

帷千广告 iOS SDK 安 装手册

iOS SDK v. 3.2.0

目 录

1	简介.....	2
2	帷干广告 SDK 嵌入.....	2
2.1	SDK 嵌入步骤.....	2
2.2	Banner 广告的使用.....	4
2.3	插播广告的使用.....	6
2.4	视频广告.....	8
2.5	推荐墙广告的使用.....	8
2.6	获取正式广告.....	9
3	远程管理.....	9
3.1	广告栏位申请.....	9
3.2	管理广告栏位.....	9
3.3	应用程序报表.....	10
4	常见问题.....	10
4.1	兼容 iOS 系统版本.....	10
4.2	广告没有成功展示问题.....	10



1 简介

欢迎使用 WQMobile iOS SDK。本着更方便开发者嵌入和使用帷千 SDK 的原则，WQMobile iOS SDK 3.0.0 之后版本在集成和使用方式上都有较大的变动，因此无论是否是第一次使用 WQMobile SDK，请仔细阅读以下内容。为了能够正常接收 WQMobile 的广告，请严格按照以下步骤嵌入 SDK。文档中的示例代码段都可以在 WQAdSample 工程中找到。

特别注意:一个应用中只能使用唯一的 AccountKey，并且保证这个 Key 是您在官网上和应用包名绑定过的；不同类型的广告请在帷千官网申请不同的 AdSlot。

