

百度 MobileSSP

移动应用推广 SDK

用户手册 (iOS 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	2
二. 阅读对象	2
三. 版本支持	2
四. 工程设置	2
1. 创建应用	2
2. 嵌入 SDK	3
3. 提交 Store	3
4. 更新版本	3
5. 应用 id 与代码位 id	3
五. 开发示例	4
◆ 添加 SDK	5
1. 横幅广告	7
2. 视频贴片广告	9
3. 信息流广告	10
4. 开屏广告	19
◆ 全屏开屏广告的使用	19
◆ 半屏开屏广告的使用	20
5. 插屏广告	22
◆ 插屏广告的使用	22
◆ 视频前贴片插屏广告的使用	24
◆ 视频暂停插屏广告的使用	25
6. Feeds 联盟	25
7. 可能遇到的问题	26
六. 联系我们	26

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 6.0+ 支持 IPv6

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。

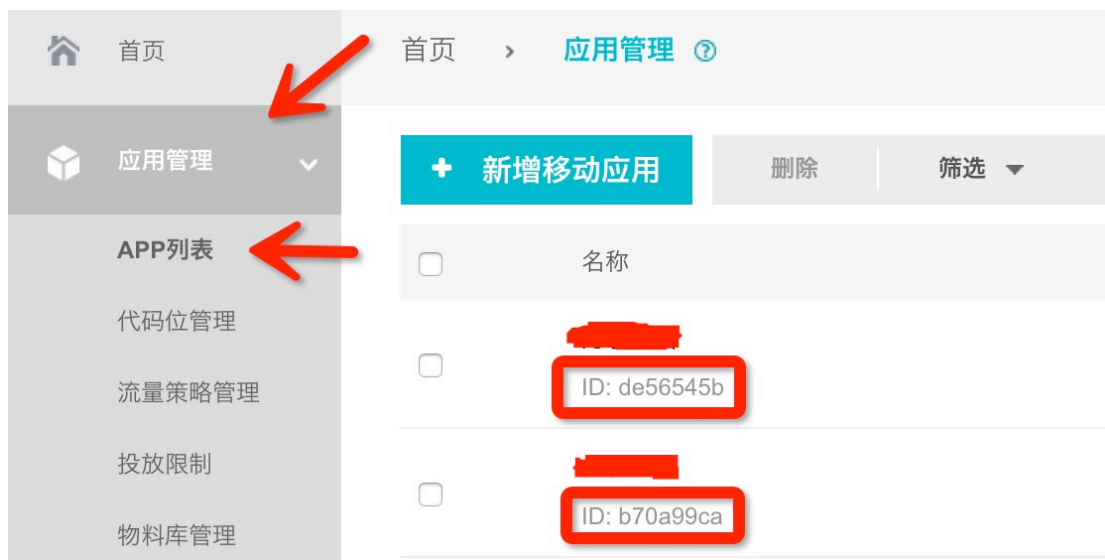
3. 提交 Store

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.

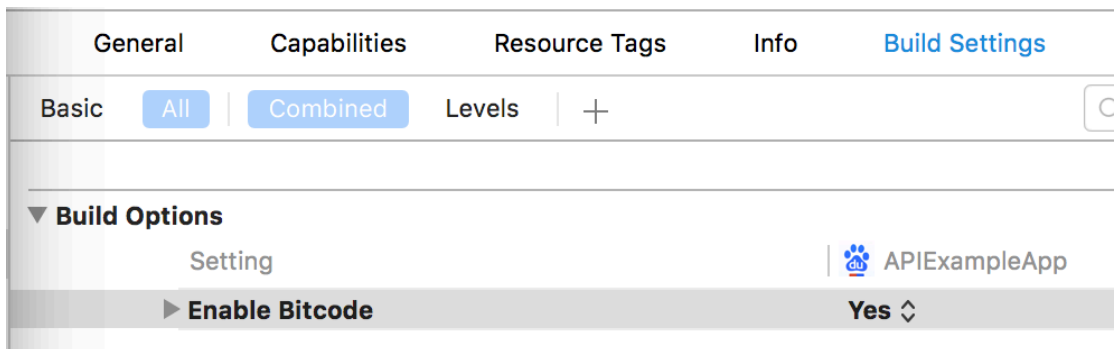


五. 开发示例

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



2. 在 plist 中, 在 NSAppTransportSecurity 下设置 NSAllowsArbitraryLoads 为 true.
注: 在 Xcode 7.1 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

