

百度 MobileSSP

移动应用推广 SDK

用户手册 (iOS 3.2 版)

百度在线网络技术 (北京) 有限公司

(版权所有,翻版必究)

目录

一、	简介.....	3
二、	阅读对象.....	3
三、	版本支持.....	3
四、	工程设置.....	3
1	创建应用	3
2	嵌入 SDK.....	3
3	提交 Store	3
4	更新版本	4
5	应用 id 与代码位 id.....	4
五、	开发示例.....	5
1.	Banner 广告的开发示例.....	9
2.	插屏广告的开发示例	11
3.	开屏广告的开发示例	15
5.	视频广告的开发示例	20
6.	可能遇到的问题	22
六、	联系我们.....	23

一、 简介

百度移动应用推广 SDK(iOS 版)是百度推出的移动应用推广 SDK 在 iOS 平台上的版本 (以下简称 SDK)。该文档提供了对如何使用 SDK 的一个详细说明,建议阅读时下载我们的用例工程,代码是最好的使用说明。

如有其他问题可以参考网站的 FAQ,或者及时与我们联系。

二、 阅读对象

本文档面向所有使用该 SDK 的开发人员、测试人员以及对此感兴趣的其他用户。

三、 版本支持

iOS 5.0+

四、 工程设置

1 创建应用

请先在 apple itunes connect 里面创建一个应用,设置应用包名和版本,并且获取应用 itunes url。在百度 MobileSSP 网站中创建 iOS 版 APP 时填写正确的**应用包名**,**版本**和 **itunes url**,并且等待应用通过审核,百度 MobileSSP 的审核周期为 1-2 个工作日,请您耐心等待。(注意:错误的信息会导致后续审核失败)

2 嵌入 SDK

当您的 APP 通过审核以后,可以获取由 4.1 中创建的 APP ID (APP ID 的获取详见 附件 1),在下述的开发过程中填写,如果设置正确,你可以看到有效的广告展示。

3 提交 Store

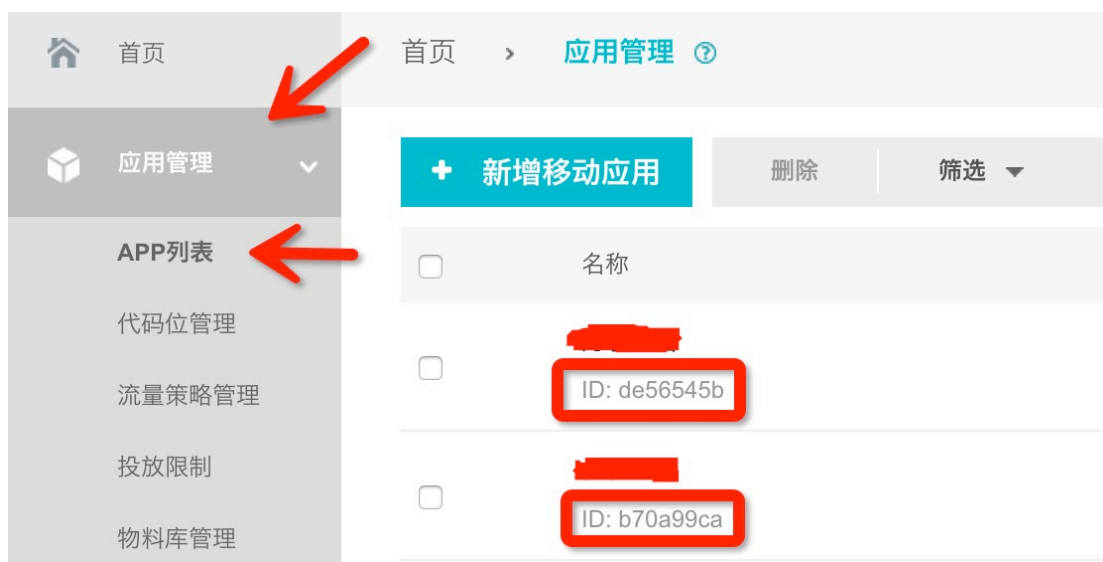
注意,该应用必须是 4.1 中 apple itunes connect 创建的那个 APP,并且已经正确的部署好百度 MobileSSP 的 SDK 及 APP ID,有广告展示。如有修改请同时修改百度

MobileSSP 网站中“应用管理”的 APP 信息，确保信息一致。此时您可以提交应用至 App Store 中发布您的应用。

4 更新版本

请在百度 MobileSSP 网站中添加一个新的应用及其版本，确保版本信息和 AppStore 的更新版本一致，并重新提交审核。重新提交审核时为了确保您在查看报告时能够区分查看不同版本的 APP 的详细数据。旧版本和新版本的应用可以同时接受广告。

5 应用 id 与代码位 id



上图所示为应用 id, 下图所示为代码位 id.



