

ADMobGenSDK iOS接入文档 Nv1.7.0

ADMobGenSDK iOS接入文档 Nv1.7.0

[修订历史](#)

[导入ADMobGenSDK](#)

[1.1 概述](#)

[2.1 采用cocoapods进行SDK的导入](#)

[2.2 手动导入SDK方式](#)

[3.1 工程环境配置](#)

[4.1 集合SDK的初始化](#)

[4.2 开屏广告 - ADMobGenSplashAd](#)

[4.3 banner广告 - ADMobGenBannerView](#)

[4.4 信息流模板广告 - ADMobGenNativeExpressAd](#)

[4.5 激励视频广告 - ADMobGenRewardvodAd](#)

[4.6 插屏广告 - ADMobGenInterstitialAd](#)

[4.7 全屏视频广告 - ADMobGenFullscreenVideoAd](#)

[4.8 原生自渲染广告 - ADMobGenUnifiedNativeAd](#)

[问题排查](#)

[作者](#)

[商务合作](#)

修订历史

文档版本	修订日期	修订说明
v1.5.1	2019-3-1	同一广告样式支持多个广告位ID、信息流广告样式优化、信息和banner广告提供广告关闭按钮、统一广告来源logo、优化广点通广告关闭回调、修复已知bug
V1.5.2	2019-3-13	新增draw沉浸式视频广告、ADMoblie广告SDK品牌升级、修复已知bug、新增Swift Demo
v1.5.4	2019-5-10	新增支持inmobi平台、Mobvsita平台、修复已知bug、新增信息流三小图样式、广点通SDK更新适配
V1.5.5	2019-5-26	新增支持谷歌平台、更新广点通和头条SDK、Mobvsita平台激励视频上线、激励视频优化、支持广点通banner2.0、deeplink广告优化、修复已知bug
V1.5.7	2019-7-9	可自自定义信息流模版广广告的样式、更新头条SDK、修复已知bug
V1.5.8	2019-8-12	头条的banner仅支持模板banner、更新广点通、头条SDK、修复已知bug
v1.6.1	2019-09-16	各平台适配iOS13、信息流模板广告新增双图双文样式、原生广告优化、移除部分多余的信息流模板广告样式、ADMoblie支持激励视频广告
v1.6.2	2019-11-14	升级各平台SDK、各平台初始化时机调整、ADMoblie平台广告优化，新增信息流视频广告、banner支持自定义刷新、修复已知bug
v1.6.3	2019-12-10	新增插屏广告、头条SDK禁用自渲染类型：信息流更新为模板信息流广告，激励视频更新为模板激励视频广告、更新广点通、头条等平台广告SDK、不接管banner关闭按钮、不接管头条和广点通的信息流模板广告的关闭按钮、修复已知bug
v1.6.5	2020-02-04	Xcode低版本适配、各广告位增加展示回调、信息流模板广告增加加载状态、升级各平台SDK、修复已经bug
v1.6.6	2020-02-20	开屏内部逻辑优化、定位权限分离、信息流模板广告统一不接管关闭按钮、支持头条沉浸式模板广告、最低支持iOS9.0、修复已知bug、广点通升级为4.11.5版本
v1.6.8	2020-05-07	开屏内部逻辑优化、汇量平台的开屏、bannner广告使用模版、增加全屏视频广告、升级各平台SDK、修复已知bug
v1.7.0	2020-06-19	SDK启动速度优化、修复已知bug

导入ADMobGenSDK

注：1、ADMobGen广告SDK品牌升级，1.5.2版本之后，ADMobGenMCAd必须更换为ADMobGenAd，否则会导致崩溃；

2、GDT平台4.11.2版本的banner尺寸异常，需要升级到banner2.0，如有发现联系ADMobGen方的媒介同学。

推荐使用pod命令

```
// 挑选其中需要导入的平台
pod 'ADMobGenSDK'
pod 'ADMobGenAd'
pod 'ADMobGenGDT'
pod 'ADMobGenSDKWM'
pod 'ADMobGenInmobi'
pod 'ADMobGenMTG'
// 根据情况酌情考虑是否导入
pod 'ADSuyiLocationManagerGPS' // 含有系统定位代码
```