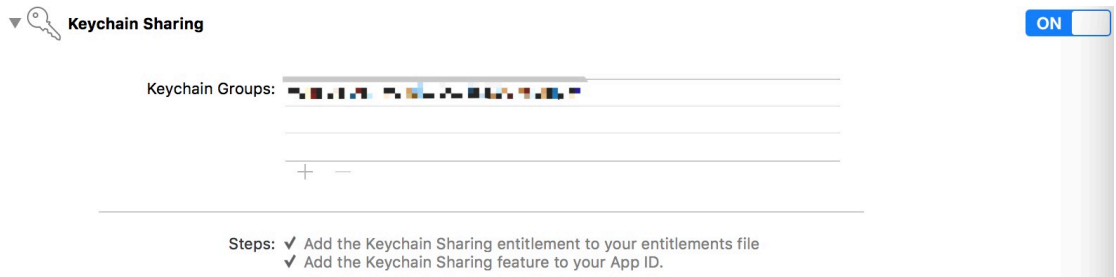
(下图为 Xcode 7.1 设置, 上图为 7.0 及以前的设置)

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

3. ATS 默认开启状态, 可根据需要关闭 App Transport Security Settings, 设置关闭 BaiduMobAdSetting 的 supportHttps, 以请求 http 广告, 多个产品只需要设置一次.

[BaiduMobAdSetting sharedInstance].supportHttps = NO;

4. 设置 Capabilities 里打开 KeyChain Sharing, 同时在开发者中心进行相应设置, 以推送更精准广告

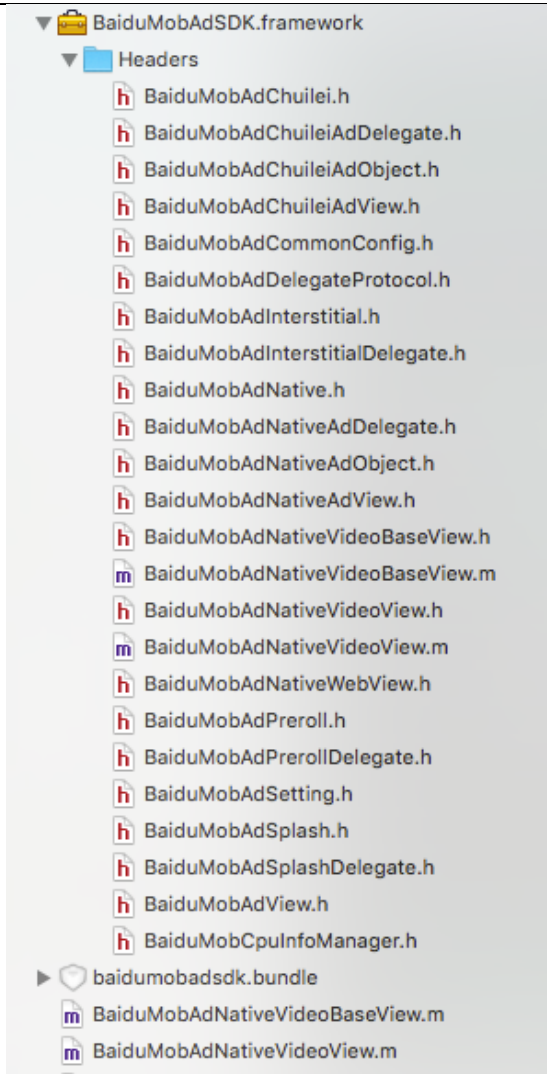


编译前的准备工作:

◆ 添加 SDK

需要添加的内容包括: SDK、Frameworks.

