

百度百青藤

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

| | |
|------------------|----|
| 概 述 | 4 |
| 说明 | 4 |
| 背景 | 4 |
| 广告位创建流程 | 5 |
| 支持广告类型 | 6 |
| SDK 项目部署 | 7 |
| 说明 | 7 |
| 手动部署 | 7 |
| 其他配置 | 8 |
| 代码接入 | 10 |
| Banner—横 幅 | 10 |
| 简介 | 10 |
| 主要 API..... | 10 |
| 接入代码示例 | 11 |
| 开屏广告 | 13 |
| 简介 | 13 |
| 主要 API..... | 14 |
| 接入代码示例 | 15 |
| 插屏广告 | 16 |
| 简介 | 16 |
| 主要 API..... | 17 |
| 接入代码示例 | 18 |
| 原生/信息流广告 | 20 |
| 简介 | 20 |
| 主要 API..... | 22 |
| 接入代码示例 | 23 |
| 信息流组件 | 25 |

| | |
|--------------|----|
| 简介 | 25 |
| 接入代码示例 | 26 |
| 接入代码示例 | 28 |
| 接入代码示例 | 28 |
| 智能优选 | 28 |
| 简介 | 29 |
| 接入代码示例 | 30 |
| 小视频 | 32 |
| 简介 | 32 |
| 接入代码示例 | 33 |
| 激励视频 | 34 |
| 简介 | 34 |
| 主要 API..... | 35 |
| 接入代码示例 | 36 |
| 贴片广告 | 38 |
| 简介 | 38 |
| 主要 API..... | 38 |
| 接入代码示例 | 39 |
| JSSDK..... | 41 |
| 简介 | 41 |
| 内容联盟 | 42 |
| 简介 | 43 |
| 接入代码示例 | 43 |
| 问题排查 | 44 |
| 说明 | 44 |
| 错误码 | 44 |

概述

说明

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。

背景

开发环境

确保您的开发及部署环境符合以下标准：

- **开发工具**：Xcode
- **部署目标**：iOS 8.0及以上版本
- **SDK版本**：官网最新版本

注册开户

开发者需在百度联盟上进行注册，在平台审核通过后，开发者就成为了百度广告联盟的正式会员。

术语介绍

- **APPID**：媒体ID

您在百度联盟创建媒体时获得的ID，这个ID是我们在广告网络中识别您应用的唯一ID。

一般为数字和字母混排，一个应用一般只有一个APPID

- **AdID**：广告位ID

您在百度联盟为您的应用创建的某种类型（Banner、开屏、插屏、原生）的广告位ID。

一般为纯数字，每个广告位归属于一个特定APPID，一个APPID可以拥有多个AdID

广告位创建流程

首先，在百度联盟MSSP平台，注册和认证您的应用，在您的应用通过审核后，可以在百度联盟MSSP平台选择“新建广告位”。

步骤说明：

1. 在创建广告位之前，确保您已经下载最新版 SDK 并阅读完《说明文档》，了解如何

在您的媒体中使用广告位 ID

2. 选择需要嵌入广告的应用名称。注意此处只可选择已通过审核的应用

3. 设置广告位的名称及格式。广告位名称用于区分媒体内设置的各个广告位，方便您在后台进行统一管理和查看数据报表；目前提供选择的广告有横幅广告、原生广告、插屏广告、贴片广告，激励视频、内容联盟和开屏广告

4. 点击完成按钮以后即获得对应该广告位的广告位 ID

5. 成功得到广告位 ID 以后，您可以在应用代码内使用该 ID 进行广告联调测试（见

“接入代码”部分）。测试时，请替换 iOS 包名，并替换应用 APPID 以及对应广告位 ID。

6. 通过测试后，与联盟商务确认后，即可发布带有联盟广告位的 iOS 应用

支持广告类型

百度联盟支持以下几种广告类型，您可以根据开发需要选择合适的广告：

| 广告类型 | 简介 | 适用场景 |
|-----------|---|----------------------------------|
| Banner 广告 | 一般位于顶部或底部，位置固定的自动刷新广告 | 文章底部或 app 底部，小说底部等 |
| 插屏广告 | 弹出式广告，用户点击广告前往广告落地页。可以选择点击关闭/倒计时关闭等多种形式。 | 在应用停顿点，适合投放这类广告 |
| 开屏广告 | 应用启动展示的广告 | 应用启动时，常会使用开屏广告 |
| 原生/信息流 | SDK 提供特定物料组合的广告物料，开发者可以自由使用和渲染广告，在自定场景内使用 | 支持返回元素/模板，媒体自主渲染控制样式，也支持利用模板渲染试用 |
| 信息流视频/小视频 | 高度可定制化，应用于特定场景的 feed 视频广告 | 全屏小视频或视频流场景使用 |
| 智能优选 | 基于普通信息流，可定制样式组合，SDK 提供组件，完成相应渲染功能 | 便于快速集成开发。且应用可直接挂载使用智能优选的渲染组件 |
| 贴片广告 | 适用于应用组件的贴片，例如视频组件 | 可在视频播放前，暂停中，播放结束，附加到视频组件上 |
| 激励视频 | 用户通过观看短视频，获取应用内奖励（免广告，任务奖励，游戏金币） | 常见游戏任务，游戏内死亡，奖励获取，注册等场景 |
| 内容联盟 | 提供 webview 资源，提供推文/自媒体文章，内含广告 | 用户需要阅读场景时 |
| JSSDK | 用户 web 页内嵌广告的支持，用法详见 JS 广告文档 | |