常用SDK最新的版本号

```
// 基础库
ADMobGenFoundation (1.0.2)
ADMobGenAdapter (1.7.5)
ADMobGenNetwork (0.5.5)
ADMobGenSDK (1.7.0)
ADMobGenDeviceInfo (1.8.0)
ADSuyiLocationManager (0.1.2.05261)
ADSuyiLocationManagerGPS (0.1.2.05261)
// admob平台
ADMobGenAd (3.1.7)
// 广点通平台
ADMobGenGDT (4.11.8.1)
// 头条平台
ADMobGenSDKWM (2.9.5.8)
// 谷歌平台
ADMobGenGoogle (7.60.0)
// inmobi 平台
ADMobGenInmobi (7.4.6)
// mobvista 平台
ADMobGenMTG (6.3.2.0)
// 百度平台
ADMobGenBaidu (4.68)
```

1.1 概述

尊敬的开发者朋友，欢迎您使用艾狄墨搏集合SDK平台。通过本文档，您可以在几分钟之内轻松完成广告的集成过程。

操作系统：iOS 9.0 及以上版本

运行设备：iPhone（iPad上可能部分广告正常展示，但是存在填充很低或者平台不支持等问题，建议不要在iPad上展示广告）

- ADMobGenSDK Objective-C Demo地址 [\[ADMobGenSDK Objective-C Demo\]](#)
- ADMobGenSDK Swift Demo地址 [\[ADMobGenSDK Swift Demo\]](#)

2.1 采用cocoapods进行SDK的导入

SDK为了支持插件化的广告平台，采用了cocoapods进行广告集成，只需要pod对应平台，就可以集成该平台广告，无须进行其他配置（可指定版本）

注意：如果不导入ADMobGenSDK(艾狄墨搏)平台会导致崩溃

- [\[Link\]](#) 广告调用framework

```
pod 'ADMobGenSDK'
```

- [\[Link\]](#) 广点通平台

```
pod 'ADMobGenGDT'
```

- [\[Link\]](#) 艾狄墨搏平台

```
pod 'ADMobGenAd'
```

- [\[Link\]](#) 头条平台

```
pod 'ADMobGenSDKWM'
```

- [\[Link\]](#) 百度平台

```
pod 'ADMobGenBaiDu'
```

- [\[Link\]](#)(<http://121.41.108.203/ADMobGenKit-Modules/ADMobGenInmobi>) Inmobi平台

```
pod 'ADMobGenInmobi'
```

- [Link]](<http://121.41.108.203/ADMobGenKit-Modules/ADMobGenGoogle>) Google平台

注：Google平台导入之后，必须在项目中的infoplist文件中添加GADApplicationIdentifier字段，该字段的值可以从ADMobGen媒介获取，不添加该字段会导致崩溃

```
pod 'ADMobGenGoogle'
```

- [Link]](<http://121.41.108.203/ADMobGenKit-Modules/ADMobGenMTG>) Mobvsita平台

```
pod 'ADMobGenMTG'
```

2.2 手动导入SDK方式

[点击进入SDK下载地址](#)下载各SDK拖入到工程中，在Demo中的Pods/Pods目录下也可以找到必须导入的framework

- 广告调用所需framework

[ADMobGenSDK.framework](#)

[ADMobGenAdapter.framework](#)

[ADMobGenFoundation.framework](#)

[ADMobGenNetwork.framework](#)

[ADMobGenDeviceInfo.framework](#)

[ADSuyiKit.framework](#)

[ADSuyiNetwork.framework](#)

[ADSuyiLocationManager.framework](#)

[ADSuyiLocationManagerGPS.framework](#)

- 艾狄墨搏平台

[ADMobGenAdSDK.framework](#)

- 广点通平台

[ADMobGenGDT.framework](#)

- 头条平台

同时导入 [ADMobGenWM.framework](#), [BUAdSDK.framework](#) 以及资源文件[BUAdSDK.bundle](#)