五、 开发示例

注：以下截图均在示例程序（ios_demo）中可以找到。

#warning:iOS9适配说明:

1.在plist中,添加baidumobads到LSApplicationQueriesSchemes数组下. 注意:Xcode7.0

模拟器上会error:(null),真机上不会报错,这里错误可以忽略.

```
<key>LSApplicationQueriesSchemes</key>
```

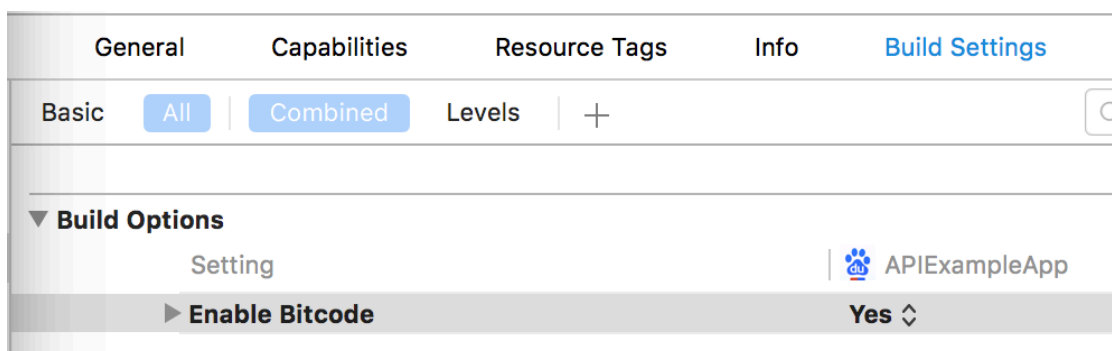
```
<array>
```

```
<string>baidumobads</string>
```

```
</array>
```

▼ LSApplicationQueriesSchemes	▲	Array	(1 item)
Item 0		String	baidumobads

2. ENABLE_BITCODE=YES. SDK已支持BITCODE,若不需要则直接设置为NO即可.



3. 在plist中,在NSAppTransportSecurity下设置NSAllowsArbitraryLoads为true. 注:

在Xcode7.1请设置App Transport Security Settings下设置 Allow Arbitrary Loads

为true

<key>NSAppTransportSecurity</key>

<dict>

<key>NSAllowsArbitraryLoads</key>

<true/>

</dict>

▼ App Transport Security Settings	▲	Dictionary	(1 item)
Allow Arbitrary Loads	▲	Boolean	YES

(下图为 Xcode7.1设置, 上图为7.0及以前的设置)

▼ Information Property List	Dictionary	(15 items)
▼ NSAppTransportSecurity	Dictionary	(1 item)
NSAllowsArbitraryLoads	Boolean	YES

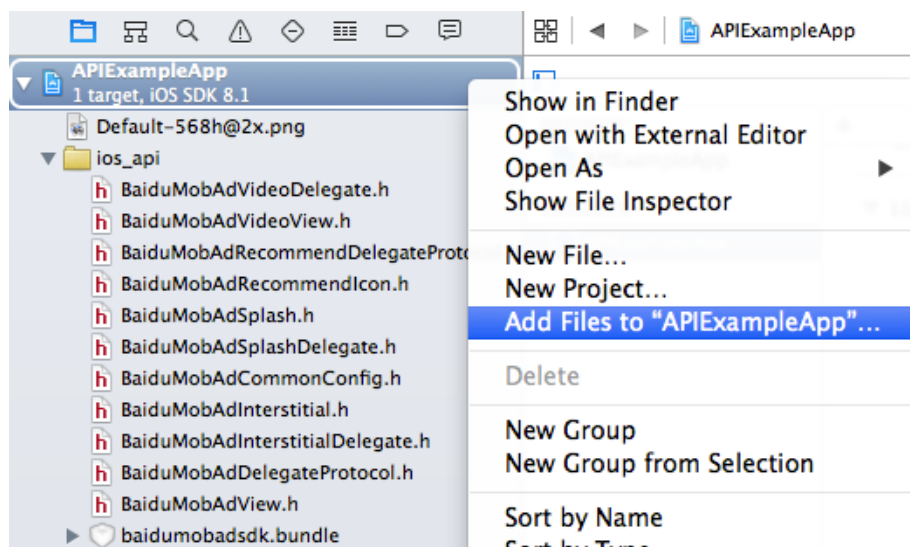
- SDK用到的.dylib 库在Xcode7都变成了.tbd后缀, 导入相应名字的库即可.