SDK 项目部署

说明

使用广告 SDK，您的 iOS 应用可以借助百度广告 SDK 完成广告创收。本章介绍了在 iOS 应用中集成广告 SDK 的方法。

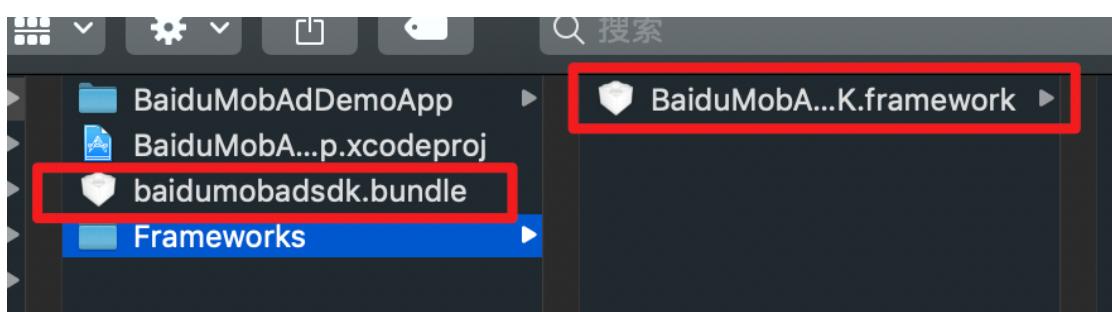
手动部署

本章会指导您手动将 iOS SDK 进行集成。如果您没有项目，请先创建一个空白项目。

添加依赖库

将解压缩后的 lib 文件 copy 到工程文件夹中，然后在项目中选中项目文件，选择 Add Files to “your project name”：

在弹出来的窗口中，请将如下文件添加到工程中：



需要引入的系统库

| 库名称 |
|--------------------------|
| StoreKit.framework |
| SafariServices.framework |

| 库名称 |
|-------------------------------|
| MessageUI.framework |
| CoreMedia.framework |
| libc++.tbd |
| CoreMotion.framework |
| SystemConfiguration.framework |
| CoreLocation.framework |
| CoreTelephony.framework |
| AVFoundation.framework |
| AdSupport.framework |

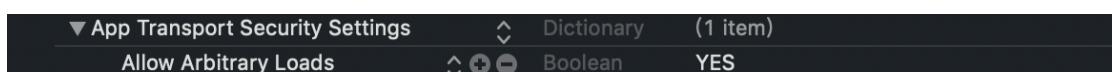
其它设置

在 Target->Build Settings -> Other Linker Flags 中添加-ObjC, 字母 o 和 c 大写。

CocoaPods 目前版本过旧，不建议使用 CocoaPods 更新，请前往百度联盟 MSSP 平台下载最新的 SDK，集成和使用

其他配置

苹果公司在 iOS9 中升级了应用网络通信安全策略，默认推荐开发者使用 HTTPS 协议来进行网络通信，并限制 HTTP 协议的请求。为了避免出现无法拉取到广告的情况，我们推荐开发者在 info.plist 文件中增加如下配置来实现广告的网络访问：（信任 HTTP 请求）



以及定位权限

| | |
|---|--------|
| Privacy - Location When In Use Usage Description | String |
| Privacy - Location Always and When In Use Usage Description | String |

若有获取 SDK 版本的需求，也可直接调用接口：

Objective-C



```
⑥ #import <UIKit/UIKit.h>
⑦
⑧ #ifndef BaiduMobAdSdk_BaiduMobAdCommonConfig_h
⑨ #define BaiduMobAdSdk_BaiduMobAdCommonConfig_h
10 // SDK版本号
11
12 #define SDK_VERSION_IN_MSSP @"4.65"
13
14 typedef void (^BaiduMobAdViewCompletionBlock)(NS
```

查看此项，可以和 MSSP 版本号对比，来判断当前 SDK 是否是最新的 SDK 版本

SDK 默认请求 https 广告，若期望请求 http 的广告，直接调用如下接口：

```
[BaiduMobAdSetting sharedInstance].supportHttps = NO
```

4.67 及以后版本适配 iOS 13。

代码接入

Banner—横幅

简介

基本信息

Banner 广告(横幅广告)位于一般位于 app 底部或顶部，Banner 广告会停留在屏幕上，并可在一段时间后自动刷新

适用场景：文章底部，app 底部/顶部，小说底部等。



推荐 20:3 宽高比

也可选择 7 : 3 , 3 : 2 , 或 2 : 1

主要 API

生命周期事件回调

您可以实现全部或部分 BaiduMobAdViewDelegate，以跟踪点击成功或者 Banner 请求失败等广告生命周期事件。

请务必先设置 delegate:

Objective-C

```
bannerView.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|-----------------------|---------|
| willDisplayAd | 广告即将载入 |
| failedDisplayAd | 广告载入失败 |
| didAdImpressed | 曝光回调 |
| didAdClicked | 广告被点击 |
| didDismissLandingPage | 广告落地页消失 |
| didAdClose | 用户点击关闭 |

接入代码示例

加载并显示 Banner 广告

1.在您的 View Controller 文件中，引入对应头文件，定义好 Banner 对象以及定义清楚本类支持 BaiduMobAdViewDelegate 协议，如下例所示：

Objective-C

```
#import "BaiduMobAdView.h"  
@interface BannerViewController () <BaiduMobAdViewDelegate>  
@property (nonatomic, strong) BaiduMobAdView *bannerView;
```

2.在 viewDidLoad 方法内，需要添加如下代码来对定义好的 BaiduMobAdView 进行初始化。

Objective-C

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    [self loadAdAndShow:nil];
}

- (BaiduMobAdView *)bannerView
{
    if (!_bannerView) {
        CGRect rect = {CGPointZero, CGSizeMake(200, 30)};
        _bannerView = [[BaiduMobAdView alloc]
                      init];
        _bannerView.AdUnitedTag = AdID;
        _bannerView.AdType = BaiduMobAdViewTypeBanner;
        _bannerView.frame = rect;
        _bannerView.presentAdViewController = self;
        [_bannerView start];
    }
    return _bannerView;
}
```

3.将 APPSID 和 AdID 替换成你自己的 APPSID 和 AdID。

4.在需要显示 bannerView 的逻辑内，需要先判断当前上下文是否已经存在一个 bannerView 的实体，如果存在，应当先将其 remove 掉，然后再进行创建 bannerView 的逻辑。

Objective-C

```
- (IBAction)loadAdAndShow:(id)sender {
    if (self.bannerView.superview) {
        [self.bannerView removeFromSuperview];
    }
    [self.view addSubview:self.bannerView];
    ...
    [self.bannerView start];
```

```
    }

    - (IBAction)removeAd:(id)sender {
        [self.bannerView removeFromSuperview];
        self.bannerView = nil;
    }
}
```