2 帷千广告 SDK 嵌入

2.1 SDK 嵌入步骤

2.1.1 添加 WQMobile iOS SDK 到工程























将 SDK 包中的 WQAdSDK 文件夹内的文件都加入项目。



2.1.2 添加 library

请为项目加入以下框架（本 SDK 只支持 iOS 6.0 及以上版本）：

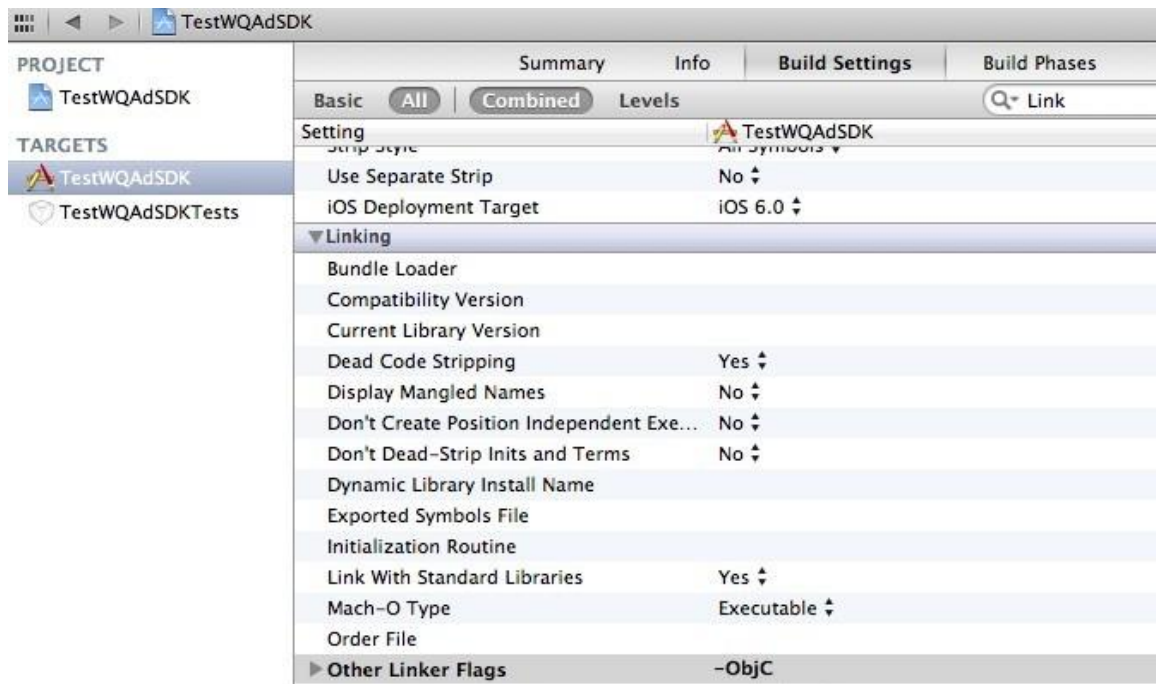


Name	Status
 libsqlite3.dylib	Required ▲▼
 StoreKit.framework	Optional ▲▼
 UIKit.framework	Required ▲▼
 SystemConfiguration.framework	Required ▲▼
 QuartzCore.framework	Required ▲▼
 MobileCoreServices.framework	Required ▲▼
 MessageUI.framework	Required ▲▼
 MediaPlayer.framework	Required ▲▼
 MapKit.framework	Required ▲▼
 libz.dylib	Required ▲▼
 libsqlite3.0.dylib	Required ▲▼
 Foundation.framework	Required ▲▼
 EventKit.framework	Required ▲▼
 EventKitUI.framework	Required ▲▼
 CoreMotion.framework	Required ▲▼
 CoreLocation.framework	Required ▲▼
 CoreGraphics.framework	Required ▲▼
 CoreFoundation.framework	Required ▲▼
 CFNetwork.framework	Required ▲▼
 AdSupport.framework	Optional ▲▼
 CoreTelephony.framework	Required ▲▼
 WQADSDK.a	Required ▲▼

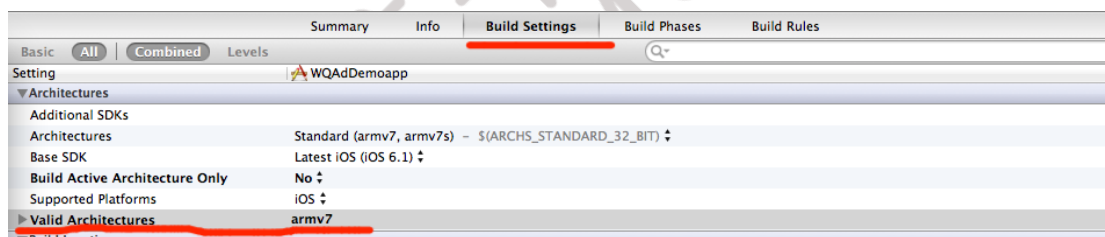
+ - Drag to reorder frameworks

2.1.3 项目配置

在项目信息中，编译的选项卡，找到 Linking->Other Linker Flags，添加内容-ObjC：



再找到 Valid Architectures，改成图中所示：



2.2 Banner 广告的使用

通过代码的方式创建 WQMobileAdView。在代码中添加广告，设置 ADSLOT 和 ACCOUNT_KEY，在需要显示广告的 View 的 ViewController 中，添加委托 WQAdViewDelegate（可选）。添加成员：

```
@interface WQViewController : UIViewController <WQAdViewDelegate>{  
  
    WQAdView* myAdView;  
  
}
```

在创建 View 的时候，添加以下代码

```
myAdView = [[WQAdView alloc] init] autorelease];           //创建广告控件  
  
myAdView.storeKitEnabled = YES;    //该参数为 YES 时 iOS 6.0 及以上系统会在应用内打开 App
```



Stroe 界面；如果您的应用是越狱的，建议把该参数设置为 false

(若你的应用提供位置服务，建议使用myAdView = [[WQAdView alloc] initWithLocation:autorelease]，将使你的广告价值最大化。)

```
[myAdView setFrame:CGRectMake(0, 0, 320, 480)]; //设置大小
```

//通过使用提供的AdSlotID和AccountKey在指定的controller中开始获取广告（在这个方法里，最后一个参数，即：InViewController最好不为空，因为为空时，有时候会导致点击广告没反应；这个参数是用来点击广告的时候作为presentingViewController来present显示广告的controller的）

```
- (void)startWithAdSlotID:(NSString *)adslotid AccountKey:(NSString *)accountKey  
InViewController:(UIViewController *)controller; //开始获取广告
```

```
[self.view addSubview:myAdView]; //将广告控件添加到视图
```

完成以上步骤之后，广告视图就成功的添加到应用程序中，当应用程序中放有广告视图的界面显示后，SDK 自动请求广告,成功获取广告后会自动出现在界面上。在需要停止View 的时候，通过以下代码从父视图中删除 View

```
[myAdView removeFromSuperview]; //从父视图中删除
```

监听广告视图界面

当指定的 AdView 请求广告成功或者失败后,该接口 WQAdViewDelegate 会通知监听者。该接口的设计是为了方便那些有需要在广告条展示广告时进行一些界面调整的开发者。

