



# mobiSage\_iOS\_SDK

[使用说明]

SDK Version: mobiSage\_iOS\_SDK\_6.3.0

2014-6

## 目 录

1. SDK 主要功能指南.....	3
.1 导入 SDK .....	3
.2 配置.....	3
.3 申请并设置 PublishID & SlotToken .....	4
.4 横幅广告.....	6
.5 插屏广告.....	8
.6 开屏广告.....	10
.7 信息流广告.....	12
2. 使用建议 .....	14

## 1. SDK 主要功能指南

### . 1 导入 SDK

访问 <http://mobi.adsage.cn/sdk>，下载 SDK 包，解压后按照如下步骤在工程中引入 SDK：

右键点击工程，选择 Add File to...，选择解压 SDK 包后得到的 Mobisage 的文件夹（其中包括 minizip 开源文件夹、libmobisageSDK.a、MobiSageSDK.h 和 MobiSageNative.h），点击 Add。或者将文件拖入 XCode 工程目录结构中，在弹出的界面中勾选“Copy items into destination group's folder(if needed)”，并确保 Add To Targets 勾选相应的 target。

### . 2 配置

按照以下步骤添加依赖框架（framework）：

点击工程，选择 Targets->Build Phases->Link Binary With Libraries->Add Items” +”，选择以下 framework->Add

```
Foundation.framework//Required
Security.framework//Required
CFNetwork.framework // Required
MediaPlayer.framework //Required
UIKit.framework//Required
CoreGraphics.framework//Required
QuartzCore.framework//Required
SystemConfiguration.framework//Required
CoreLocation.framework//Required
CoreTelephony.framework // Required
MessageUI.framework // Required
AdSupport.framework //Optional
StoreKit.framework //Optional
PassKit.framework //Optional
CoreMotion.framework //required
MobileCoreServices.framework // required
AddressBook.framework // required
QuartzCore.framework //required
CoreAudio.framework //required
CoreImage.framework // required
AudioToolbox.framework // required
```

```
AVFoundation.framework // required
ImageIO.framework // required
CoreMedia.framework // required
QuartzCore.framework // required
```

### .3 申请并设置 PublishID & SlotToken

访问 <http://mobi.adsage.cn>, 注册用户并创建 App, 将获取到每个 App 的 Publish ID, 一个应用程序可以嵌入多种形式的广告, 但需要使用同一个 Publish ID; 一个 Publish ID 只能绑定一个应用程序; 如需发布多个程序, 请分别获取各自的 Publish ID。

**注意:** 请在初始化广告对象前设置 Publish ID

#### 3.1 设置应用信息

#### 3.2 创建广告位

InApp列表 > 新建InApp

1 设置应用信息

2 创建广告位

3 下载sdk并上传应用

特别注意：  
设置完广告位信息后，请用生成的广告位ID去配置SDK。如果您的应用没有创建广告位，将接收不到艾德思奇广告平台的广告。

+ 新建广告位

名称	类型	状态	广告位ID	创建时间	操作
抱歉，没有找到相关记录					

展示第0条记录，共计0条

首页 上一页 下一页 末页

保存继续 取消

特别注意：  
设置完广告位信息后，请用生成的广告位ID去配置SDK。如果您的应用没有创建广告位，将接收不到艾德思奇广告平台的广告。

+ 新建广告位

名称	类型	状态	广告位ID	创建时间	操作
信息流广告	信息流广告位	运行	k5KTQPwEHomgh7XJFABlrh+9	2014-06-23 13:44:37	编辑 删除
荐计划广告	荐计划广告位	运行	19bXBLhAWs3kw/GNUEQM6lv0	2014-06-23 13:44:30	编辑 删除
开屏	开屏广告位	运行	w8LDEKxUTnw1+WZRFAY/k/n	2014-06-23 13:43:58	编辑 删除
全屏	全屏广告位	运行	TE1MnyPbwVZ/WGoWy9+XccBr	2014-06-23 13:43:50	编辑 删除
横幅	横幅广告位	运行	hYSFVuoSCJ+2kaPfAhZeuAmj	2014-06-23 13:43:39	编辑 删除