注意：

1. MSSP 平台广告位配置的尺寸比例，和前端初始化尺寸必须保持一致。否则渲染会出错
2. 宽度最小保证 200，高度最低 30。且为了保证比例正确，使用浮点数做运算，切勿随意取整导致宽高错误，导致边缘渲染缺失
3. 在释放 bannerView 之前，建议在先将 bannerView 对象的 delegate 属性置为空，然后再释放 bannerView

开屏广告

简介

基本信息

开屏广告在 App 启动时展现。用户可以点击广告跳转到广告落地页；或者点击右上角的“跳过”按钮，跳转到 app 内容首页

适用场景：开屏广告会在您的应用开启时加载，展示完毕后自动关闭并进入您的应用主界面。展示时间和跳过广告样式，可以通过 MSSP 后台配置并配合前端自定义修改。

分类：开屏广告分为半屏和全屏，其中半屏开屏广告支持开发者控制广告容器，设置开屏底部的界面，用以展示媒体 LOGO。



主要 API

生命周期事件回调

您可以实现全部或部分 `BaiduMobAdSplashDelegate`，以跟踪点击成功或者开屏广告请求失败等广告生命周期事件。

Objective-C

```
self.splashAd.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|----------------------------|-----------|
| splashSuccessPresentScreen | 开屏广告成功展示 |
| splashFailPresentScreen | 开屏广告展示失败 |
| splashDidClicked | 开屏广告点击回调 |
| splashDidDismissScreen | 开屏广告关闭回调 |
| splashDidDismissLp | 广告落地页已经关闭 |
| splashDidReady | 广告加载完成 |
| splashAdLoadSuccess | 广告请求成功 |
| splashAdLoadFail | 广告请求失败 |

接入代码示例

加载并显示开屏广告

1.在 AppDelegate 头文件中导入头文件并声明实例

Objective-C

```
#import "BaiduMobAdSDK/BaiduMobAdSplash.h"
@interface AppDelegate ()<BaiduMobAdSplashDelegate>
@property (nonatomic, strong) BaiduMobAdSplash
```

2.在 AppDelegate 的实现文件中初始化并加载广告数据，开屏广告目前支持全屏开屏

和半屏开屏广告两种形式，半屏需要媒体控制容器，根据广告生命周期。

Objective-C

```
BaiduMobAdSplash *splash = [[BaiduMobAdSplash alloc] init];
```

```
splash.delegate = self;
splash.AdUnitTag = @"2058492";
splash.canSplashClick = YES;
self.splash = splash;
self.customSplashView = [[UIView
alloc]initWithFrame:self.window.frame];
self.customSplashView.backgroundColor = [UIColor whiteColor];
[self.window addSubview:self.customSplashView];
[splash loadAndDisplayUsingContainerView:baiduSplashContainer]
```

3.将 YOUR_APP_ID 和 YOUR_PLACEMENT_ID 替换成你自己的 APPSID 和 AdID。

说明：开屏容器，必须大于 200*200 且必须在屏幕内，不建议开屏旋转，但支持横屏的开屏。

开屏广告拉取超时时间

默认为 3 秒

在不展示开屏广告时执行调试

在您的 ViewController 里实现如下回调，处理开屏广告加载失败的问题：

Objective-C

```
- (void)splashFailPresentScreen:(BaiduMobAdSplash *)splash
withError:(BaiduMobFailReason)reason {
    NSLog(@"splashFailPresentScreen withError %d", reason);
    [self removeSplash];
}
```

插屏广告

简介

基本信息

插屏广告是弹出式广告，用户点击广告前往广告落地页。可以选择点击关闭/倒计时关闭等多种形式

适用场景：在应用停顿点，适合投放这类广告

分类：插屏广告分为全屏插屏，前贴插屏和暂停插屏，具体种类可以在 SDK 进行选择：



主要 API

生命周期事件回调

您可以实现全部或部分 `BaiduMobAdInterstitialDelegate`，以跟踪点击成功或者插屏广告请求失败等广告生命周期事件。

在实现上述事件回调之前，请务必先设置 delegate:

Objective-C

```
self.interstitial.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|-----------------------------------|--------------|
| interstitialSuccessToLoadAd | 插屏广告预加载成功回调 |
| interstitialFailToLoadAd | 插屏广告预加载失败回调 |
| interstitialWillPresentScreen | 插屏广告将要展示回调 |
| interstitialSuccessPresentScreen | 插屏广告视图展示成功回调 |
| interstitialDidDismissScreen | 插屏广告展示结束回调 |
| interstitialFailPresentScreen | 展示失败回调 |
| interstitialDidDismissScreen | 插屏结束 |
| interstitialDidAdClicked | 插屏广告点击回调 |
| interstitialDidDismissLandingPage | 广告落地页将要关闭 |

接入代码示例

加载并显示插屏广告

1.在您的 View Controller 文件中，引入对应头文件，定义好 BaiduMobAdInterstitial 对象以及定义清本类支持 BaiduMobAdInterstitialDelegate 协议，如下例所示：

Objective-C

```
#import "BaiduMobAdSDK/BaiduMobAdInterstitial.h"
@interface BaiduMobAdNormalInterstitialViewController : UIViewController <BaiduMobAdInterstitialDelegate>
```

```
@property (nonatomic, strong) BaiduMobAdInterstitial *  
interstitial
```

2.在需要加载插屏广告的逻辑处，初始化 interstitial 变量，并做相应展示。

Objective-C

```
- (IBAction)loadAd:(id)sender {  
    if (self.interstitial) {  
        self.interstitial.delegate = nil;  
    }  
    self.interstitial = [[BaiduMobAdInterstitialAd alloc] init];  
    self.interstitial.delegate = self;  
    self.interstitial.AdUnitedTag = AdID;  
    self.interstitial.interstitialType = BaiduMobAdViewTypeInterstitialOther/BaiduMobAdViewTypeInterstitialBeforeVideo/BaiduMobAdViewTypeInterstitialPauseVideo  
  
    [self.interstitial load];  
}  
  
- (IBAction)showAd:(id)sender {  
    [self.interstitial presentFromRootViewController:self];  
}
```