//广告视图获取广告成功

```
- (void)onWQAdReceived:(WQAdView *)adview;
```

//广告视图获取广告失败

```
- (void)onWQAdFailed:(WQAdView *)adview;
```

//广告视图将要关闭

```
- (void)onWQAdDismiss:(WQAdView *)adview;
```

//广告视图获取缓慢提醒

```
- (void)onWQAdLoadTimeout:(WQAdView*) adview;
```



2.3 插播广告的使用

插播广告分成两种类型：全屏插播和半屏插播。两种类型的广告区别在于插播全屏是弹出式的广告，插播半屏是嵌入式的广告，半屏对应的广告尺寸有 300X250、600X500，您可以根据您的应用类型和使用场景选择对应的广告形式。具体使用方式见以下步骤：

- 1) 导入 WQInterstitialAdView.h 头文件；
- 2) 声明一个 WQInterstitialAdView 的实例；
- 3) 让您的 UIViewController 实现 WQInterstitialAdViewDelegate 协议；
- 4) 在您的 Controller 的 m 文件中,设置 ADSLOT 和 ACCOUNT_KEY 以及负责展示插播广告的 UIViewController 来初始化一个 WQInterstitialAdView；
- 5) 在初始化广告控件后，尽快调用 loadInterstitialAd 方法,来预加载插播广告；
- 6) 在想要展示插播广告的时候，调用 showInterstitialAd 方法,来展示插播广告；
- 7) 在展示完成后，建议重新通过 loadInterstitialAd 方法,加载一条新广告,用于下次展示。

下面的代码，简单的实现了上面描述的过程：

```
@property(retain, nonatomic) WQInterstitialAdView *_adView;

//SDK支持 iOS 5.0 及以上版本，调用其中的类之前需要判断这个类在当前设备是否可用，不用获取系统版本信息，只需要像以下代码一样调用class方法看其是否返回非空则可
if ([WQInterstitialAdView class])
{
    WQInterstitialAdView *tAdView = [[WQInterstitialAdView alloc] initWithFrame:CGRectMake(0, 0, 300, 250) andAdSlotID:kViewControllerWQAdSlotIDString andAccountKey:kViewControllerWQADAccountKeyString];

    tAdView.storeKitEnabled = YES; //该参数为 YES 时，iOS 6.0 及以上系统会在应用内打开 App Store 界面；如果您的应用是越狱的，需要把该参数设置为 false
    tAdView.delegate = self;
    self._adView = tAdView;
    [tAdView release];
}

//在需要展示插播广告之前，先通过isInterstitialAdReady方法检查广告是否准备就绪

if ([self._adView isInterstitialAdReady])
```



```
{  
  
    //展示广告  
  
    [self._adView showInterstitialAd];  
  
} else {  
  
    //如果广告没有就绪，调用loadInterstitialAd  
  
    [self._adView loadInterstitialAd];  
  
}
```

注：插播全屏的初始化必须为以下方式，否则可能导致应用审核不通过

```
CGRect appFrame = [[UIScreen mainScreen] applicationFrame];  
WQInterstitialAdView *tAdView = [[WQInterstitialAdView alloc] initWithFrame:  
appFrame andAdSlotID:kViewControllerWQADSlotIDString andAccountKey:kViewControllerWQ  
ADAccountKeyString];
```

实现 WQInterstitialAdViewDelegate:

您可以通过实现 WQInterstitialAdViewDelegate 中定义的方法,来跟踪广告生命周期中的各个阶段。所有这些方法也都定义在 WQInterstitialAdView.h 中,如下:

```
@protocol WQInterstitialAdViewDelegate <NSObject>  
  
    //当插播广告被成功加载后，回调该方法，pAllLoaded 为 YES，表示所有的广告已经加载完成（一次广告加载可能会加载多条广告）  
  
    -(void) onInterstitialAdRequestLoaded:(WQInterstitialAdView*)pAdView  
    allLoaded:(BOOL)pAllLoaded;  
  
    //当插播广告被加载失败后，回调该方法  
  
    -(void) onInterstitialAdRequestFailed:(WQInterstitialAdView*)pAdView;  
  
    //当插播广告要展示出来前，回调该方法  
  
    -(void) onInterstitialAdPresentPresent:(WQInterstitialAdView*)pAdView;  
  
    //当插播广告被关闭后，回调该方法
```




```
-(void) onInterstitialAdDismiss:(WQInterstitialAdView*)pAdView;

//当要呈现 Modal View 时，回调该方法，如打开内置浏览器

-(void) wqInterstitialAdWillPresentModalView:(WQInterstitialAdView*)pAdView;

//当呈现的 Modal View 即将关闭时，回调该方法，如关闭内置浏览器

-(void) wqInterstitialAdDidDismissModalView:(WQInterstitialAdView*)pAdView;

//打开内置浏览器时，回调改方法，点击广告的时候作为 presentingViewController 来 present 显示
广告的 controller

-(UIViewController*)
controllerForPresentingModelViewInInterstitialAdView:(WQInterstitialAdView*)pAdView;
```