1. SDK framework.



注意：信息流视频需要从 BaiduMobAdSDK.framework 引入 BaiduMobAdNativeVideoBaseView.m 和 BaiduMobAdNativeVideoView.m 到 project 目录中












2. 需要添加的 Frameworks (以 Demo 为准)

AdSupport.framework
 StoreKit.framework
 Libc++.dylib
 CoreMotion.framework
 MessageUI.framework
 CoreLocation.framework
 CoreTelephony.framework
 SystemConfiguration.framework
 SafariServices.framework

CoreMedia.framework

AVFoundation.framework

▼ Linked Frameworks and Libraries

Name	Status
 SafariServices.framework	Required ⚙
 MessageUI.framework	Required ⚙
 CoreMedia.framework	Required ⚙
 BaiduMobAdSDK.framework	Required ⚙
 libc++.tbd	Required ⚙
 CoreMotion.framework	Required ⚙
 SystemConfiguration.framework	Required ⚙
 CoreLocation.framework	Required ⚙
 CoreTelephony.framework	Required ⚙
 AdSupport.framework	Required ⚙
 AVFoundation.framework	Required ⚙
+ -	

1. 横幅广告

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)

按照宽高比的不同, 横幅广告 Banner 广告分为 4 种:20:3, 3:2, 7:3, 2:1

这里以嵌入宽高比为 20:3 的 banner 为例进行说明, 其它 3 种比例横幅的嵌入方式类似。

注意:

1 嵌入 Banner 广告的时候, Banner view 的宽高比例需要保持与 mssp 网站设置该 Banner 广告位的比例一致。创建的时候非 20:3 的广告位建议选择模板, 如果宽和高都不指明具体大小, 则该 banner 的宽高比默认设为 20:3, 宽的大小为屏幕的宽度。

2 Banner 的宽最小为 200, 高最小为 30。即如果 Banner 的宽如果小于 200, 则强制设为 200, 如果 Banner 的高小于 30, 则强制设为 30。

(以 iPhone6 为例)

2.1 对于 20:3 的比例，建议宽高为 375*56

2.2 对于 3:2 的比例，建议宽高为 375*250;

2.3 对于 7:3 的比例，建议宽高为 375*160;

2.4 对于 2:1 的比例，建议宽高为 375*187;

```
#import "BaiduMobAdSDK/BaiduMobAdView.h"
#import "BaiduMobAdSDK/BaiduMobAdDelegateProtocol.h"
- (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];
}
```

3.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) 注意，iOS 和 android 的 app 请使用对应不同的 app ID，如果 appid 的操作系统错误，将无法返回广告

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

4.可选 delegate 接口 (根据新广告法, 横幅添加了关闭按钮, 并有相关回调)

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

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


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:

Key	Type	Value
▼ Information Property List	Dictionary	(17 items)
NSLocationWhenInUseUsageDescription	String	

```

-(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. 视频贴片广告

视频贴片广告，可在视频正片播放之前使用。

1.注意，iOS 和 android 的 app 请使用对应不同的 app ID，如果 appid 的操作系统错误，将无法返回广告

```

#import "BaiduMobAdSDK/BaiduMobAdPrerollDelegate.h"
#import "BaiduMobAdSDK/BaiduMobAdPreroll.h"

- (void)loadAd {
    UIView *baseview = [[UIView alloc] initWithFrame:CGRectMake(10, 64, 320, 240)];
    [self.view addSubview:baseview];
    self.prerollAd = [[[BaiduMobAdPreroll alloc] init] autorelease];
    self.prerollAd.publisherId = @"ccb60059";//设置 iOS app id
    self.prerollAd.adId = @"2015347";//设置视频贴片代码位 id
    self.prerollAd.renderBaseView = baseview;//设置贴片的 baseview
    self.prerollAd.delegate = self;//设置 BaiduMobAdPrerollDelegate
    [self.prerollAd request];//请求贴片广告
}

```

2. 可选 delegate 接口

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

```
//贴片加载完成
- (void)didAdReady:(BaiduMobAdPreroll *)preroll {
    NSLog(@"didAdReady");
}
//贴片加载失败
- (void)didAdFailed:(BaiduMobAdPreroll *)preroll withError:(BaiduMobFailReason) reason{
    NSLog(@"didAdFailed");
    self.button.userInteractionEnabled = YES;
}
//贴片开始播放
- (void)didAdStart:(BaiduMobAdPreroll *)preroll {
    NSLog(@"didAdStart");
}
//贴片结束播放
- (void)didAdFinish:(BaiduMobAdPreroll *)preroll {
    NSLog(@"didAdFinish");
    self.button.userInteractionEnabled = YES;
}
//广告点击
- (void)didAdClicked:(BaiduMobAdPreroll *)preroll {
    NSLog(@"didAdClicked");
}
```

3. 信息流广告

1.导入头文件,设置代理, BaiduMobAdNativeVideoView 为百度提供的播放器