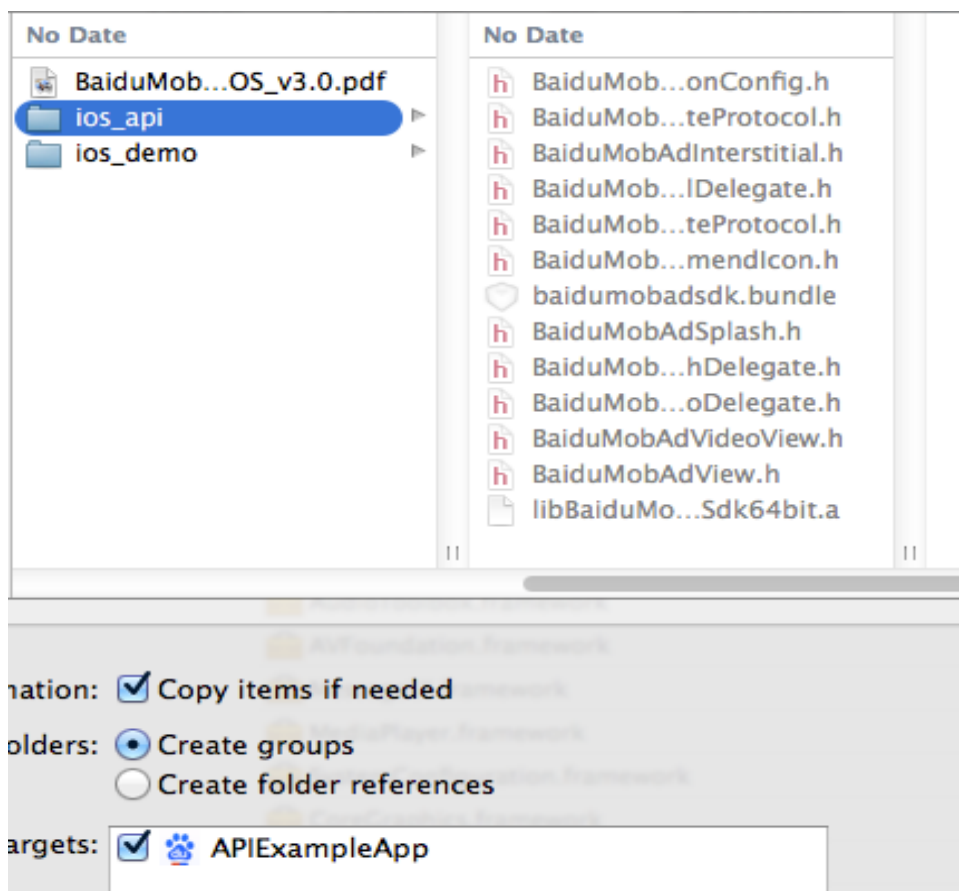
编译前的准备工作:

◆ 添加 SDK

需要添加的内容包括: SDK 静态库 (头文件)、Frameworks、编译参数设置.

- SDK 静态库. 建议直接把 ios_api 文件夹拖入工程中.