3.将 AdID 替换成你自己的 AdID。

注意：

在一次展现完成后，需要再次调用 load 方法来加载新的广告，用于下次展现。

关于释放，建议在释放 ViewController 之前将 _interstitial 的 delegate 属性设置为 nil。

插屏支持视频物料，需要在 MSSP 平台配置广告位或新建广告。

插屏可提前请求广告，监听请求成功回调，然后使用时调用展现接口。

在不展示插屏广告时执行调试

在您的 ViewController 里实现如下回调，处理插屏广告加载失败的问题：

Objective-C

```
- (void)interstitialFailToLoadAd:(BaiduMobAdInterstitialAd *)interstitial
{
    self.interstitial.delegate = nil;
    self.interstitial = nil;
}
```

原生/信息流广告

简介

基本信息

SDK API 提供特定物料组合的广告物料，开发者可以自由使用和渲染广告，在自定场景内使用

示例

一条常规的广告包含如下字段，如有需要您可以创建以下字段相关视图：

1. Logo
2. 广告大图
3. 广告标题
4. 广告描述

除此之外，开发者还可以拿到三小图广告的图片集合。

具体字段在 `BaiduMobAdNativeAdObject`，其含义如下表所示：

| 字段 | 描述 |
|--------------------|--------------------------|
| title | 广告标题 |
| text | 广告描述 |
| iconImageURLString | Logo 图 URL |
| mainImageURLString | 大图 URL |
| adLogoURLString | 【广告】字样图 URL |
| baiduLogoURLString | 百度 LOGO(目前为百香果) |
| morepics | 三小图广告的图片集合 |
| videoURLString | 视频 URL |
| videoDuration | 视频时长 |
| brandName | 品牌名/app 名 |
| materialType | 广告类型 |
| w | 大图宽度 |
| h | 大图高度 |
| style_type | 广告样式类型 (智能优选) |
| size_type | 容器宽高是尺寸 or 比例表达 (智能优选) |
| container_width | 容器宽度 (智能优选) |

| 字段 | 描述 |
|------------------|---------------|
| container_height | 容器高度 (智能优选) |

主要 API

生命周期事件回调

您可以实现全部或部分 BaiduMobAdNativeAdDelegate , 以跟踪点击成功或者原生广告请求失败等广告生命周期事件。

请务必先设置 delegate:

Objective-C

```
self.nativeAd.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|---|----------------------|
| nativeAdObjectSuccessLoad: nativeAd: | 原生广告加载广告数据成功回调 |
| nativeAdsFailLoad: nativeAd: | 原生广告加载广告数据失败回调 |
| nativeAdClicked | 原生广告点击之后回调 |
| didDismissLandingPage | 原生广告点击以后，广告落地页被关闭时回调 |

| 回调函数名 | 回调函数含义 |
|------------------|--------|
| nativeAdExposure | 信息流曝光 |

接入代码示例

加载并显示原生广告

1.在控制器头文件中加入 SDK 头文件，声明 nativeAD 变量和 nativeAD 对应的 object 对象 BaiduMobAdNativeAdObject

Objective-C

```
#import "BaiduMobAdSDK/BaiduMobAdNative.h"
@interface NativeViewController ()<BaiduMobAdNativeAdDelegate>
@property (nonatomic, strong) BaiduMobAdNativeAd *nativeAd;
@property (nonatomic, strong) NSArray *adArray;
```

2.在 ViewController 的实现文件中初始化并加载广告数据：

Objective-C

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.nativeAd = [[BaiduMobAdNativeAd alloc] init];
    self.nativeAd.publisherId = APPSID;
    self.nativeAd.adId = AdID;
    [self.nativeAd requestNativeAds];
}
```

3.将 APPSID 和 AdID 替换成你自己的 APPSID 和 AdID。

4.需要在广告内容准备就绪的时候展示广告。您需要在拿到数据的回调方法，即 nativeAdObjectSuccessLoad 里处理数据并做广告展示：

Objective-C

```
- (void)nativeAdObjectSuccessLoad:(NSArray *)nativeAds
```

```
{  
    /*广告数据拉取成功，存储并展示*/  
    self.adArray = nativeAds;  
    注意判断 nativeAds 数组和每个广告 Object 的数据是否足够渲染  
    BaiduMobAdNativeAdObject *object = nativeAds[0];  
    dispatch_async(dispatch_get_main_queue(), ^{  
        ...  
    } );  
}  
}
```

在不展示原生广告时执行调试

在您的 ViewController 里实现如下回调，处理原生广告加载失败的问题：

Objective-C

```
- (void)nativeAdsFailLoad:(BaiduMobFailReason)reason  
{  
    NSLog(@"%@", reason);  
    /*广告数据拉取失败*/  
    dispatch_async(dispatch_get_main_queue(), ^{  
        ...  
    } );  
}
```

说明：

1. 建议在释放原生广告 NativeAd 对象之前将 NativeAd 对象的 delegate 属性设置为 nil.
2. 当用户点击广告弹出内置目标页时，请勿释放 BaiduMobAdNativeAdObject 实例。
3. 广告从拉取到使用超过 30 分钟，将作为无效广告。利用 isExpired 方法验证。
4. 广告数据渲染完毕，即将展示时，必须调用 trackImpression 方法，否则将不会产生曝光记录，也无法进行计费。点击必须调用 handleClick 方法。

5. MSSP 广告位的返回物料组合配置/返回模板，当 APP 广告上线后，切勿随意变更，确保前端不会发生崩溃。且前端开发需要对物料进行判空校验，以确定物料是否满足渲染条件，若有不符需要抛弃广告，防止 crash。

信息流组件

简介

基本信息

可以帮助开发者渲染基本的原生/信息流样式，或信息流视频以及信息流模板

(html)。开发者可以在 MSSP 平台配置你的广告位的选择。

组件均需要正常请求原生/信息流广告，利用 BaiduMobAdNativeAdObject 对象初始化

一 . 原生/信息流组件

BaiduMobAdNativeAdView : 基于广告元素，构造信息流 view，支持大部分原生/信息流组合选择。

组件元素列表

| 元素名 | 含义 |
|---------------|---------|
| iconImageView | 小图/应用图标 |

| 元素名 | 含义 |
|--------------------|------------|
| mainImageView | 大图 |
| morePicsArray | 三图 |
| adLogoImageView | 【广告】图 |
| baiduLogoImageView | 【百度】图 |
| titleLabel | 标题 |
| textLabel | 描述 |
| brandLabel | 应用名/来源 |
| videoView | 视频组件 |
| webView | 模板(html)组件 |