```
#import " BaiduMobAdSDK/ BaiduMobAdNative.h"
#import " BaiduMobAdSDK/BaiduMobAdNativeAdObject.h"
#import " BaiduMobAdSDK/BaiduMobAdNativeAdView.h"
#import " BaiduMobAdSDK/BaiduMobAdNativeVideoView.h"
#import " BaiduMobAdSDK/BaiduMobAdNativeWebView.h"
@interface ViewController : UIViewController<BaiduMobAdNativeAdDelegate>
@property(nonatomic,retain)BaiduMobAdNative *native;
```

开发者可在 MSSP 上配置广告位物料类型, 如模板, 视频, 元素等
设置 landingpage 的 navigationbar 背景颜色

`[BaiduMobAdSetting setLpStyle:BaiduMobAdLpStyleDefault];`

2.设置是否自己处理点击事件,不设置或者设置为 false 时, 如果开发者自己不添加点击事件, 广告 View 必须为 BaiduMobAdNativeAdView

```
//设置开发者自己处理 View 点击事件,只需要设置一次,不设置时, sdk 内部采用默认方式处理  
[BaiduMobAdNativeAdView dealTapGesture:true];
```

3. 请求广告, 用 BaiduMobAdNative 请求广告.

```
-(IBAction)pressToLoadAd:(id)sender  
{  
    if(!self.native)  
    {  
        self.native = [[BaiduMobAdNative alloc] init];  
        self.native.delegate = self;  
    }  
    //请求原生广告  
    [self.native requestNativeAds];  
}
```

4. 设置在 mssp 上的应用 id 以及广告位 id

```
//创建后获得的应用 id,上线前请换为自己的 id  
-(NSString*)publisherId  
{  
    return @"ccb60059";  
}  
  
//创建后获得的信息流广告位 id,上线前请换为自己的 id  
-(NSString*)apId  
{  
    return @"2058621";  
}
```

5. nativeAdsSuccessLoad 回调函数表明请求广告成功,

(NSArray*)nativeAdViewArray 参数标示成功返回的广告数组。

BaiduMobAdNativeAdObject.h 中可获取的属性见下表,可通过以下属性返回广告字段。

```
// 标题 text
@property (copy, nonatomic) NSString *title;
//描述 text
@property (copy, nonatomic) NSString *text;
//小图 url
@property (copy, nonatomic) NSString *iconImageURLString;
//大图 url
@property (copy, nonatomic) NSString *mainImageURLString;
// 广告标识图标 url
@property (copy, nonatomic) NSString *adLogoURLString;
// 百度 logo 图标 url
@property (copy, nonatomic) NSString *baiduLogoURLString;
//多图信息流的 image url array
@property (strong, nonatomic) NSArray *morepics;
//视频 url
@property (copy, nonatomic) NSString *videoURLString;
//视频时长, 单位为 s
@property (copy, nonatomic) NSNumber *videoDuration;
//品牌名称, 若广告返回中无品牌名称则为空
@property (copy, nonatomic) NSString *brandName;
//在 MSSP 上配置广告位物料类型后, 对返回的广告单元, 需判断 MaterialType 再决定使用何种渲染组件,VIDEO 需使用视频播放组件, HTML 需使用 web 组件
@property MaterialType materialType;
//返回广告单元的点击类型, 可用于 video 设置引导按钮
@property (nonatomic) BaiduMobNativeAdActionType actType;
//是否过期, 默认为 false, 30 分钟后过期, 需要重新请求广告
-(BOOL) isExpired;
//重要, 使用视频播放组件, 一定要调用这个方法发送视频状态事件和当前视频播放位置, 单位为 s
-(void)trackVideoEvent:(BaiduAdNativeVideoEvent)event withCurrentTime:(NSTimeInterval)currentPlaybackTime;
```

6. 建立广告视图, 初始化时传广告 icon 图, 广告大图, 广告品牌名称, 广告标题, 广告描述视图和视频 view, 如果不需要的 view/label 传 nil 即可. 按照 MaterialType 选择是否创建视频 view。