当前显示第 1 条 到 第 5 条数据，总记录数：5 条

首页 上一页 1 下一页 末页

3.3 下载 SDK 并上传应用

InApp列表 > 编辑InApp

编辑InApp

Publish ID : zczNHqJZQNf+2euXSg==

App名称：决战南非

设备类型：

App类型：体育竞速

App地理位置信息：☒ 获取地理位置

App描述：世界杯信息实时更新，比赛时间及结果分享与分析！

```
[[MobiSageManager getInstance]setPublisherID:@"zczNHqJZQNf+2euXSg=="
deployChannel:@"mobisage"];
```

## 4 横幅广告

### ➤ 创建横幅广告

通过下面代码创建横幅广告：

```
//实例化横幅广告，并设定广告大小，slotToken_ID 广告位 ID
MobiSageBanner* adBanner = [[MobiSageBanner alloc] initWithDelegate:self
                                adSize:Default_size
                                slotToken:slotToken_ID];

//设置横幅广告的位置
[adBanner setFrame:CGRectMake(0, 100, 320, 50)];

//显示横幅广告
[self.view addSubview:adBanner];

//释放 （非 ARC 状态下使用）
[adBanner release];
```

横幅广告提供以下尺寸：

广告尺寸	枚举值	适用设备
320X50(推荐)	Banner_iphone	iPhone
728X90(推荐)	Banner_ipad	iPad
自适应	Default_size	iPad/iphone

### ➤ 设置横幅广告需要的 UIViewController 及回调方法

在包含横幅广告的 UIViewController 中，需要声明遵循 MobiSageBannerDelegate 协议。

实现方式如下所示：

```
=====
//ViewController.h 文件
#import "MobiSageSDK.h"

@interface ViewController : UIViewController<MobiSageBannerDelegate>
{}
@end
=====
```

```
@optional

/**
 * 描述：当SDK需要弹出自带的Browser以显示mini site, in app purchase时需要使用当前
广告所在的控制器。
 * 返回：一个视图控制器对象
 * 说明：如果没有实现此回调，将使用keyWindow.rootViewController
 */
- (UIViewController*)viewControllerToPresent{
    return self;
}

//横幅广告成功展示时，触发此回调方法，用于统计广告展示数
- (void)mobiSageBannerSuccessToShowAd: (MobiSageBanner*)adBanner {
}

//横幅广告被点击时，触发此回调方法，用于统计广告点击数
- (void)mobiSageBannerClick: (MobiSageBanner*)adBanner {
}

//横幅广告展示失败时，触发此回调方法
- (void)mobiSageBannerFaildToShowAd: (MobiSageBanner*)adBanner withError: (NSError *)error{
}

//横幅广告点击后，打开 LandingSite 时，触发此回调方法，请勿释放横幅广告
- (void)mobiSageBannerPopADWindow: (MobiSageBanner*)adBanner {
}

//关闭 LandingSite 回到应用界面时，触发此回调方法
- (void)mobiSageBannerHideADWindow: (MobiSageBanner*)adBanner{
}

@end
```

注意事项：请在销毁广告对象前，把对象的 delegate 置为 nil: adBanner.delegate = nil

## . 5 插屏广告

### ➤ 创建插屏广告

通过调用下面的方法来实例化插屏广告：

广告参数：Float\_size\_0 // for iPhone iTouch 300\*250 iPad 600\*500

Float\_size\_3 // for iPhone iTouch 320\*480 iPad 640\*960

//实例化插屏广告，

```
MobiSageFloatWindow *floatWindow = [[MobiSageFloatWindow alloc] initWithAdSize:Float_size_0
                                     delegate:self
                                     slotToken:slotToken_ID];
```

//或（默认 Flot\_size\_0）

```
MobiSageFloatWindow *floatWindow = [[MobiSageFloatWindow alloc] initWithDelegate:self
                                     slotToken:slotToken_ID];
```