2. 需要添加的 Frameworks

CoreMedia.framework
 QuartzCore.framework
 Security.framework
 AdSupport.framework
 StoreKit.framework
 libz.dylib
 AudioToolbox.framework
 CoreMotion.framework
 MediaPlayer.framework
 AVFoundation.framework
 MessageUI.framework
 CoreLocation.framework
 CoreTelephony.framework
 SystemConfiguration.framework




















UIKit.framework

Foundation.framework

CoreGraphics.framework

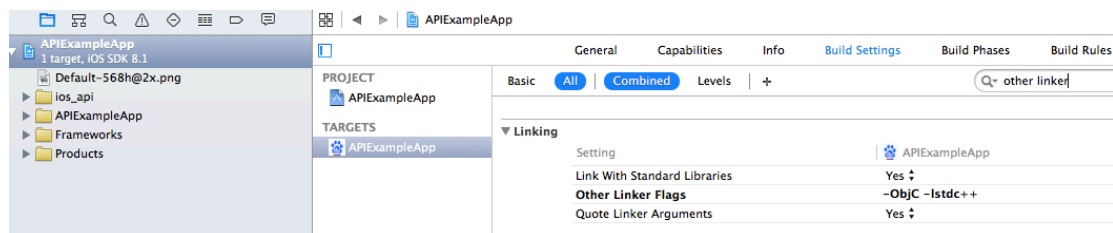
WebKit.framework

▼ Linked Frameworks and Libraries

Name	Status
 CoreMedia.framework	Required ⇅
 WebKit.framework	Required ⇅
 libBaiduMobAdSdk64bit.a	Required ⇅
 Security.framework	Required ⇅
 AdSupport.framework	Required ⇅
 CoreTelephony.framework	Required ⇅
 QuartzCore.framework	Required ⇅
 StoreKit.framework	Required ⇅
 libz.dylib	Required ⇅
 CoreMotion.framework	Required ⇅
 AudioToolbox.framework	Required ⇅
 AVFoundation.framework	Required ⇅
 MessageUI.framework	Required ⇅
 MediaPlayer.framework	Required ⇅
 SystemConfiguration.framework	Required ⇅
 CoreGraphics.framework	Required ⇅
 Foundation.framework	Required ⇅
 CoreLocation.framework	Required ⇅
 UIKit.framework	Required ⇅

3. 编译参数设置

targets->BuildSettings->Other Linker Flags ->添加 -ObjC -lstdc++



1. Banner 广告的开发示例

1. 初始广告视图:

1. 初始化广告视图实例
2. 设置 frame 和 delegate, 广告类型等属性。
3. 启动广告 (start)

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    //使用嵌入广告的方法实例。
    sharedAdView = [[BaiduMobAdView alloc] init];
    //把在mssp.baidu.com上创建后获得的代码位id写到这里
    sharedAdView.AdUnitTag = @"2015347";
    sharedAdView.AdType = BaiduMobAdViewTypeBanner;
    sharedAdView.frame = kAdViewPortraitRect;
    sharedAdView.delegate = self;
    [self.view addSubview:sharedAdView];
    [sharedAdView start];
}
```

2. 设置信息

请从百度 MobileSSP 网站获取应用 ID (APP ID); 设置函数 publisherId 返回值为 APPID.(这里是测试 id,请上线前换为申请到的 APPID,下同)

```
- (NSString *)publisherId
{
    return @"ccb60059"; //@"your_own_app_id";
}
```

3.可选 delegate 接口

```

-(void) willDisplayAd:(BaiduMobAdView*) advview
{
    NSLog(@"delegate: will display ad");
}

```

其他相关回调函数请参见 BaiduMobAdDelegateProtocol.h

获取地理位置信息的可选接口 enableLocation. 对于 iOS8 及以后如果获取地理位置时需要进行适配。

可以根据自己需要进行权限的控制，使用 requestWhenInUseAuthorization 或者 requestAlwaysAuthorization。获取方式可参照 demo 里的 plist 文件，相关部分截图如下：

Custom iOS Target Properties			
Key	Type	Value	
NSLocationWhenInUseUsageDescript...	Boolean	YES	

```

-(BOOL) enableLocation
{
    //启用location会有一次alert提示,请根据系统进行相关配置
    return NO;
}

```

2. 插屏广告的开发示例

◆ 插屏广告的使用

1. 预加载插屏广告（可选择在 viewDidLoad 等方法中启动插屏广告的加载）：

- 1、初始化插屏广告实例
- 2、设置 delegate，广告类型等属性。
- 3、预加载广告（load）
- 4、展示广告（show）