- 百度平台

同时导入 [ADMobGenBaiDu.framework](#), [BaiduMobAdSDK.framework](#) 以及资源文件[baidumobadsdk.bundle](#)

- Mobvsita平台

同时导入 [ADMobGenMTG.framework](#), [MTGSDK.framework](#) , [MTGSDKReward.framework](#)

- Google平台 注: Google平台导入之后, 必须在项目中的infoplist文件中添加 `GADApplicationIdentifier` 字段, 该字段的值可以从ADMoblie媒介获取, 不添加该字段会导致崩溃

同时导入 [ADMobGenGoogle.framework](#), [GoogleMobileAds.framework](#) , [GoogleUtilities.framework](#) , [GoogleAppMeasurement.framework](#)

- Inmobi平台

同时导入 [ADMobGenInmobiSDK.framework](#), [InMobiSDK.framework](#)

手动方式导入,需要添加如下依赖库:

```
AdSupport.framework•
CoreLocation.framework•
QuartzCore.framework•
SystemConfiguration.framework
CoreTelephony.framework
libz.tbd•
WebKit.framework (Optional)
libxml2.tbd
Security.framework•
StoreKit.framework
MessageUI.framework
libc++.dylib
CoreGraphics.framework
EventKit.framework
EventKitUI.framework
SafariServices.framework
CoreMotion.framework
MediaPlayer.framework
MobileCoreServices.framework(1.4.3及以后)
CoreMedia.framework(1.4.3及以后)
AVFoundation.framework
libsqlite3.dylib
Accelerate.framework
ImageIO.framework
```

以下为Mobvsita平台所需

```
UIKit.framework
Foundation.framework
AVFoundation.framework
```

头条还需要添加依赖库:

```
libresolv.9.tbd(1.4.3及以后)
```

3.1 工程环境配置

1. 打开项目的 app target, 查看 Build Settings 中的 Linking-Other Linker Flags 选项, 确保含有 -ObjC 一值, 若没有则添加。
2. 在项目的 app target 中, 查看 Build Settings 中的 Build options - Enable Bitcode 选项, 设置为 NO。
3. info.plist 添加支持 Http访问字段

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
```

4. Info.plist 添加定位权限字段 (使用ADMobGenLocation可不设置)

```
NSLocationWhenInUseUsageDescription
NSLocationAlwaysAndWhenInUseUsageDeion
```

5. Info.plist推荐设置白名单

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>dianping</string>
  <string>imeituan</string>
  <string>com.suning.SuningEBuy</string>
  <string>openapp.jdmobile</string>
  <string>vipshop</string>
  <string>snssdk141</string>
  <string>ctrip</string>
  <string>suning</string>
  <string>qunariphone</string>
  <string>QunarAlipay</string>
  <string>qunaraphone</string>
  <string>yohobuy</string>
  <string>kaola</string>
  <string>agoda</string>
  <string>openapp.xzdz</string>
  <string>beibeiapp</string>
  <string>taobao</string>
  <string>tmall</string>
  <string>openjd</string>
  <string>jhs</string>
  <string>yhd</string>
  <string>wirelessl688</string>
  <string>GomeEShop</string>
  <string>wbmain</string>
```

```
<string>xhsdiscover</string>  
<string>douyin</string>  
<string>pinduoduo</string>  
<string>jdmobile</string>  
<string>tbopen</string>  
<string>pddopen</string>  
<string>mogujie</string>  
<string>koubei</string>  
<string>eleme</string>  
<string>youku</string>  
</array>
```


4.1 集合SDK的初始化

申请的appid和你的包名相对应，根据双方商务协调，去确定需要拉取的广告平台

```
[ADMobGenSDKConfig initWithAppId:@"appid" completionBlock:^(NSError *error) {  
    if (error) {  
        // SDK启动失败  
    }  
}];
```

开启定位权限, 更加精准的投放广告

```
[ADMobGenSDKConfig setGpsOn];
```

用户日志输出等级

```
//默认为ADMobGenLogLevelNone  
[ADMobGenSDKConfig setLogLevel:ADMobGenLogLevelError];
```