### ➤ 展示插屏广告

通过调用下面方法显示插屏广告

//建议在广告预加载成功的回调方法中调用

```
[floatWindow showAdvView];
```

### ➤ 设置插屏广告需要的 UIViewController 及回调方法

在弹出插屏广告的 UIViewController 中，需要声明遵循 MobiSageFloatWindowDelegate 协议。

//ViewController.h 文件

```
#import "MobiSageSDK.h"
```

```
@interface ViewController : UIViewController<MobiSageFloatWindowDelegate >
```

```
{
}
```

```
@end
```

```
@optional
```

//插屏广告预加载成功时，触发此回调方法，建议在此回调方法触发之后，再调用插屏广告的展示方法

```
- (void)mobiSageFloatSuccessToRequest:(MobiSageFloatWindow*)adFloat{
}
```

//插屏广告被点击时，触发此回调方法，多用于聚合中统计广告点击数



```

- (void)mobiSageFloatClick: (MobiSageFloatWindow*)adFloat {
}

//插屏广告预加载失败时，触发此回调方法，多用于聚合
- (void)mobiSageFloatFailedToRequest: (MobiSageFloatWindow*)adFloat withError: (NSError *)error{
}

//插屏广告被关闭后回到应用界面时，触发此回调方法
- (void)mobiSageFloatClose: (MobiSageFloatWindow*)adFloat{
}

@end

```

注意事项：请在销毁广告对象前，把对象的 delegate 置为 nil: floatWindow.delegate = nil

## . 6 开屏广告

### ➤ 创建开屏广告

开屏广告的使用场景是在应用刚刚开启时,在 SDK 内分为**缓存开屏**和**实时开屏**两种。缓存开屏的加载机制为,本次加载,缓存本次的广告应答以及相关资源,下次开启时展现。而实时开屏在每次开启时都会实时请求广告并加载和缓存资源,在超时时间前加载完成即自动展现。详细使用方法可见 Sample 中的 App 委派类的 application:didFinishLaunchingWithOptions: 方法中使用的示例代码。

### ➤ 通过调用下面的方法来实例化**缓存开屏**广告:

```
// 实例化开屏广告, 建议在应用 didLaunchFinished 方法中调用
// Orientation 开屏广告展示方向
// background 应用启动画面图片
// delegate 广告所使用的委托
self.adSplash = [[MobiSageSplash alloc] initWithOrientation:MobiSage_Orientation_Portrait
                                     background:backgroundImg
                                     withDelegate:self
                                     slotToken:slotToken_ID];

[adSplash startSplash];
```

### ➤ 通过调用下面的方法来实例化**实时开屏**广告:

```
//实例化开屏广告, 建议在应用 didLaunchFinished 方法中调用
//Orientation 开屏广告展示方向
//background 应用启动画面图片
// delegate 广告所使用的委托
MobiSageAdRTSplash* adSplash = [[MobiSageAdRTSplash alloc]
                                initWithOrientation:MobiSage_Orientation_Portrait
                                background:backgroundImg
                                withDelegate:self
                                slotToken:slotToken_ID];

[adSplash startSplash];
```

开屏广告提供以下尺寸：

广告方向	枚举值	适用设备
竖屏	MS_Orientation_Portrait	iPhone/iPad
横屏	MS_Orientation_Landscape	iPad

注意事项：

1、请在开屏广告的整个生存期都持有 MobiSageSplash 对象的引用，释放该对象会导致内部逻辑异常，可以在广告关闭或者请求失败的回调中释放对象，请在销毁广告对象前，把对象的 delegate 置为 nil：  
`nil: self.adSplash.delegate = nil。`

2、在应用启动 didFinishLaunchingWithOptions 方法中首先创建开屏广告，在此时不要调用 `[self.window makeKeyAndVisible]`，否则会广告出现前提前进入应用的首界面，影响用户体验；