```
- (IBAction)clickLoadAd:(id)sender {
    self.interstitialView = [[[BaiduMobAdInterstitial alloc] init] autorelease];
    self.interstitialView.delegate = self;
    //把在mssp.baidu.com上创建后获得的代码位id写到这里
    self.interstitialView.AdUnitTag = @"2058554"
    self.interstitialView.interstitialType = BaiduMobAdViewTypeInterstitialOther;
    // 加载全屏插屏。每次仅加载一个广告物料,若需多次使用请在下次展示前重新执行load方法
    [self.interstitialView load];
}
```

插屏广告有以下几种应用场景：

- 1-普通场景中使用 BaiduMobAdViewTypeInterstitialOther
- 2-视频图片前贴片 BaiduMobAdViewTypeInterstitialBeforeVideo
- 3-视频暂停贴片 BaiduMobAdViewTypeInterstitialPauseVideo

请根据您的实际场景设置插屏广告实例的 interstitialType 属性。

说明：其中 1 默认请求全屏尺寸插屏广告,可在 mssp 中设置返回的物料类型，新添加插屏返回视频物料。插屏返回物料为视频的使用方法与普通全屏插屏一致，开发者判断插屏 isReady 后调用 presentFromRootViewController 加载视频。另外，需要请求插屏视频的需要在 MSSP 平台上配置广告位物料。

2 和 3 为插屏图片广告, 2 有 5s 倒计时自动关闭, 2 和 3 可手动设置广告大小, SDK 会默认返回请求尺寸或接近请求尺寸比例的广告, 未铺满部分由灰色填充。

2. 设置信息

请从百度 MobileSSP 网站获取应用 ID(APP ID), 设置函数 publisherId 返回值为 APPID。

```
- (NSString *)publisherId
{
    return @"ccb60059"; //@"your_own_app_id";
}
```

3. 展示插屏广告 (譬如: 用户点击了游戏的暂停按钮等)

```
-(IBAction)clickShowAd:(id)sender{
    //请注意:这里要确定isReady之后才可以调用presentFromRootViewController:方法,否则不会有显示.可以根据interstitialSuccessToLoadAd:方法确定.
    if (self.interstitialView.isReady){
        [self.interstitialView presentFromRootViewController:self];
    }
}
```

4. 可选 delegate 接口

在插屏广告的生命周期内, 会通过 delegate 通知回调相关事件: 插屏广告加载是否成功、展示是否成功、展示开始、展示结束等信息。如需要, 您可以利用这些信息进行处理。具体方法请参见 BaiduMobAdInterstitialDelegate.h。地理位置获取的 delegate 接口设置同 banner。

◆ 视频前贴片插屏广告的使用

视频前贴片是一种新的插屏样式, 出现在视频播放之前, 有 5s 倒计时, 可以自定义大小。

使用方法和插屏类似，区别在于需要指定插屏尺寸和插屏类型

1. 预加载插屏广告 (可选择在 viewDidLoad 等方法中启动插屏广告的加载) 可以传入广告的大小 (注意，必须大于 100 * 150 或 150*100)

```
- (IBAction)loadInterAd:(id)sender {
    self.adInterstitial = [[BaiduMobAdInterstitial alloc] init];
    self.adInterstitial.delegate = self;
    //把在mssp.baidu.com上创建后获得的代码位id写到这里
    self.adInterstitial.AdUnitTag = @"2058554";
    // 视频前贴片5s，倒计时
    self.adInterstitial.interstitialType = BaiduMobAdViewTypeInterstitialBeforeVideo;
    //加载自定义大小的插屏,这里的宽高可自定义.
    [self.adInterstitial loadUsingSize:CGRectMake(0, 0, 300, 300)];
}
```

2. 设置信息

请从百度 MobileSSP 网站获取应用 ID(APP ID) 设置函数 publisherId 返回值为 APPID。

```
- (NSString *)publisherId
{
    return @"ccb60059"; //@"your_own_app_id";
}
```

3. 加载好的时候展示前贴片广告.也可以参照普通插屏的方法在需要的时候展示.