```

UILabel *brandLabel = [[UILabel alloc] initWithFrame:CGRectMake(85, 15, 212, 20)];
UILabel *titleLabel = [[UILabel alloc] initWithFrame:CGRectMake(85, 60, 200, 10)];
UILabel *textLabel = [[UILabel alloc] initWithFrame:CGRectMake(85, 40, 200, 10)];
UIImageView *iconImageView = [[UIImageView alloc] initWithFrame:CGRectMake(10, 10, 60, 60)];
    UIImageView *mainImageView = [[UIImageView alloc] initWithFrame:CGRectMake(10, 80, 300,
250)];
    UIImageView *baiduLogoView = [[[UIImageView alloc] initWithFrame:CGRectMake(10, 310, 18,
18)]autorelease];
    UIImageView *adLogoView = [[[UIImageView alloc] initWithFrame:CGRectMake(286, 318, 24,
12)]autorelease];
    BaiduMobAdNativeVideoView *videoView = nil;
    if (object.materialType == VIDEO) {
        videoView = [[BaiduMobAdNativeVideoView alloc] initWithFrame:CGRectMake(0, 80, 310, 270)
andObject:object];
        videoView.isAutoPlay = NO; // 是否自动播放，默认为 yes
        //在大图上添加播放按钮引导用户点击视频
        UIImage *image = [self imageResoureForName:@"play_big_image"];
        UIImageView *playImage = [[UIImageView alloc ]initWithImage:image];
        playImage.frame = CGRectMake(86, 61, 128, 128);
        [mainImageView addSubview:playImage];
        //这里根据网站上的设置.如果没有设置这么多则相应字段设置为 nil 即可.
        nativeAdView = [[BaiduMobAdNativeAdView alloc] initWithFrame:frame brand-
Name:brandLabel
        title:titleLabel text:textLabel icon:iconImageView mainImage:mainImageView vide-
oView:videoView];
        nativeAdView.baiduLogoImageView = baiduLogoView;
        [nativeAdView addSubview:baiduLogoView];
        nativeAdView.adLogoImageView = adLogoView;
        [nativeAdView addSubview:adLogoView];
        //添加广告字样和 logo 到视图上. Normal 同样. Html 不需要.
    } else if (object.materialType == NORMAL) {
        nativeAdView = [[BaiduMobAdNativeAdView alloc] initWithFrame:frame
        brandName:brandLabel
        title:titleLabel
        text:textLabel
        icon:iconImageView
        mainImage:mainImageView];
    } else if (object.materialType == HTML) { //如果是 html 请使用 BaiduMobAdNativeWebView
加载
        BaiduMobAdNativeWebView *webview = [[BaiduMobAdNativeWebView al-
loc] initWithFrame:frame andObject:object];
        nativeAdView = [[BaiduMobAdNativeAdView alloc] initWithFrame:frame
        }
    }
}

```

-
7. loadAndDisplayNativeAdWithObject: completion:加载和显示广告视图。

展示前检查广告是否过期[object isExpired]

开发者确保显示视图后，调用 trackImpression 通知 sdk 广告视图已成功展示

广告展现时播放视频 BaiduMobAdNativeVideoView [videoView play]

```

//nativeAds 为成功返回的广告数组,个数可能少于请求的数量.
-(void)nativeAdObjectsSuccessLoad:(NSArray *)nativeAds
{
    NSLog(@"nativeAdObjectsSuccessLoad:%ld",nativeAds.count);
    [self.mainArray removeAllObjects];
    for (int i = 0; i<nativeAds.count; i++)
    {
        //取到每一条广告 object
        BaiduMobAdNativeAdObject*object = [nativeAds objectAtIndex:i];
        //创建每一条的广告 view
        BaiduMobAdNativeAdView*view = [self createAdViewWithFrame:CGRectMake(0, 0, WIDTH,
350) object:object];

        //自己添加点击事件时, 必须提前设置[BaiduMobAdNativeAdView dealTapGesture:true];
        UITapGestureRecognizer *tapGesture = [[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(tapGesture:)];
        tapGesture.view.tag = i;
        [view addGestureRecognizer:tapGesture];

        //展现前检查是否过期, 30 分钟广告过期, 请放弃展示并重新请求
        if ([object isExpired]) {
            continue;
        }
        //根据广告 object 加载和显示广告内容
        [view loadAndDisplayNativeAdWithObject:object
         completion:^(NSArray *errors) {
             if (!errors) {
                 [self.mainArray addObject:view];
                 // 确定视图显示在 window 上之后再调用 trackImpression, 不要太早调用,
                 //旧版本的调用方式: [view trackImpression], 新版本支持开发者自定义 UI, 不依//赖
                 BaiduMobAdNativeAdView, 展现的调用方式为[object trackImpression:view]
                 // 本 demo 中在 tableViewCell 展现时调用了 trackImpression
                 [object trackImpression:view];
             }
         }];
    }
}