2.4 视频广告

帷千 SDK 支持视频广告。视频广告是插播广告的一种。如果需要植入视频广告或者体验视频广告效果请与帷千的客户代表联系。

为更好兼容各种广告形式,需要在广告被隐藏或者遮挡时调用 `pauseVideo` 方法停止广告播放;在某些情况下需要暂停视频广告播放,当恢复到正常时,可以调用 `continueVideo` 方法让视频广告继续播放。

2.5 推荐墙广告的使用

推荐墙广告有两种嵌入方式:

(1) 自定义推荐按钮，需要在官网申请推荐按钮广告类型对应的 ADSLOT 和 ACCOUNT_KEY，在触发展示推荐墙的事件处理函数添加以下代码：

```
NSString *tAdSlotIDStr = @"7657e2ff65e7649b837407f9f1494116";

NSString *tKeyStr = @"c73d49b0e4a05f3993e4b25eee44623f";

[WQAdView openAdWallWithAdSlotId:tAdSlotIDStr accountKey:tKeyStr
inViewController:self withStoreKitEnabled: YES];
```



当触发事件发生时,就可以直接打开应用推荐墙界面。

(2) 帷千推荐按钮,添加方式和 Banner 类型广告视图一样,只需将 Banner 广告视图中 ADSLOT 和 ACCOUNT_KEY 修改成推荐墙广告 ADSLOT 和 ACCOUNT_KEY,就可以请求到帷千推荐按钮广告,点击推荐按钮,即可进入应用推荐墙界面。

2.6 获取正式广告

在应用通过审核前,对应的正式 AccountKey 和 AdSlot 只能展示测试广告。请于当日 16 时之前提交应用程序,帷千工作人员会在 24 小时之内完成审核工作,审核通过之后,对应的正式 AccountKey 和 AdSlot 才够获取正常的付费广告。。任何时候,都可以使用帷千公共测试 AccountKey 和 AdSlot 来测试帷千的广告效果及兼容性。使用公共测试 AccountKey 和 AdSlot 的应用在任何时候都只展示测试广告。

3 远程管理

3.1 广告栏位申请

- 1) 登陆帷千官网 www.wqmobile.com, 注册账户;
- 2) 登陆账户,进入“应用程序管理”页面,选择“新建应用程序”;
- 3) 填写应用程序信息;
- 4) 设置广告位: 点击“创建广告栏位”,填写广告栏位名称,选择广告类型,点击“保存并生成广告栏位 ID”,生成广告栏位对应 ID 等信息;将对应信息连同 SDK 植入至应用程序中并生成安装包;
- 5) 上传第(4)步中生成的安装包,等待审核;
- 6) 等待审核时可获取到测试广告,审核通过后方可获取付费广告。

注:除了横幅和推荐墙这两种广告类型之外,其他类型的广告嵌入请联系客服。

3.2 管理广告栏位

一个应用程序可以拥有一个或多个广告栏位,每个广告栏位都对应其唯一的 ID,每个账号对应一个发布 ID (即 KEY)。创建广告栏位并完成保存后将在每个广告栏位的信息中出现对应的 ID。



3.3 应用程序报表

可通过帷千平台上的应用程序报表查看应用程序广告播放的相关实时数据。具体操作流程：

- 1) 登陆帷千平台，选择“报表管理”中的“应用程序报表”；
- 2) 选择“创建新报表”，选填相关信息项保存后可查看所选的应用程序相关实时数据。

4 常见问题

4.1 兼容 iOS 系统版本

WQMobile iOS SDK 兼容 iOS 系统 6.0 及以上版本，对于 6.0 以下版本可能会有兼容性问题。

4.2 广告没有成功展示问题

在添加广告视图时，请避免以下情况，否则将影响广告的获取和展示：

- 1) 广告视图被其他任何视图遮挡；
- 2) 广告视图的范围在屏幕可视范围之外；
- 3) 广告视图为透明或隐藏；