获取ADMobGenSDK版本号

```
//获取SDK版本号  
NSString *sdkVersion = [ADMobGenSDKConfig getSDKVersion];
```

测试Debug包需要，一般情况下不开启,需要时会另行通知

```
//广告SDK Debug模式  
[ADMobGenSDKConfig setDebugMode:@(YES)];
```

4.2 开屏广告 - ADMobGenSplashAd

开屏广告会在您的应用开启时加载，拥有固定展示时间，展示完毕后自动关闭并进入您的应用主界面。

开屏广告 - ADMobGenSplashAd：

```
@interface ADMobGenSplashAd : NSObject

/**
代理回调
*/
@property (nonatomic, weak) id<ADMobGenSplashAdDelegate> delegate;

/**
开屏的默认背景色,或者启动页
*/
@property (nonatomic, copy) UIColor *backgroundColor;

/**
加载并展示到window中

@param window window
@param bottomView 底部logo视图，高度不能超过屏幕的25%，建议：开屏的广告图片默认640 /
960比例，可以用 screenHeight - screenWidth * (960 / 640.0) 得出bottomview的高度
(iPhone x不适用)
*/
- (void)loadAndShowInWindow:(UIWindow *)window withBottomView:(UIView
*)bottomView;

/**
加载并展示到window中

@param window window
@param bottomView 底部logo视图，高度不能超过屏幕的25%，建议：开屏的广告图片默认640 /
960比例，可以用 screenHeight - screenWidth * (960 / 640.0) 得出bottomview的高度
(iPhone x不适用)
@param splashIndex 广告位序号，默认为0
*/
- (void)loadAndShowInWindow:(UIWindow *)window withBottomView:(UIView
*)bottomView withSplashIndex:(NSInteger)splashIndex;

@end
```

开屏广告代理回调 - ADMobGenSplashAdDelegate

```
@protocol ADMobGenSplashAdDelegate <NSObject>

/**
 开屏展现成功

  @param splashAd 开屏实例
  */
- (void)admg_splashAdSuccessToPresentScreen:(ADMobGenSplashAd *)splashAd;

/**
 开屏展现失败

  @param splash 开屏实例
  @param error 错误描述
  */
- (void)admg_splashAd:(ADMobGenSplashAd *)splash failToPresentScreen:(NSError *)error;

/**
 开屏广告点击

  @param splashAd 开屏实例
  */
- (void)admg_splashAdClicked:(ADMobGenSplashAd *)splashAd;

/**
 开屏被关闭

  @param splashAd 开屏实例
  */
- (void)admg_splashAdClosed:(ADMobGenSplashAd *)splashAd;

/**
 开屏展示

  @param splashAd 广告实例
  */
- (void)admg_splashAdExposure:(ADMobGenSplashAd *)splashAd;

@end
```