接入代码示例

初始化

```

/**
 * 单图信息流
 */
- (id)initWithFrame:(CGRect)frame
    brandName:(UILabel *)brandLabel
    title:(UILabel *)titleLabel
    text:(UILabel *)textLabel
    icon:(UIImageView *)iconView
    mainImage:(UIImageView *)mainView;

/**
 * 多图信息流

```

```

*/
- (id)initWithFrame:(CGRect)frame
    brandName:(UILabel *)brandLabel
    title:(UILabel *)titleLabel
    text:(UILabel *)textLabel
    icon:(UIImageView *)iconView
    mainImage:(UIImageView *)mainView
    morepics:(NSMutableArray *)imageViewArray;

/**
 * 信息流视频
 * 推荐使用SDK的BaiduMobAdNativeVideoView视频组件，也可自定义视频view
传入
 */
- (id)initWithFrame:(CGRect)frame
    brandName:(UILabel *)brandLabel
    title:(UILabel *)titleLabel
    text:(UILabel *)textLabel
    icon:(UIImageView *)iconView
    mainImage:(UIImageView *)mainView
    videoView:(BaiduMobAdNativeVideoBaseView *)videoView;
/**
 * 信息流模板
 */
- (id)initWithFrame:(CGRect)frame
    webview:(BaiduMobAdNativeWebView *) webView;

```

渲染

```

/**
 * 广告渲染
 */

-
(void)loadAndDisplayNativeAdWithObject:(BaiduMobAdNativeAdObject
*)object
completion:(BaiduMobAdViewCompletionBlock)completionBlock;

```

二 . 信息流视频组件

BaiduMobAdNativeVideoAd 只可以使用信息流 Object 初始化和播放。

提供基本的渲染能力和视频播放能力，带有默认图和尾帧效果。

接入代码示例

初始化

```
- (instancetype)initWithFrame:(CGRect)frame  
andObject:(BaiduMobAdNativeAdObject *)object;
```

播放器操作

```
/**  
 * 播放  
 */  
- (void)play;  
/**  
 * 暂停  
 */  
- (void)pause;  
/**  
 * 停止  
 */  
- (void)stop;
```

三 . 信息流模板 (html) 组件

MSSP 平台，信息流广告位，选择返回模板时，必须使用信息流模板组件

BaiduMobAdNativeWebView 渲染。

接入代码示例

初始化和渲染

```
- (instancetype)initWithFrame:(CGRect)frame  
andObject:(BaiduMobAdNativeAdObject *)object
```

智能优选

简介

基本信息

智能优选是原生/信息流广告的新对接方式，可以从 MSSP 后台创建或修改广告位，使用智能优选对接信息流。

需在 MSSP 后台，信息流样式控制选择【智能优选】，从 8 种样式中挑选。

且 SDK 支持直接渲染，开发者直接挂载就可以使用。



目前 SDK 支持全部样式，包含视频。BaiduMobAdNativeAdObject 中 style_type 和上述类型一致。不强制必须使用 SDK 渲染。开发者仍可以拿到物料，自主渲染全部效果。

FeedType_PIC_BOTTOM_TITLE = 28,//大图底部文字

FeedType_PIC_TOP_TITLE = 29,//大图上部文字

FeedType_PIC_LOGO = 30,//大图上部文字+logo

FeedType_LEFT_PIC = 33,//左图右文

FeedType_RIGHT_PIC = 34,//右图左文

FeedType_GROUP_PIC = 35,//三图图文

FeedType_GROUP_PIC_LOGO = 36,//三图图文+logo

FeedType_VIDEO_TOP_TITLE = 37//视频+文字

接入代码示例

加载并显示原生广告

1.在控制器头文件中加入 SDK 头文件，声明 nativeAD 变量和 nativeAD 对应的 object 对象 BaiduMobAdNativeAdObject，和获取普通原生广告一致，但是需要替换智能优选广告位 ID。

2.在 ViewController 的实现文件中声明并加载智能优选组件：

Objective-C

```
#import "BaiduMobAdSmartFeedView.h"

//广告返回成功

//成功返回广告字段，BaiduMobAdNativeAdObject array

- (void)nativeAdObjectsSuccessLoad:(NSArray*)nativeAds{

    self.adViewArray = [NSMutableArray array];

    self.adsArray = [nativeAds copy];

    for(int i = 0; i < nativeAds.count; i++){

        BaiduMobAdNativeAdObject *object = [nativeAds
objectAtIndex:i];

        // 展现前检查是否过期，30 分钟广告将过期，如果广告过期，请放弃展示
        // 并重新请求

        if ([object isExpired]) {

            continue;

        }

        BaiduMobAdSmartFeedView *view = [[BaiduMobAdSmartFeedView
alloc] initWithObject:object frame:CGRectMake(0, 0, [UIScreen
mainScreen].bounds.size.width, 351)];
    }
}
```

```

//非智能优选广告，无法使用，需要判空

if (view) {

    [self.adViewArray addObject:view];

    UITapGestureRecognizer *tapGesture =
[[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(tapGesture:)];

    [view addGestureRecognizer:tapGesture];
}

[self.tableView reloadData];

}

```

3.将 BaiduMobAdSmartFeedView 添加到界面上。

```

BaiduMobAdSmartFeedView *view = [self.adViewArray
objectAtIndex:indexPath.row];

for (UIView *subview in cell.contentView) {

    if ([subview isKindOfClass:[BaiduMobAdSmartFeedView
class]]) {

        [subview removeFromSuperview];
    }
}

[cell addSubview: view];

```

4.需要在广告即将加载到界面前，以及点击时，触发组件的检查展现和点击广告接口

```

BaiduMobAdSmartFeedView *adView = (BaiduMobAdSmartFeedView
*)view;

[adView trackImpression];//重要！！！！！涉及展现计费

[adView handleClick]; //重要！！！！！涉及点击计费且触发点击