3、建议将 `[self.window makeKeyAndVisible]` 放在开屏广告展示失败或者关闭时调用；

4、设置的开屏广告方向需与应用的方向一致，否则会因为方向冲突引起应用的崩溃；

5、在初始化时，需地传入背景图片，将启动页面图片传入开屏广告的背景图片参数中；

➤ 设置开屏广告需要的回调方法

应用的 AppDelegate 中，需要声明遵循 MobiSageSplashDelegate 协议，并实现 MobiSageSplashDelegate 中定义的广告事件回调方法。

```
=====
//在 AppDelegate.m 中实现回调方法
@optional
//开屏广告展示成功时，回调此方法
- (void)mobiSageSplashSuccessToShow: (MobiSageSplash*)adSplash{
}
//开屏广告展示失败时，回调此方法，在此回调方法中，需释放广告，且在此时弹出应用的界面
- (void)mobiSageSplashFailedToRequest: (MobiSageSplash*)adSplash withError: (NSError *)error{
    [adSplash release]; （非 ARC 状态下使用）
    [self.window makeKeyAndVisible];
}
//开屏广告失败时，回调此方法，需释放广告，且在此时弹出应用的界面
- (void)mobiSageSplashClose: (MobiSageSplash*)adSplash{
    [adSplash release]; （非 ARC 状态下使用）
    [self.window makeKeyAndVisible];
}
@end
```

## . 7 信息流广告

信息流广告与普通横幅广告的差别除了可自定义尺寸外,主要的差别在于信息流广告会返回广告本身的一些信息。目前包含的信息有:广告宽度,广告高度(注:不是请求的宽高而是返回广告的宽高,服务器会根据开发者请求的宽高返回接近的尺寸,此处获得的即为实际尺寸),广告标题,广告 logo 网址 (为一张图片), image, 广告描述信息, 广告星级, 广告包大小。

### ➤ 创建信息流广告

通过调用下面的方法来实例化信息流广告:

```
// 实例化信息流广告
//根据 width(或 height) 请求广告. (mobisageNativeSuccessToRequest 请求成功后, 返回实际的 height(或 width))
adView = [[MobiSageNative alloc] initWithWidth:(self.view.frame.size.width-20)
        slotToken:slotToken_ID
        completion:^(CGSize size, NSDictionary *content) {
            [tableView reloadData];
        }];
adView.delegate=self;
```

注意事项:

1、根据 size 请求广告(宽高比在 0.5-2 之间为有效 size), 我们会根据有效 size 自适应填充广告内容

2、请在销毁广告对象前, 把对象的 delegate 置为 nil (例: `adView.delegate = nil`)

### ➤ 设置信息流广告需要的回调方法

应用的 AppDelegate 中, 需要声明遵循 MobiSageNativeDelegate 协议, 并实现 MobiSageNativeDelegate 中定义的广告事件回调方法。

```
=====
//在 AppDelegate.m 中实现回调方法
@optional

/**
 * 描述: 当SDK需要弹出自带的Browser以显示mini site, in app purchase时需要使用当前
广告所在的控制器。
 * 返回: 一个视图控制器对象
 * 说明: 如果没有实现此回调, 将使用keyWindow.rootViewController
 */
- (UIViewController*)viewControllerToPresent{
```

```

        return self;
    }
    //广告请求结果.
    -(void)mobiSageNativeSuccessToRequest:(MobiSageNative*) aNative
    {
        //可以根据 aNative.content 对象中的 width,height 设置广告位大小.
        // content 的内容详见 信息流广告元素内容
    }

    -(void)mobiSageNativeFailedToRequest:(MobiSageNative*) aNative withError:(NSError*) error
    {
        NSLog(@"请求失败%@", error);
    }

    //广告加载.
    -(void)mobiSageNativeSuccessToLoaded:(MobiSageNative*) aNative
    {
        NSLog(@"广告加载成功");
    }
    -(void)mobiSageNativeFailedToLoaded:(MobiSageNative*) aNative withError:(NSError*) error
    {
        NSLog(@"广告加载失败%@", error);
    }

    //广告 landingPage
    -(void)mobiSageNativePopADWindow:(MobiSageNative*) aNative
    {
    }
    -(void)mobiSageNativeHideADWindow:(MobiSageNative*) aNative
    {
    }

    //广告被点击
    -(void)mobiSageNativeClick:(MobiSageNative*) aNative
    {
    }

    //广告已展示
    -(void)mobiSageNativeAppeared:(MobiSageNative*) aNative
    {
    }

    @end