开屏广告请求示例：

```
#import <ADMobGenSDK/ADMobGenSplashAd.h>

//注意：1、当你在控制器中加载开屏时，请勿在viewWillAppear中加载开屏，该方法会调用多次，使得
//展示多次开屏广告。2、在viewDidLoad中加载开屏广告的时候，如果该控制器没有用导航栏承载，会出现
//无法展示广告，却走了加载成功的回调方法。3、当你在rootViewController中有做模态弹出其他的控
//制器（包含UIAlertController）的操作时，可能会出现广告加载不出的情况（广点通广告无法加载）。
// 1 初始化
_splashAd = [[ADMobGenSplashAd alloc] init];

// 2 设置默认启动图（一般设置启动图的平铺颜色为背景颜色，使得视觉效果更加平滑）
_splashAd.backgroundColor = [UIColor getColorWithImage:[UIImage
imageNamed:@"750x1334"] withNewSize:[UIScreen mainScreen].bounds.size];

_splashAd.delegate = self;

// 3 设置底部logo视图，高度不能超过屏幕的25%，除刘海屏以外建议：开屏的广告图片默认640 /
// 960比例，如果是刘海屏注意bottomViewHeight不能超过屏幕的25%
CGFloat bottomViewHeight;
if (kiPhoneX) {
    bottomViewHeight = [UIScreen mainScreen].bounds.size.height * 0.25;
} else {
    bottomViewHeight = [UIScreen mainScreen].bounds.size.height - [UIScreen
mainScreen].bounds.size.width * (960 / 640.0);
}

UIView *bottomView = [[UIView alloc] init];
bottomView.backgroundColor = [UIColor whiteColor];
bottomView.frame = CGRectMake(0, 0, [UIScreen mainScreen].bounds.size.width,
bottomViewHeight);
// 4 设置logo，根据自己的实际情况来设置
UIImageView *logoImageView = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"ADMob_Logo.png"]];
logoImageView.frame = CGRectMake(([UIScreen mainScreen].bounds.size.width-
135)/2, (bottomViewHeight-46)/2, 135, 46);
[bottomView addSubview:logoImageView];

// 5 展示
UIWindow *window = [UIApplication sharedApplication].delegate.window;
[_splashAd loadAndShowInWindow:window withBottomView:bottomView];
//      [_splashAd loadAndShowInWindow:window withBottomView:bottomView
//withSplashIndex:0];

// 6 代理回调
#pragma mark - ADMobGenSplashAdDelegate
// 开屏展现回调
- (void)admg_splashAdSuccessToPresentScreen:(ADMobGenSplashAd *)splashAd{
```

```

}
// 加载失败调用该方法
- (void)admg_splashAd:(ADMobGenSplashAd *)splash failToPresentScreen:(NSError *)error{
    _splashAd = nil;
    if (error) {
        //NSLog(@"");ADMobGenLogLevelError自动打印错误信息，也可打印error查看
    }
}
// 加载成功，开屏被关闭调用该方法
- (void)admg_splashAdClosed:(ADMobGenSplashAd *)splashAd{
    _splashAd = nil;
}

```

开屏广告注意事项：

- 推荐在 AppDelegate 的 didFinishLaunchingWithOptions 方法的 return YES 之前调用开屏。
- 销毁开屏对象请在开屏加载失败和开屏关闭的回调中置空，切忌在点击回调中置空，可能会造成从落地页返回App的时候出现广告视图依旧存在的情况，因为开屏的生命周期是请求-展示-/落地页(点击)-/关闭。
- 当使用控制器承载开屏广告时，有以下三个需要注意的事项：
- 1、请勿在 viewWillAppear 中调用开屏广告，否则会出现开屏方法多次被调用。
- 2、当前控制器需要导航栏控制器承载，否则会出现广告请求成功却不展示的情况。
- 3、当你在rootViewController中有做模态弹出其他的控制器的操作时，可能会出现广告加载不出的情况（广点通广告无法加载）。
- 4、切勿在点击回调中将开屏实例对象置空，否则广点通下载类广告落地页无法关闭，头条广告悬停在屏幕无法消失。

4.3 banner广告 - ADMobGenBannerView

横幅广告推荐放置在页面固定位置

- 1、如果已经有广点通和头条的banner广告位id，需要确认广点通的banner广告位id是否为banner2.0，头条的banner广告id是否为模版样式；
- 2、banner的关闭按钮不接管，意味着closeButtonHidden方法不生效；
- 3、banner增加自定义刷新时间，时间30s-120s之间；
- 4、GDT平台4.11.2版本的banner尺寸异常，需要升级到banner2.0，如有发现联系ADMoblie方的媒介同学。

横幅广告目前只支持640：100尺寸：

```
typedef NS_ENUM(NSUInteger, ADMobGenBannerAdSize) {
    ADMobGenBannerAdSizeNormal = 0, //640:100
    ADMobGenBannerAdSize600_150 = 1, //600:150(禁用)
    ADMobGenBannerAdSize690_388 = 2, //690:388(禁用)
    ADMobGenBannerAdSize600_400 = 3, //600:400(禁用)
    ADMobGenBannerAdSize600_260 = 4, //600:260(禁用)
};
```