```
- (void)interstitialSuccessToLoadAd:(BaiduMobAdInterstitial *)interstitial
{
    if (!self.customInterView) {
        UIView *customInterView = [[[UIView alloc] initWithFrame:CGRectMake(0, 100,
300, 300)]autorelease];
        self.customInterView = customInterView;
    }
    [self.view addSubview:self.customInterView];
    //展示自定义大小的插屏
    [interstitial presentFromView:self.customInterView];
}
```

4. 在展示结束时移除视图

```
- (void)interstitialDidDismissScreen:(BaiduMobAdInterstitial *)interstitial
{
    [self.customInterView removeFromSuperview];
}
```

◆ 视频暂停插屏广告的使用

视频暂停贴片是一种新的插屏样式，出现在视频播放暂停时，可以自定义大小，没有倒计时功能。使用方法和视频前贴片插屏类似，区别在于 load 时候设定插屏类型为 BaiduMobAdViewTypeInterstitialPauseVideo

```
- (IBAction)loadInterAd:(id)sender {
    self.adInterstitial = [[BaiduMobAdInterstitial alloc] init];
    self.adInterstitial.delegate = self;
    //把在mssp.baidu.com上创建后获得的代码位id写到这里
    self.adInterstitial.AdUnitTag = @"2058554";
    // 视频暂停贴片
    self.adInterstitial.interstitialType = BaiduMobAdViewTypeInterstitialPauseVideo;
    //加载自定义大小的插屏,这里的宽高可以自定义.
    [self.adInterstitial loadUsingSize:CGRectMake(0, 0, 300, 300)];
}
```

3. 开屏广告的开发示例

◆ 开屏广告的使用

1. 展示开屏广告

在- (BOOL)application:(UIApplication *)application

didFinishLaunchingWithOptions:(NSDictionary *)launchOptions 中

- 1、初始化开屏广告实例
- 2、设置 delegate，广告类型等属性。
- 3、加载广告（load）

开屏属性中 canSplashClick 为开屏是否可以点击属性，默认为可以点击。(投不可点击的广告主较少,建议使用默认设置即可.)

```
BaiduMobAdSplash *splash = [[BaiduMobAdSplash alloc] init];
splash.delegate = self;
//把在mssp.baidu.com上创建后获得的代码位id写到这里
splash.AdUnitTag = @"2058492";
splash.canSplashClick = YES;
[splash loadAndDisplayUsingKeyWindow:self.window];
self.splash = splash;
```

Splash 对象要写在.h 文件中, 避免提前释放.

2. 设置信息

请从百度 MobileSSP 网站获取应用 ID(APP ID),设置函数 publisherId 返回值为 APPID。

```
- (NSString *)publisherId
{
    return @"ccb60059"; //your_own_app_id
}
```

3. 可选 delegate 接口

在开屏广告的生命周期内，会通过 delegate 通知回调相关事件：开屏广告加载是否成功、展示是否成功、展示开始、展示结束等信息。如需要，您可以利用这些信息进行处理，建议在开屏广告展示成功结束，或者广告展示失败的回调进行后续处理。详细使用方法请参见 `BaiduMobAdSplashDelegate.h`

地理位置获取的 delegate 接口设置同 banner。

◆ 半屏开屏广告的使用

为了方便开屏广告嵌入 app 的 icon 和自定义背景，引入半屏开屏广告

1. 展示开屏广告（在 `-(BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions` 中）：

- 1、初始化自定义开屏 `customSplashView` 并添加到 window
- 2、初始化开屏广告容器 `baiduSplashContainer` 并添加到 `customSplashView`
- 3、初始化 `BaiduMobAdSplash`
- 4、设置 delegate 和开屏属性
- 5、在 `baiduSplashContainer` 内加载百度广告
(`loadAndDisplayUsingContainerView`)

开屏属性中 `canSplashClick` 为开屏是否可以点击属性，默认为可以点击。


```
BaiduMobAdSplash *splash = [[[BaiduMobAdSplash alloc] init]autorelease];
splash.delegate = self;
//把在mssp.baidu.com上创建后获得的代码位id写到这里
splash.AdUnitTag = @"2058492";
splash.canSplashClick = YES;
self.splash = splash;
//可以在customSplashView上显示包含其他内容的自定义开屏
self.customSplashView = [[[UIView alloc] initWithFrame:self.window.frame]autorelease];
self.customSplashView.backgroundColor = [UIColor whiteColor];
[self.window addSubview:self.customSplashView];

//在baiduSplashContainer用做上展现百度广告的容器，注意尺寸必须大于200*200，
并且baiduSplashContainer需要全部在window内，同时开机画面不建议旋转
UIView * baiduSplashContainer = [[[UIView alloc] initWithFrame:CGRectMake(0, 0, 320,
420)]autorelease];
[self.customSplashView addSubview:baiduSplashContainer];