```

5.视频样式接口，设置静音：

```
[view setVideoMute:NO];
```

说明：

1. 初始化组件，以传入的 frame 的宽度重新计算高度，所以需要调用 BaiduMobSmartView 的 viewWidth 和 viewHeight 接口获取真实宽高。
2. 如果 BaiduMobAdNativeAdObject 不是智能优选的广告，无法渲染，返回为 nil。
3. 点击（ handleClick ）和展现检查（ trackImpress ）务必使用组件提供的接口。
4. 组件提供 5 组控制间距和字体的接口，调用完需 reSize 刷新组件，且修改后必须重新获取组件高度。更新支持设置图片背景颜色

小视频

简介

基本信息

当开发者，需要为全屏小视频增加全屏视频广告/全屏图文广告时，可以选择接入小视频。小视频广告也是原生/信息流广告的一种定制场景扩展

生命周期事件回调

您可以实现全部或部分 BaiduMobAdVideoViewDelegate，以跟踪点击成功或者是视频广告生命周期事件。以及 BaiduMobAdNativeCacheDelegate

请务必先设置 delegate:

Objective-C

```
videoView.delegate = self;
```

BaiduMobAdVideoViewDelegate 回调函数列表

| 回调函数名 | 回调函数含义 |
|----------------------------------|----------|
| fullscreenVideoAdDidStartPlaying | 视频组件开始播放 |
| fullscreenVideoAdDidComplete | 视频组件播放完成 |
| fullscreenVideoAdDidFailed | 视频组件播放失败 |
| fullscreenVideoAdDidClick | 视频组件被点击 |

BaiduMobAdNativeCacheDelegate 回调函数列表

| 回调函数名 | 回调函数含义 |
|---------------------------|--------|
| nativeVideoAdCacheSuccess | 视频缓存成功 |
| nativeVideoAdCacheFail | 视频缓存失败 |

接入代码示例

加载并显示原生广告

1.在控制器头文件中加入 SDK 头文件，利用对应的 object 对象初始化

BaiduMobAdVideoView

Objective-C

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

```
BaiduMobAdVideoView *videoView = [[BaiduMobAdVideoView alloc] initWithFrame:CGRectMake(0, 0, kScreenWidth, self.view.frame.size.height) withObject:object];
```

2. 在 ViewController 的实现文件中操作小视频广告组件

Objective-C

```
播放: [videoView play]  
暂停: [videoView pause]  
重播: [videoView replay]  
停止: [videoView stop]
```

说明 :

1. 小视频广告组件，准备载入，即将离开界面，等需要开发者操作播放或暂停
2. 小视频组件进入前台或切入后台，需要开发者操作播放或暂停
3. 页面销毁时需要同时销毁小视频组件
4. BaiduMobAdVideoView 开发者可自定义元素添加到组件内渲染

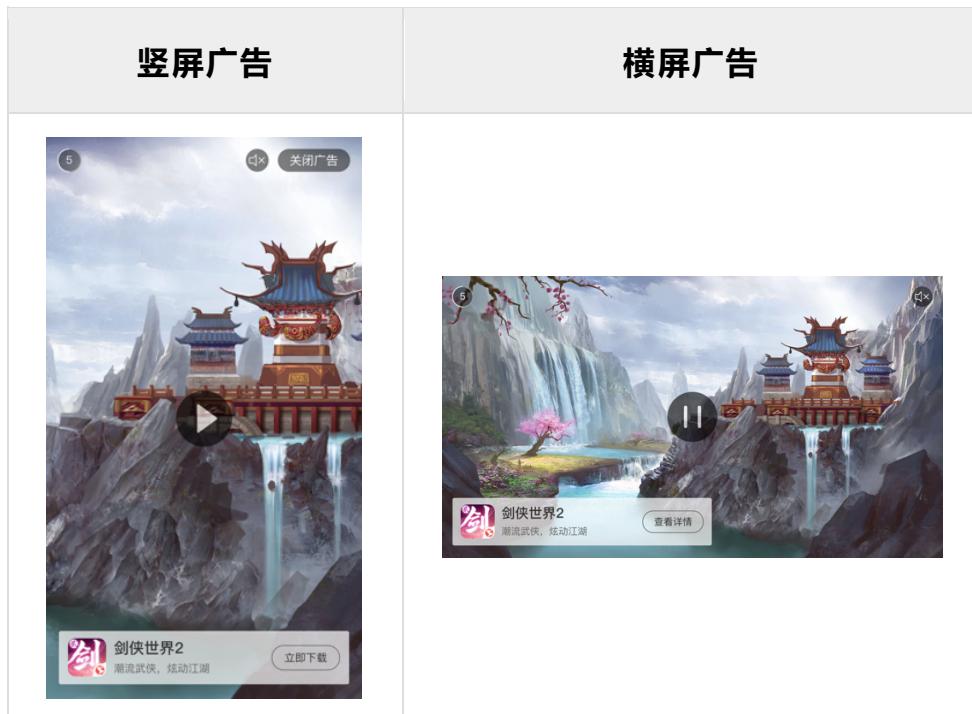
激励视频

简介

基本信息

激励视频广告是用户通过观看短视频，获取应用内奖励（免广告，任务奖励，游戏金币）等。开发者定义触发点。

分类：目前的视频包括横版及竖版 2 种样式，您可以根据需要创建对应的广告位。样式图如下：



主要 API

激励视频广告生命周期事件回调

您可以实现全部或部分 `BaiduMobAdRewardVideoDelegate`，以跟踪点击成功或者激励视频广告请求失败等广告生命周期事件。

请务必先设置 delegate:

Objective-C

```
self.rewardVideoAd.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|---------------------------|--------|
| rewardedVideoAdLoaded | 视频缓存成功 |
| rewardedVideoAdLoadFailed | 视频缓存失败 |

| 回调函数名 | 回调函数含义 |
|------------------------------|--------------------|
| rewardedVideoAdDidStarted | 激励视频开始播放 |
| rewardedVideoAdShowFailed | 广告展示失败 |
| rewardedVideoAdDidPlayFinish | 激励视频广告播放完成 |
| rewardedVideoAdDidClose | 激励视频广告关闭 (附带播放进度) |
| rewardedVideoAdDidClick | 用户点击广告/下载 (附带播放进度) |
| rewardedAdLoadSuccess | 激励广告请求成功 |
| rewardedAdLoadFail | 激励广告请求失败 |

接入代码示例

加载并显示激励广告

1.在控制器头文件中加入 SDK 头文件，声明 rewardVideoAd 属性

Objective-C