横幅广告 - ADMobGenBannerView:

```
@interface ADMobGenBannerView : UIView

/**
代理回调
*/
@property (nonatomic, weak) id<ADMobGenBannerViewDelegate> delegate;

- (id)new NS_UNAVAILABLE;
- (id)init NS_UNAVAILABLE;
/*
初始化bannerView
@param size 期望的banner广告尺寸比例，默认为640:100
*/
- (instancetype)initWithFrame:(CGRect)frame withBannerSize:
(ADMobGenBannerAdSize)bannerSize;

/*
初始化bannerView
@param size 期望的banner广告尺寸比例，默认为640:100
@param bannerIndex 广告位序号，默认为0，不同的页面使用，可以传入不同的序号
*/
- (instancetype)initWithFrame:(CGRect)frame withBannerSize:
(ADMobGenBannerAdSize)bannerSize withBannerIndex:(NSInteger)bannerIndex;
```

```

/*
初始化bannerView
@param size 期望的banner广告尺寸比例, 默认为640:100
@param bannerIndex 广告位序号, 默认为0, 不同的页面使用, 可以传入不同的序号
@param gdtBanner 是否为广点通2.0, 默认为NO不支持广点通2.0
*/
- (instancetype)initWithFrame:(CGRect)frame withBannerSize:
(ADMobGenBannerAdSize)bannerSize withBannerIndex:(NSInteger)bannerIndex
withGdtBanner:(BOOL)gdtBanner;

/*
初始化bannerView
@param size 期望的banner广告尺寸比例, 默认为640:100
@param bannerIndex 广告位序号, 默认为0, 不同的页面使用, 可以传入不同的序号
@param gdtBanner 是否为广点通2.0, 默认为NO不支持广点通2.0
@param rootViewController
*/
- (instancetype)initWithFrame:(CGRect)frame withBannerSize:
(ADMobGenBannerAdSize)bannerSize withBannerIndex:(NSInteger)bannerIndex
withGdtBanner:(BOOL)gdtBanner withViewController:(UIViewController
*)rootViewController;

/*
初始化bannerView
@param bannerIndex 广告位序号, 默认为0, 不同的页面使用, 可以传入不同的序号
@param gdtBanner 是否为广点通2.0, 默认为NO不支持广点通2.0
@param rootViewController banner被展示的控制器的控制器
@param refershTime banner广告刷新时间间隔, 30-120s之间, 其他时间无效
*/
- (instancetype)initWithFrame:(CGRect)frame withBannerIndex:
(NSInteger)bannerIndex withGdtBanner:(BOOL)gdtBanner withViewController:
(UIViewController *)rootViewController refershTime:(NSInteger)refershTime
NS_DESIGNATED_INITIALIZER;

/**
加载并显示广告
@warning 请确保banner视图显示在屏幕内的时候,调用load方法
*/
- (void)loadWithError:(NSError *__autoreleasing *)error;

/**
banner广告关闭按钮是否隐藏

@param isHidden 默认为NO, 不隐藏
*/
- (void)closeButtonHidden:(BOOL)isHidden NS_UNAVAILABLE;

@end

```

横幅广告 - ADMobGenBannerViewDelegate

```
@protocol ADMobGenBannerViewDelegate <NSObject>

/**
 广告获取成功

  @param bannerView banner实例
  */
- (void)admg_bannerViewDidReceived:(ADMobGenBannerView *)bannerView;

/**
 广告拉取失败

  @param bannerView banner实例
  @param error 错误描述
  */
- (void)admg_bannerViewFailToReceived:(ADMobGenBannerView *)bannerView error:
(NSError *)error;

/**
 广告点击

  @param bannerView 广告实例
  */
- (void)admg_bannerViewClicked:(ADMobGenBannerView *)bannerView;

/**
 广告关闭

  @param bannerView 广告实例
  */
- (void)admg_bannerViewClose:(ADMobGenBannerView *)bannerView;

/**
 广告展示

  @param bannerView 广告实例
  */
- (void)admg_bannerViewExposure:(ADMobGenBannerView *)bannerView;

@end
```