//自定义点击事件时, 需要在点击事件里面调用
[object handleClick:view];

```

8. nativeAdFailLoad 是请求广告失败， nativeAdClicked 是广告点击的回调函数， didDismissLandingPage 是广告详情页被关闭，对于视频广告， nativeAdVideoAreaClick 表示点击视频预览图的回调

```
- (void)nativeAdsFailLoad:(BaiduMobFailReason)reason
{
    NSLog(@"nativeAdFailLoad===== %d", reason);
}

//广告被点击，打开后续详情页面，如果为视频广告，可选择暂停视频
- (void)nativeAdClicked:(BaiduMobAdNativeAdView *)nativeAdView
{
    if ([nativeAdView isKindOfClass:[BaiduMobAdNativeAdView class]]) {
        BaiduMobAdNativeAdView *bdAdView = (BaiduMobAdNativeAdView *) nativeAdView;
        if ([bdAdView.videoView isKindOfClass:[BaiduMobAdNativeVideoView class]]) {
            [(BaiduMobAdNativeVideoView *)bdAdView.videoView pause];
        }
    }
    NSLog(@"nativeAdClicked");
}

//广告详情页被关闭，如果为视频广告，可选择继续播放视频
- (void)didDismissLandingPage:(UIView *)nativeAdView
{
    if ([nativeAdView isKindOfClass:[BaiduMobAdNativeAdView class]]) {
        BaiduMobAdNativeAdView *bdAdView = (BaiduMobAdNativeAdView *) nativeAdView;
        if ([bdAdView.videoView isKindOfClass:[BaiduMobAdNativeVideoView class]]) {
            [(BaiduMobAdNativeVideoView *)bdAdView.videoView play];
        }
    }
    NSLog(@"didDismissLandingPage");
}
```


9. 可选 delegate 接口，详见 BaiduMobAdNativeAdDelegate.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:

Key	Type	Value
▼ Information Property List	Dictionary	(17 items)
NSLocationWhenInUseUsageDescription	String	

```

-(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;
}

```

关于 html 渲染样式，BaiduMobAdNativeAdDelegate 提供模板宽高设置

```

//模板高度，仅用于信息流模版广告
-(NSNumber*)height {
    return [NSNumber numberWithInt:180];
}
// 模板宽度，仅用于信息流模版广告
-(NSNumber*)width {
    return [NSNumber numberWithInt:320];
}

```

10. 关于视频播放，我们提供 BaiduMobAdNativeView 提供了渲染视频的能力

具体实现参照 DEMO 中的 NativeVideoTableViewController，该类实现了监听滚动状态实现自动播放逻辑，这里的是否 wifi 下自动播放可以在 Mssp 上配置。视频控件直接使用 BaiduMobAdNativeVideoView 即可

```

BaiduMobAdNativeAdView *nativeAdView;

    BaiduMobAdNativeVideoView *videoView = [[BaiduMobAdNativeVi-
deoView alloc] initWithFrame:CGRectMake(10, 70, kScreenWidth - 20, 180)
andObject:object];

    //在播放器内部渲染大图，大图 View 传参数 nil
    nativeAdView = [[BaiduMobAdNativeAdView alloc] initWithFrame:frame
                    brandName:brandLabel
                    title:titleLabel
                    text:textLabel
                    icon:iconImageView
                    mainImage:nil //如果调用
BaiduMobAdNativeVideoView 来展示视频广告，不需设置大图控件
                    videoView:videoView

    ];

    nativeAdView.baiduLogoImageView = baiduLogoView;
    [nativeAdView addSubview:baiduLogoView];
    nativeAdView.adLogoImageView = adLogoView;
    [nativeAdView addSubview:adLogoView];
    return nativeAdView;

```

其中滚动监听事件，在滚动停止后，媒体通知 sdk 滚动停止

