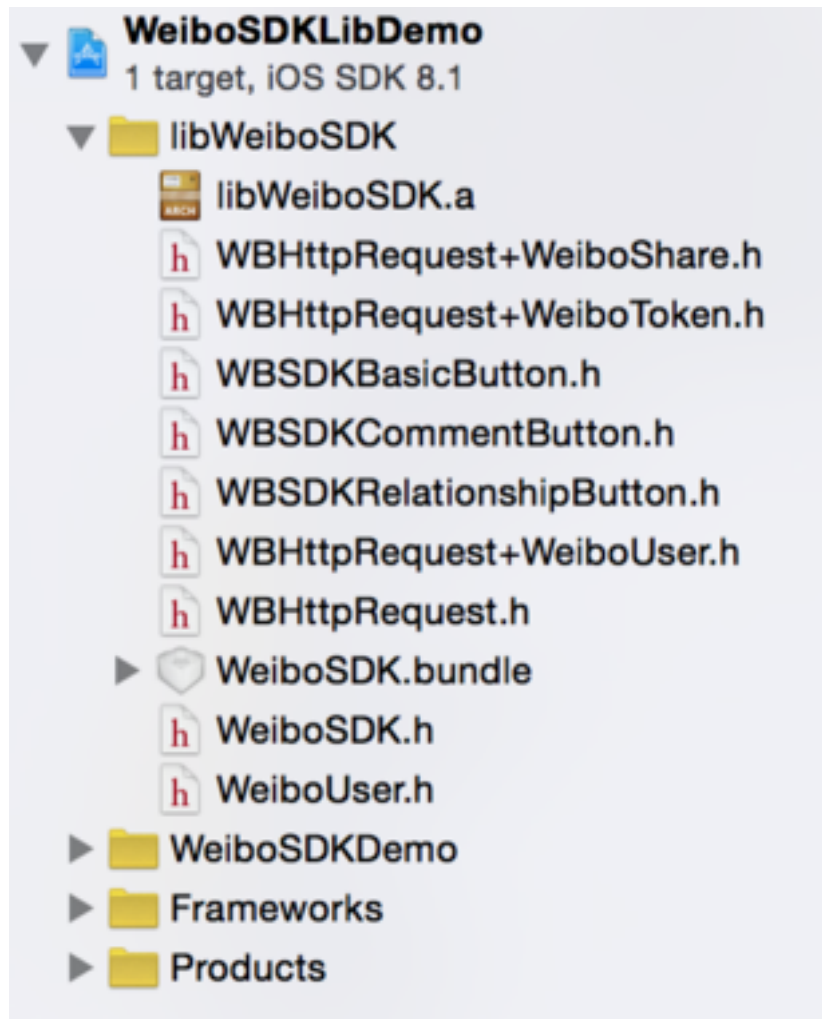
4. 设置工程回调URL Scheme

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

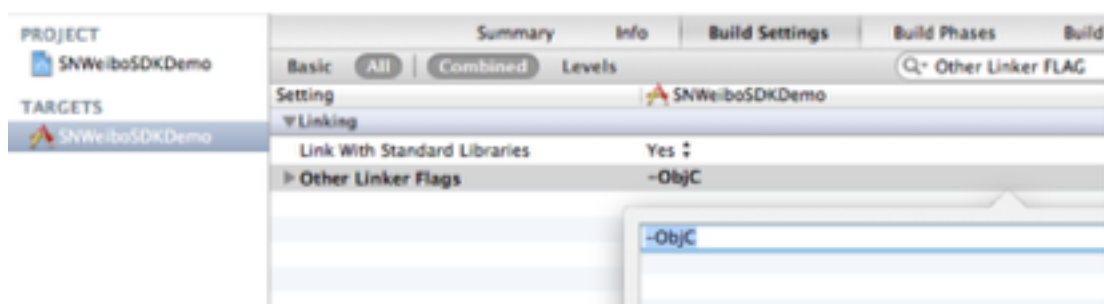


5. 添加SDK文件到工程

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。

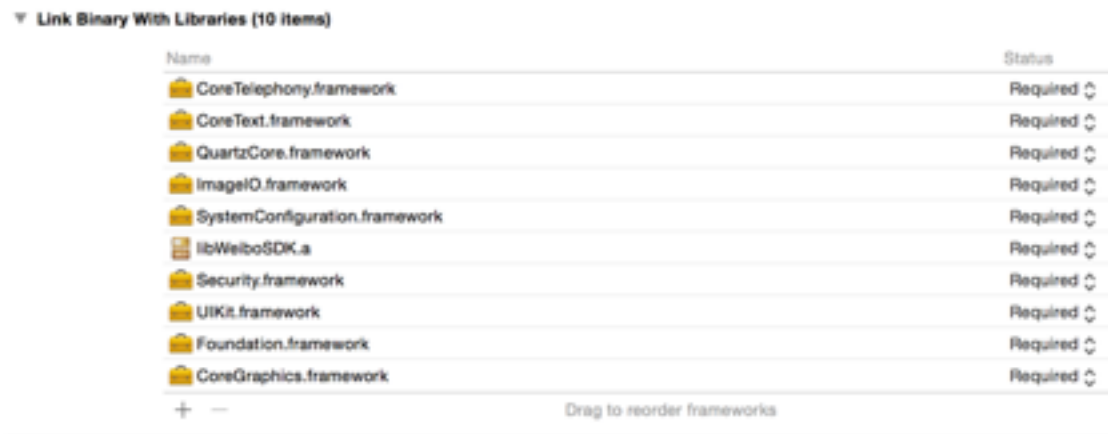


6. 在工程中引入静态库之后,需要在编译时添加 `-objc` 编译选项,避免静态库中类加载不全造成程序崩溃。方法:程序 Target->Build Settings->Linking 下 Other Linker Flags 项添加-ObjC。



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->Build 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:@"%s", 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:@"%s", 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。

```
/*
 *method
 *abstract
 *Creates a request representing a Open API call to the "friendships/followers".
 *discussion
 *Simplifies preparing a request and sending request to retrieve the user's followers.
 *A successful Open API call will return an NSDictionary of objects which contains an array of <WeiboUser> objects representing the
 *user's followers.
 *You can see more details about this API in http://open.weibo.com/wiki/2/friendships/followers/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 *)requestForFollowersListOfUser:(NSString *)currentUserID
withAccessToken:(NSString *)accessToken
andOtherProperties:(NSDictionary *)otherProperties
queue:(NSOperationQueue *)queue
withCompletionHandler:(WBRequestHandler)handler;
```

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,S D K 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。

```

@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

```

WBOderObject *order = [[[WBOderObject alloc] init] autorelease];
[order setOrderString:@"out_trade_no=13951324772254pr1ce=0.01&sign_type=md5&lockCheck&ind=1&sign=70091c901e3d75da14fb2583ad59e24f"];

WBPaymentRequest *request = [WBPaymentRequest requestWithOrder:order];
[WBSDK 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

参数	参数说明	类型	是否必填	备注
source	AppKey	String	是	在微博开放平台申请应用时分配的AppKey，调用接口时候代表应用的唯一身份
seller_id	商户微博UID	int	是	
sign_type	签名方式	String	是	目前签名仅支持md5
sign	签名	String	是	根据url参数以及密钥生成, 详见“3.签名机制”