横幅广告请求示例:

```
#import <ADMobGenSDK/ADMobGenBannerView.h>

if (_bannerView) {
```



```
[_bannerView removeFromSuperview];  
_bannerView = nil;  
}
```

// 1 初始化banner视图, bannerSize为banner的宽高比, 以视图的宽度为准, 当需要使用除默认尺寸以外的请与我们的商务联系, 注意: 请求广点通的banner2.0以及头条的模板banner时, 需要传入准确的宽高

```
_bannerView = [[ADMobGenBannerView alloc] initWithFrame:CGRectMake(0, 0,  
[UIScreen mainScreen].bounds.size.width, [UIScreen  
mainScreen].bounds.size.width * 5 / 32) withBannerIndex:0 withGdtBanner:YES  
withViewController:self refershTime:35];  
_bannerView.delegate = self;  
_bannerView.backgroundColor = [UIColor redColor];
```

// 2 添加到父视图上, ADMobGenBannerView的比例请尽量与bannerSize保持一致

```
CGFloat height = [UIScreen mainScreen].bounds.size.width * (5 / 32.0);  
CGFloat width = [UIScreen mainScreen].bounds.size.width;  
_bannerView.frame = CGRectMake(0, [UIScreen mainScreen].bounds.size.height -  
height, width, height);  
[self.view addSubview:_bannerView];
```

// 3 加载并显示广告 注意: 请确保banner视图显示在屏幕内的时候, 调用load方法

```
[_bannerView loadWithError:nil];
```

// 4 代理回调

```
#pragma mark - ADMobGenBannerViewDelegate
```

// 加载成功

```
- (void)admg_bannerViewDidReceived:(ADMobGenBannerView *)bannerView{  
NSLog(@"load banner success");  
}
```

// 加载失败

```
- (void)admg_bannerViewFailToReceived:(ADMobGenBannerView *)bannerView error:  
(NSError *)error{  
if (error) {  
//请求失败时, 将banner视图的高度置为0, 就不会出现空白区域  
CGRect rect = _bannerView.frame;  
_bannerView.frame = CGRectMake(rect.origin.x, rect.origin.y, rect.size.width,  
0);  
}  
}
```

// 广告关闭

```
- (void)admg_bannerViewClose:(ADMobGenBannerView *)bannerView{  
  
}
```

4.4 信息流模板广告 - ADMobGenNativeExpressAd

信息流模板广告是一种自动化展现的原生广告，返回的是广告视图，可以支持多个样式（广告视图生命周期和模板广告实例对象绑定）。各个样式描述如下（如有需要可联系ADMobile媒介小姐姐）：

- 1、头条和广点通的关闭不接管，意味着closeButtonHidden方法对头条广点通不生效，需要在各自后台新建广告位时去设置，并且头条模板广告样式设置有限。
- 2、头条19年11月28日渲染代码位新建入口关闭，之后新增的信息流广告位都需要设置buNativeExpressType为YES，不导入头条SDK请忽略

```
//同类型的高度一致，以真实返回视图的高度为准，如有其他需求请联系我方商务
typedef NS_ENUM(NSUInteger, ADMobGenNativeAdType) {
    ADMobGenNativeAdTypeNormal = 0, //上图下文，默认为图文，图片比例为16：9
    ADMobGenNativeAdTypePic, //纯图片，图片比例16：9
    ADMobGenNativeAdTypeRightPic, //右图左文，图片比例3：2
    ADMobGenNativeAdTypeLeftPic, //左图右文，图片比例3：2
    ADMobGenNativeAdTypeCenterPic, //上文下图，图片比例16：9
    ADMobGenNativeAdTypeVerticalPic, //竖版纯图，图片比例3：2
    // 以上的广告支持图文和视频混合，默认静音，4G下不会自动播放。以下的广告类型是纯视频流，填充率会降低，并且头条的视频流，是无法关闭声音的，广点通的视频流默认静音，4G下不会自动播放。
    ADMobGenNativeAdTypeThreePic, //三小图，inmobi、Mobvsita不支持
    ADMobGenNativeAdTypeDoublePic, //双图双文
};
```