```
#import "BaiduMobAdSDK/BaiduMobAdRewardVideo.h"
@interface RewardVideoViewController ()<BaiduMobAdRewardVideoDelegate>
@property (nonatomic, strong) BaiduMobAdRewardVideo *rewardVideoAd;
```

2.在 ViewController 的实现文件中初始化并加载广告数据：

Objective-C

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.rewardVideoAd = [[BaiduMobAdRewardVideo alloc] init];
```

```

    self.rewardVideoAd.AdUnitedTag = AdID;
    self.rewardVideoAd.publisherId = APPSID
    self.rewardVideoAd.delegate = self;
    [self.rewardVideoAd load];
}

```

3.将 APPSID 和 AdID 替换成你自己的 APPSID 和 AdID。

4.需要在广告内容准备就绪的时候展示广告。推荐您在收到激励视频数据下载成功回调

之后，再在合适的时机做激励视频的展示：

Objective-C

```

- (void)rewardedVideoAdLoaded:(BaiduMobAdRewardVideo *)rewardedVideoAd
{
    NSLog(@"视频文件加载成功");
}

// 确认激励视频广告有效性之后，在合适的时机展示广告
- (IBAction)playVideo:(UIButton *)sender {
    if (![self.rewardVideoAd isReady]) {
        selfStatusLabel.text = @"广告已过期，请重新拉取";
        return;
    }
    [self.rewardVideoAd showFromViewController:self];
}

```

在不展示激励视频广告时执行调试

在您的 ViewController 里实现如下回调，处理激励视频广告加载失败的问题：

Objective-C

```

- (void)rewardedVideoAdLoadFailed:(BaiduMobAdRewardVideo *)rewardedVideoAd withReason:(BaiduMobFailReason)reason
{
    NSLog(@"ERROR: %@", reason);
}

```

说明：

1. 加载激励视频，缓存视频信息在本地。播放时调取本地播放。（不做该操作则在线播放，在线播放可能存在视频加载卡顿现象）
2. 建议在释放激励视频广告 `BaiduMobAdRewardVideo` 对象之前将 `BaiduMobAdRewardVideo` 对象的 `delegate` 属性设置为 `nil`.
3. 当用户点击广告弹出内置目标页时，请勿释放 `BaiduMobAdRewardVideo` 实例。
4. 广告从拉取成功会提供 `isReady` 属性表示广告的是否有效，展示广告之前请做广告的有效性检查，广告如果失效请提前重新拉取，否则将作无法展示广告。

贴片广告

简介

基本信息

贴片广告是常用于视频组件，用于给视频贴片，在视频预加载，暂停或结束时使用。

适用场景：视频组件的自然停顿点，适合投放这类广告

主要 API

生命周期事件回调

您可以实现全部或部分 `BaiduMobAdPrerollDelegate`，以跟踪点击成功或者贴片广告

请求失败等广告生命周期事件。

在实现上述事件回调之前，请务必先设置 delegate:

Objective-C

```
self.preroll.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|----------------------|--------------|
| prerollAdloadSuccess | 贴片广告预加载成功回调 |
| prerollAdLoadFail | 贴片请求失败 |
| prerollAdDidStart | 贴片广告视图展示成功回调 |
| prerollAdDidFinish | 贴片广告展示结束回调 |
| prerollAdDidFailed | 展示失败回调 |
| prerollAdDidClicked | 贴片广告点击回调 |

接入代码示例

加载并显示贴片广告

1. 在您的 View Controller 文件中，引入对应头文件，定义好 BaiduMobAdPreroll 对象以及定义清本类支持 BaiduMobAdPrerollDelegate 协议，如下例所示：

Objective-C

```
#import "BaiduMobAdSDK/BaiduMobAdPreroll.h"  
@interface BaiduMobAdNormalPrerollViewController :  
UIViewController <BaiduMobAdPrerollDelegate>  
  
@property (nonatomic, strong) BaiduMobAdPreroll * preroll
```

2.在需要加载贴片广告的逻辑处，初始化 preroll 变量，并做相应展示。

Objective-C

```
- (IBAction)loadAd:(id)sender {
    if (self.preroll) {
        self.preroll.delegate = nil;
    }
    self.preroll = [[BaiduMobAdPreroll alloc] init];
    self.preroll.delegate = self;
    self.preroll.adId = AdID;
    self.preroll.publisherId = APPSID;
    self.preroll.renderBaseView = self.baseView;

    [self.preroll load];
}
```

3.将 AdID 替换成你自己的 AdID。

4.新增支持，可以贴片支持广告尺寸重置。

```
[self.prerollAd reSize];//无动画的切换。以当前容器尺寸变化适配
[self.prerollAd reSizeInAnimateDuration:0.5
targetFrame:CGRectMake(0, 0, CGRectGetHeight([UIApplication
sharedApplication].keyWindow.bounds),
CGRectGetWidth([UIApplication
sharedApplication].keyWindow.bounds))];//有动画的切换，需要和尺寸
```

注意：

- 在一次展现完成后，需要再次调用 load 方法来加载新的广告，用于下次展现。
- 关于释放，建议在释放 ViewController 之前将_preroll 的 delegate 属性设置为 nil

在不展示贴片广告时执行调试

在您的 ViewController 里实现如下回调，处理贴片广告加载失败的问题：

Objective-C

```
- (void)prerollAdDidFailed:(BaiduMobAdPreroll *)preroll withReason:(BaiduMobFailReason)reason
{
}
```

JSSDK

简介

基本信息

对 JS 广告 (webview) 的端能力支持，接入方式详见 JS 广告文档，SDK 此处仅进行功能补充。不涉及广告渲染。目前支持 WKWebView，因 iOS13 废弃 UIWebView，所以不再支持 UIWebView

请务必先设置 delegate:

Objective-C