//在的baiduSplashContainer里展现百度广告
[splash loadAndDisplayUsingContainerView:baiduSplashContainer];
```

注意：半屏幕开屏尺寸必须大于 200*200，并且落在 window 内部

2.设置信息

```
-(NSString *)publisherId
{
    return @"ccb60059"; //your_own_app_id
}
```

请从百度 MobileSSP 网站获取应用 ID(APP ID) ,设置函数 publisherId 返回值为 APPID。

3.可选 delegate 接口

在开屏广告的生命周期内，会通过 delegate 通知回调相关事件：开屏广告加载是否成功、展示是否成功、展示开始、展示结束等信息。如需要，您可以利用这些信息进行处理，建议在开屏广告展示成功结束，或者广告展示失败的回调进行后续处理。对于半屏开屏，建议在广告展示失败和结束时移除自定义的 customSplashView

```
- (void)splashSuccessPresentScreen:(BaiduMobAdSplash *)splash{
    NSLog(@"splashSuccessPresentScreen");
}
- (void)splashIfailPresentScreen:(BaiduMobAdSplash *)splash withError:(BaiduMobFailReason)
reason{
    //自定义开屏移除
    NSLog(@"splashIfailPresentScreen withError:%d",reason);
    [self.customSplashView removeFromSuperview];
}

/**
 * 广告展示结束
 */
- (void)splashDidDismissScreen:(BaiduMobAdSplash *)splash {
    NSLog(@"splashDidDismissScreen");
    //自定义开屏移除
    [self.customSplashView removeFromSuperview];
}
```

4. 推广墙广告的开发示例

创建推广墙对象需传入入口图片，用户点击该图片后，进入推广墙页面。

```
UIImageView * image = [[[UIImageView alloc ]initWithImage:[self
imagesNamedFromCustomBundle:@"adwall_white_pic" ]]autorelease];
BaiduMobAdRecommendIcon *adRecommendView = [[[BaiduMobAdRecommendIcon
alloc]initWithFrame:CGRectMake(20, 80, image.frame.size.width, image.frame.size.height)
imageView:image delegate:self]autorelease];
[self.view addSubview:adRecommendView];
```

不需要对 imageView 进行除了位置和 image 以外的设置。

Delegate 需要实现 BaiduMobAdRecommendDelegate 的方法，应用在 mssp.baidu.com 上的 id，即 publisherId。以及代码位 id，即 apId。

```
- (NSString *)publisherId
{
    return @"ccb60059"; //@"your_own_app_id";
}

- (NSString *)apId
{
    return @"2078476";
}
```

两个可选的回调方法

* 推荐页面关闭

```
- (void)recommendPageDidClosed;
```

* 推荐图标点击回调

```
- (void)recommendViewDidClick;
```

5. 视频广告的开发示例

视频广告是一种新的广告样式，主要使用场景是在视频播放之前播放视频广告。

1. 创建视频广告 BaiduMobAdVideoView，请求视频广告 requestVideoAd，可选视频广告长度 duration 为 BaiduMobAdVideoDurationShort, BaiduMobAdVideoDurationMiddle, BaiduMobAdVideoDurationLong, 分别为 15 秒，30 秒，45 秒，推荐广告 frame 比例为 4:3 或 16:9，showTimeLabel 表示是否显示倒计时 label，AdUnitTag 设置代码位 id。在 didReadyForDisplay 回调函数里 startVideoAd 播放视频广告。

```
self.videoAd = [[[BaiduMobAdVideoView alloc] init] autorelease];
self.videoAd.delegate = self;
self.videoAd.AdUnitTag = @"2058633"; //设置代码位 id
self.videoAd.frame = CGRectMake(0, 64, 320, 240);
self.videoAd.duration = BaiduMobAdVideoDurationShort;//15s
self.videoAd.showTimeLabel = YES;
[self.view addSubview:self.videoAd];
[self.videoAd requestVideoAd]; //请求视频广告
```

```
- (void)didReadyForDisplay:(BaiduMobAdVideoView *)video{
    [self.videoAd startVideoAd]; //播放视频广告
}
```

2. 设置信息

请获取应用 ID (APP ID); 设置函数 publisherId 返回值为 APPID。

```
- (NSString *)publisherId
{
    return @"ccb60059"; //your_own_app_id
}
```

3.delegate 方法，广告展示结束，移除视频 view

```
/**
 * 广告展示成功
 */
- (void)videoSuccessPresentScreen:(BaiduMobAdVideoView *)video;