```

➤ 信息流广告元素内容

1. 根据 content 的内容你可以扩展广告的形式.
2. content 元素说明:

```

{
    id="10099" //广告的唯一标示,可以用来去重相同的广告
    height = 250;//广告的高
    width = 300;//广告的宽
    title = "信息流广告";//展示的标题
    desc = "";//展示描述
    image =
"http://mws.adsage.com/mobisage/bcadm/26496/44b9f6c78f64432d8ac8eb20993e7334.jpg";//广告展示图片
    logo =
"http://a1782.phobos.apple.com/us/r30/Purple/v4/39/8a/de/398ade94-d3b4-f60f-4861-9f49b318d2b7/72.png";//应用 logo
    packageSize = "7.3";//包大小, 单位 MB
    star = 0; //展示星级
}

```

#### ➤ 信息流广告点击

信息流广告单独提供了一个点击接口, 调用此方法就相当于点击了一次广告。因为信息流广告提供了一些广告信息供开发者自行拼装展现形式, 开发者可添加一个下载按钮, 点击此按钮调用点击方法即可:

```

//广告点击事件可以通过调用点击事件触发
[adView handleClick];

```

**注意事项:** content 元素包含的 id 是广告的唯一标示, 可以用来过滤相同的广告。

## 2. 使用建议

#### ➤ 注意事项:

开发及运行环境: 目前支持 iOS 5.0 的操作系统及 xCode4.5 的开发环境。

如果您的工程使用了 ARC 模式, 请在 Build Phases 中将相关资源设置为 -fno-objc-arc。

使用所有广告前应保证设置了正确的 publishID, 否则会导致广告创建失败, 返回 nil 指针。

初始化广告时请保证 rootViewController 已经初始化完毕, 故不可以在 rootViewController 的 init 或者 viewDidLoad 中初始化广告。

➤ **关于横幅广告的使用：**

横幅广告点击后，Landing Site 被打开时（触发 Landing Site 打开的回调方法），由于 Landing Site 会遮挡应用的界面，建议在此时暂停用户对应用的操作，待 Landing Site 关闭后（触发 Landing Site 关闭的回调方法），再继续用户对应用的操作。

在横幅广告所在的应用界面被遮挡或隐藏时，需将横幅广告释放掉，以防过多的无效广告请求和展示，影响广告点击率和收益。

➤ **关于插屏广告的使用：**

插屏广告支持广告的预加载，在广告被实例化时，会启动广告请求缓存线程，因此建议将插屏广告的实例化前置，这样在预加载成功的方法被回调时，调用广告的展示方法，可以提高广告的填充率。插屏广告在被用户关闭后，插屏广告会自动释放，无需在应用中对其再次进行释放。

➤ **关于开屏广告的使用：**

在使用开屏广告的时候，建议将传入的图片设置为应用的启动画面，并按照在开屏广告结束或请求失败时进入应用，否则有可能会在广告出现之前，进入应用的界面。

➤ **关于应用内 AppStore 的使用：**

应用还没有提交到 AppStore 时下载应用进行测试， 需要设置 `[mobisageManager showStoreInApp :NO]`；并且在线前去掉。（见 Sample）

如果碰到问题，如何联系我们？

答：参见 <http://mobi.adsage.cn/>，可以使用邮箱、电话、QQ、MSN 多种方式为开发者提供服务。

邮箱：mobi@adsage.com

电话：010-62662626-8294

客服：QQ：1497071580

M S N: mobiSage@live.com