```
self.manager.delegate = self;
```

回调函数列表

| 回调函数名 | 回调函数含义 |
|---------------|-----------|
| failDisplayAd | js 广告展示失败 |
| didAdImpress | js 广告展示成功 |
| didAdClick | Js 广告被点击 |

接入代码示例

1.在控制器头文件中加入 SDK 头文件，声明 manager 变量

Objective-C

```
#import "BaiduMobAdSDK/BaiduMobAdHybridAdManager.h"  
@interface HybridViewController ()< BaiduMobAdHybridAdManagerDelegate>  
@property (nonatomic, strong) BaiduMobAdHybridAdManager *manager;
```

2.在 ViewController 的实现文件中初始化并注入 webview :

Objective-C

```
- (void)viewDidLoad  
{  
    self.manager = [[BaiduMobAdHybridAdManager alloc] init];  
    self.manager.publisherId = APPSID;  
    self.manager.delegate = self;  
    [self.manager injectJavaScriptBridgeForWKWebView:webView];  
}
```

3.在 webview 事件内触发广告事件

Objective-C

```
- (void)webView:(WKWebView *)webView decidePolicyForNavigationAction:(WKNavigationAction *)navigationAction decisionHandler:(void (^)(WKNavigationActionPolicy))decisionHandler {  
    //当 shouldStartLoadForNavigationAction 返回 no 时，表示 BaiduMobAdHybridAdManager 可处理这次回调，调用 decisionHandler(WKNavigationActionPolicyCancel);  
    BOOL shouldLoad = [self.manager webView:webView shouldStartLoadForNavigationAction:navigationAction];  
    if (!shouldLoad) {  
        decisionHandler(WKNavigationActionPolicyCancel);  
    } else {  
        decisionHandler(WKNavigationActionPolicyAllow);  
    }  
}
```

内容联盟

简介

基本信息

当应用需要资讯内容和广告混合展示时，可以使用此产品。目前使用 webview 加载，开发者暂不可以原生渲染。

接入代码示例

初始化并显示内容联盟

```
#import "BaiduMobAdSDK/BaiduMobCPUInfoManager.h"
NSString *urlStr = [[BaiduMobCpuInfoManager shared]
getCpuInfoUrlWithChannelId:channelId appId:@"d77e414"];
NSString *urlStr = [[BaiduMobCpuInfoManager shared]
getCpuInfoUrlWithChannelId:self.cpuChannelId
appId:@"d77e414"];
if (urlStr) {
    NSURL *url = [NSURL URLWithString:urlStr];
    [self.webView loadRequest:[NSMutableURLRequest
requestWithURL:url]];
}
```

页面即将消失时移除 webview

```
[self.webView removeFromSuperview];
```

问题排查

说明

如果根据正常的注册流程仍然无法在嵌入百度广告 SDK 的 app 中看到广告，可以在各个广告形式的 delegate 失败回调方法中输出错误信息。

错误码

| 错误码 | 解释 |
|----------------|---------------------------|
| 0 | 无广告返回 |
| 0100000 | 请求错误 |
| 0101000 | 广告配置缺失 |
| 0101001 | 广告配置缺失 |
| 0103000 | 无广告返回 |
| 0103010 | APPID 缺失 |
| 0103011 | APPID 失效或错误 |
| 0103012 | APPID 失效或错误 |
| 0103020 | APPID 失效或错误 |
| 0103030 | 无广告返回 |
| 0103040 | 无广告返回 |
| 0103050 | 无广告返回 |
| 0103060 | 应用包名信息错误，请保证注册包名和实际请求包名一致 |
| 0104000 | 无广告返回 |
| 0104010 | 无广告返回 |

| | |
|----------------|-------|
| 0104011 | 无广告返回 |
| 0104020 | 无广告返回 |
| 0104021 | 无广告返回 |
| 0104030 | 无广告返回 |
| 0104040 | 无广告返回 |
| 0104050 | 无广告返回 |
| 0104060 | 无广告返回 |
| 0104070 | 无广告返回 |
| 0104071 | 无广告返回 |
| 0104090 | 无广告返回 |
| 0104100 | 无广告返回 |
| 0104110 | 无广告返回 |
| 0105000 | 无广告返回 |
| 0105010 | 无广告返回 |
| 0105011 | 无广告返回 |
| 0105020 | 无广告返回 |
| 0105021 | 无广告返回 |
| 0105030 | 无广告返回 |
| 0105031 | 无广告返回 |
| 0105040 | 无广告返回 |
| 0105041 | 无广告返回 |
| 0105050 | 无广告返回 |
| 0105060 | 无广告返回 |
| 0106000 | 无广告返回 |
| 0106001 | 无广告返回 |
| 0106010 | 无广告返回 |
| 0106020 | 无广告返回 |
| 0106030 | 无广告返回 |

| | |
|----------------|---------------------|
| 0107000 | 广告位 ID 缺失 |
| 0107001 | 广告位 ID 错误或失效 |
| 0107002 | 广告位 ID 错误或失效 |
| 0107003 | 广告位 ID 与 APPSID 不匹配 |
| 0107010 | 广告位信息缺失 |
| 0107020 | 广告位信息缺失 |
| 0107030 | 广告位信息缺失 |
| 0107040 | 广告位信息缺失 |
| 0200000 | 无广告返回 |
| 0201000 | 无广告返回 |
| 0201010 | 广告信息错误 |
| 0201020 | 广告信息错误 |
| 0201021 | 广告信息错误 |
| 0201030 | 广告信息错误 |
| 0201031 | 广告信息错误 |
| 0201040 | 广告信息错误 |
| 0201050 | 广告信息错误 |
| 0201060 | 广告信息错误 |
| 0201070 | 广告信息错误 |
| 0201080 | 广告信息错误 |
| 0201090 | 广告信息错误 |
| 0201100 | 广告信息错误 |
| 0201110 | 广告信息错误 |
| 1010004 | 控件/容器太小，或为空 |
| 1030001 | 权限获取异常 |
| 1030002 | ID 设置有误 |
| 3010001 | 广告解析错误 |
| 3010003 | 物料不支持 |

| | |
|----------------|----------|
| 3010004 | 跳转类型不支持 |
| 3030002 | 物料缓存错误 |
| 3030003 | 物料缓存不存在 |
| 3040001 | 广告没有有效展现 |
| 4010004 | 点击跳转失败 |