/**
 * 广告展示失败
 */
- (void)videoFailPresentScreen:(BaiduMobAdVideoView *)video
    withError:(BaiduMobFailReason) reason;

/**
 * 广告展示结束，关闭视频，移除视频 view
 */
- (void)videoDidDismissScreen:(BaiduMobAdVideoView *)video{
    [self.videoAd closeVideoAd];
    [self.videoAd removeFromSuperview];
}

/**
 * 广告点击
 */
- (void)videoClicked:(BaiduMobAdVideoView *)video;
```

6. 可能遇到的问题

1. Q:在真机上请求不到广告,在 safari 调试模式下电脑可以请求到

A:有可能是 app 本身没有设置位置请求(CLLocation 和 info.plist)的情况下,在 location == YES 时会不显示广告.可设置 location == NO 或者写相关代码请求位置.

2. Q:点击广告出现详情页面后断网然后关闭页面造成 crash

A:在断网时如果需要移除广告可 removeFromSuperview 或者直接置空,不需要同时设置.

3. Q:界面退出的时候要不要对广告进行设置

A:可参照 demo. 建议设置代理为空和 adView = nil.

4. 错误提示: 控制台会输出如下信息, 可以根据信息排查对应错误.

百度广告 SDK ErrorCode: [%ld], ErrorDesc: [%@]

详见: Baidu_Mobads_SDK_错误码对照表.pdf

5. 常见技术类问题与解答:

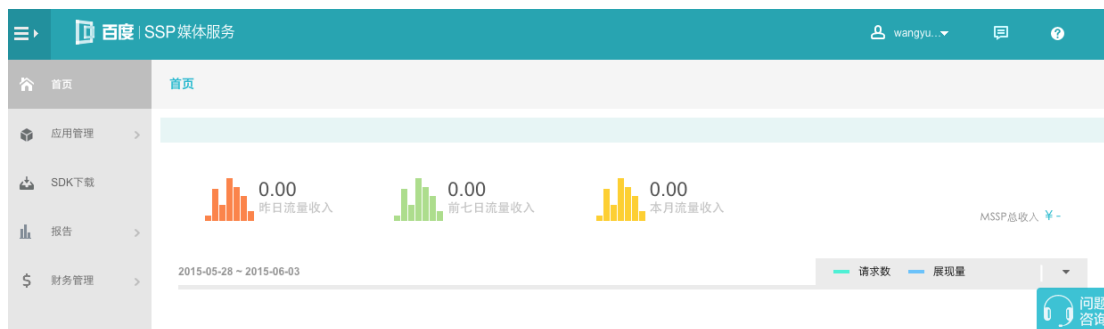
<http://yingxiao.baidu.com/zhichi/knowledge/detail.action?channelId=25&classId=14542&knowledgeId=14739>

六、 联系我们

感谢您的阅读，如果有问题请联系我们。

百度移动 SSP 官方客服 qq 群: 243927553, 请注明会员名.

也可以通过网页给我们留言咨询,我们会尽快回复.



附件 1 申请 APP ID

1. 作用

APP ID (应用 ID) 用于标识您的应用。

2. 申请流程

1. 百度 MobileSSP [注册](#) 并 登录 。

2. 创建应用

登录后，点击“应用管理”→“新增移动应用”，如下图所示。



填写您的应用信息并提供有效的下载地址后，点击“提交”，完成应用的创建。

添加成功后，会返回上述页面，此时应用的状态为“审核中”，请耐心等待审核，审核周期为 1-2 个工作日。

3. 在应用列表中，您可以获取对应的 APP ID，请在下面的代码位管理中添加需要的代码位,以获取代码位 id,如图所示.

百度 MobileSSP 媒体 首页 > 代码位管理 > 新建代码位 ②

代码位名称: * test 少于40个字符

所属应用: 测试应用 (iOS) 指定代码位所属应用

所属系统: ios iphone 应用版本: 1.0

代码位类型: 横幅 ① 积分墙和信息流仅适用于百度托管业务，不适用于自主变现业务

备注信息:

完成 取消

4. 注意：只有当您的应用状态变为“审核通过”后，才能看到有效的 APP ID，此时部署完代码后才会有正常的广告展现并开始计费。

在效果报告页面中，您可以查看您的收益以及每个应用的详细数据。