```

/**
 *滑动停止后，通知当前可见区域内的 videoview
 */
- (void)handleScrollStop{
    NSLog(@"baidu:handleScrollStop.");
    NSArray *visiblePath = [self.tableView indexPathsForVisibleRows];
    for (NSIndexPath* visible in visiblePath) {
        if ([self.adViewArray count]> visible.row) {
            //自动播放逻辑
            BaiduMobAdNativeAdView *view = [self.adViewArray objec-
tAtIndex:visible.row];
            if ([view.videoView handleScrollStop]) {
                break;
            };
        }
    }
}

```

滚动停止后需要通知可见区域的广告发送展现

```
/**
 *滑动停止后，检查当前可见区域内的 cell 来发送展现
 */
- (void)handleScrollStopImpression{
    NSArray *visiblePath = [self.tableView indexPathsForVisibleRows];
    for (NSIndexPath* visible in visiblePath) {
        if ([self.adViewArray count]> visible.row) {
            BaiduMobAdNativeAdObject *object = [self.adsArray objectAtIndex:visible.row];
            BaiduMobAdNativeAdView *view = [self.adViewArray objectAtIndex:visible.row];
            // 确定视图显示在 window 上之后再调用 trackImpression，不要太早调用
            // 在 tableview 或 scrollview 中使用时尤其要注意
            [object trackImpression:view];
        }
    }
}
```

如果您希望自己实现，请继承 BaiduMobAdNativeVideoBaseView 类，并实现如下方法

```
//自定义视频播放器需要实现的接口
@interface BaiduMobAdNativeVideoBaseView : UIView

- (void)play;
- (void)pause;
- (void)stop;

#warning 重要，一定要向 BaiduMobAdNativeAdObject 发送视频状态事件和当前视频播放的位置，只有在第一次播放才需要发送
- (void)sendVideEvent:(BaiduAdNativeVideoEvent)event currentTime:(NSTimeInterval)currentTime;

@end
```

4. 开屏广告

开屏广告展现视频时可设置缓存阈值，单位为 M, 取值范围 15M-100M,默认 30M
 [BaiduMobAdSetting setMaxVideoCacheCapacityMb:30];

◆ 全屏开屏广告的使用

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.)

Splash 对象要写在.h 文件中, 避免提前释放.

```
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;
```

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、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)

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

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

注意：半屏幕开屏尺寸必须大于 200*200，并且落在 window 内部

```
#import "BaiduMobAdSDK/BaiduMobAdSplash.h"

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    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];}
```

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。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)splashFailPresentScreen:(BaiduMobAdSplash *)splash withError:(BaiduMobFailReason) reason{
    //自定义开屏移除
    NSLog(@"splashFailPresentScreen 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];
}
```

5. 插屏广告

◆ 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 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];
}

```

3. 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)];
}

```

6. Feeds 联盟

备注：媒体伙伴必须在 Mssp 业务端选择接入 Feeds 联盟的应用与频道类型，以便在接入内容页中生成广告，从而获得广告收益。不进行相关操作，将无法获得 Feeds 联盟页面的广告收益。

APP 通过 SDK 接口获取展示 Feeds 联盟页面的 URL，APP 在 UIWebView 中加载获取到的 URL 即可展示 Feeds 联盟页面。

通过调用 SDK 接口 getCpuInfoUrl 获取 URL

```
NSString *urlStr = [[BaiduMobCpuInfoManager shared] getCpuInfoUrlWithChannelId:@"1001"
appId:@" d77e414"];
```

注意：

- Feeds 联盟 url 获取后只能展示一次，多次展示需要每次通过 SDK 接口重新获取

请在 MSSP(<http://mssp.baidu.com/home/developer.html>)获取频道 id



接入应用	APP联盟状态	接入方式	接入url	合作样式	接入类目	创建时间
MSSP PM测试 ID: e866cfb0	已通过	SDK	https://cpu.baidu.com/1033/e866cfb0	栏目	视频频道 ID: 1033	2017-06-01 16:30
测试安卓for API ID: e7627f43	已通过	SDK	https://cpu.baidu.com/1003/e7627f43	栏目	图片频道 ID: 1003	2017-04-27 19:33
iOS测试应用2 ID: c7daf3c8	已通过	SDK	https://cpu.baidu.com/1003/c7daf3c8	栏目	图片频道 ID: 1003	2017-04-20 14:24

7. 可能遇到的问题

- 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.
- 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.
- 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.
- 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
- 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。



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。