参数	参数说明	类型	是否必填	备注
notify_url	后台回调地址	String	是	用来通知商户支付结果，判断支付成功请仅以此接口为准，最长255位
out_trade_no	商户订单号	String	是	商户网站唯一订单号，最长64位
subject	商品名称	String	是	该参数最长为128个汉字
pay_type	支付业务类型	int	是	根据业务规划给出参数列表，如1=实物商品售卖，2=虚拟商品售卖，3=捐助，4=权益等
price	商品单价	float	否	单位为 RMB-Yuan. 取值范围 [0.01,9999999999.00],精确到小数点后俩位. 规则：price和quantity能替代total_fee。即存在price和quantity 就不能存在total_fee；存在total_fee就不能存在price和quantity
quantity	购买数量	int	否	规则：price和quantity能替代total_fee。即存在price和quantity 就不能存在total_fee；存在total_fee就不能存在price和quantity
total_fee	交易金额	float	否	单位为 RMB-Yuan.取值范围 [0.01,9999999999.00],精确到小数点后俩位
body	商品描述	String	否	该参数最长为500个汉字
show_url	页面回跳地址	String	否	用来通知商户支付结果，用户在网页支付成功页面，点击“返回商户”时的回跳地址，最长255位
it_b_pay	超时时间	String	否	交易请求自创建时间起，自动关闭的时间，取值范围：1m(分钟)、1h（小时）、1c(当天0点)、15d(天)，不接受小数点，默认3d
is_route	必传值为1	String	是	
req_channel	请求渠道	String	是	1：跨端支付

参数	参数说明	类型	是否必填	备注
extra	额外参数	String	否	发起支付时透传的参数，可为不包含"="、"&"等特殊字符的字符串

9.3支付接口错误返回码说明

返回错误码 error_code	含义 error	说明
100000	Succeeded	成功
220001	操作失败	操作失败
220002	非法请求	非法请求
220003	请登录	登录失败
220004	验证签名失败	验证签名失败
220005	xxxx	参数验证错误,以error描述为准
220006	无效的订单号	无效的订单号
220007	用户信息错误	用户信息错误
220008	无效的uid	无效的uid
220009	无效的价格	无效的价格
220010	无效id	无效id
220011	无效time	无效time
220012	无效平台	无效平台
220013	数据已生成	数据已生成
220014	数据为空	数据为空
220015	无效支付渠道	无效支付渠道

返回错误码 error_code	含义 error	说明
220016	无效操作	无效操作
220017	无效商家	无效商家

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两个参数外，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两个参数外，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：&，@等），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，待签名数据应该是原生值而不是encoding之后的值。

9.4.2 签名

9.4.2.1 MD5签名

在MD5签名时，需要私钥参与签名。MD5的私钥是以英文字母和数字组成的字符串。商户提交完资料审核后，会随同开发文档等一起发送到指定邮箱。

9.4.2.2 请求时签名

当拿到请求时的待签名字符串后，需要把私钥直接拼接到待签名字符串后面，形成新的字符串，利用MD5的签名函数对这个新的字符串进行签名运算，从而得到32位签名结果字符串（该字符串赋值于参数sign）。

9.4.2.3 通知返回时验证签名

当获得通知返回时的待签名字符串后，同理，需要把私钥直接拼接到待签名字符串后面，形成新的字符串，利用MD5的签名函数对这个新的字符串进行签名运算，从而得到32位签名结果字符串。此时，这个新的字符串需要与微博支付通知返回参数中的参数sign的值进行验证是否相等，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 *)access_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;
```

免费使用短信注册功能,为开发者降低投入成本,为用户降低注册/登录门槛。

WeiboSDK.h中有如下方法：

```
/*
```

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;
```

12. 私信分享

```
WBMessageObject *message = [WBMessageObject message];
WBWebpageObject *webpage = [WBWebpageObject object];
webpage.objectID = @"";
webpage.title = NSLocalizedString(@"分享内容", nil);
webpage.description = [NSString stringWithFormat:NSLocalizedString(@"分享内容-%d", nil), [[NSDate date]
    timeIntervalSince1970]];
webpage.thumbnailData = [NSData dataWithContentsOfFile:[NSBundle mainBundle] pathForResource:@"image_2"
    ofType:@"jpg"];
webpage.webpageUrl = @"http://tech.sina.com.cn/1/2015-11-19/doc-1fxkwx1517374.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