信息流模板广告 - ADMobGenNativeExpressAd：

```
typedef NS_ENUM(NSUInteger, ADMobGenNativeExpressAdStatus) {
    ADMobGenNativeExpressAdStatusDefault, // 闲置状态
    ADMobGenNativeExpressAdStatusLoading, // 加载广告中
};

@interface ADMobGenNativeExpressAd : NSObject

/*
    viewControllerForPresentingModalView
    详解：[必选]开发者需传入用来弹出目标页的ViewController，一般为当前ViewController
*/
@property (nonatomic, weak) UIViewController *controller;

/*
    status
    详解：开发者可通过状态值来判断当前广告对象是否正在加载广告，支持kvo
*/
@property (nonatomic, assign) ADMobGenNativeExpressAdStatus status;

/**
```

```

代理回调
*/
@property (nonatomic, weak) id<ADMobGenNativeExpressAdDelegate> delegate;

- (id)new NS_UNAVAILABLE;
- (id)init NS_UNAVAILABLE;

/**
信息流模板广告构造方法

@param size 期望的广告大小,高度自适应
*/
- (id)initWithSize:(CGSize)size;

/**
信息流模板广告构造方法

@param size 期望的广告大小,高度自适应
@param nativeAdType 信息流广告类型
@param flowIndex 广告位序号,默认为0,不同的页面使用,可以传如不同的序号
*/

- (id)initWithSize:(CGSize)size withNativeAdType:
(ADMobGenNativeAdType)nativeAdType withFlowIndex:(NSInteger)flowIndex
NS_DESIGNATED_INITIALIZER;

/**
信息流模板广告关闭按钮是否隐藏

@param isHidden 默认为NO,不隐藏
*/
- (void)closeButtonHidden:(BOOL)isHidden;

/**
加载广告

@param count 拉取几条广告,建议区间 1~5, 超过可能无法拉取到
*/
- (void)load:(int)count;

/**
设置原生广告类型

@param nativeAdType 广告渲染的样式,默认为图文
*/
- (void)setNativeAdType:(ADMobGenNativeAdType)nativeAdType;

/**
头条的新版信息流模板广告

```

```

    @param buNativeExpressType 头条19年11月28日自渲染代码位新建入口关闭，之后新增的信息流
    广告位都需要设置为YES，不导入头条SDK请忽略
    */
- (void)setBuNativeExpressType:(BOOL)buNativeExpressType;

@end

```

信息流模板广告代理回调 - ADMobGenNativeExpressAdDelegate

```

@protocol ADMobGenNativeExpressAdDelegate <NSObject>

/**
广告模板请求成功

@param nativeExpressAd 广告实例
@param views 广告模板集合
*/
- (void)admg_nativeExpressAdSucessToLoad:(ADMobGenNativeExpressAd
*)nativeExpressAd views:(NSArray<__kindof ADMobGenNativeExpressAdView *>
*)views;

/**
广告模板请求失败

@param nativeExpressAd 广告实例
@param error 请求错误描述
*/
- (void)admg_nativeExpressAdFailToLoad:(ADMobGenNativeExpressAd
*)nativeExpressAd error:(NSError *)error;

/**
广告模板渲染成功
如在列表中,可在此会调用刷新列表,重新调整高度

@param nativeExpressAdView 广告模板
*/
- (void)admg_nativeExpressAdViewRenderSuccess:(ADMobGenNativeExpressAdView
*)nativeExpressAdView;

/**
广告模板渲染失败

@param nativeExpressAdView 广告模板
*/
- (void)admg_nativeExpressAdViewRenderFail:(ADMobGenNativeExpressAdView
*)nativeExpressAdView;

```

```

/**
广告模板被点击

@param nativeExpressAdView 广告模板
*/
- (void)admg_nativeExpressAdViewClicked:(ADMobGenNativeExpressAdView
*)nativeExpressAdView;

/**
广告模板被关闭

@param nativeExpressAdView 广告模板
*/
- (void)admg_nativeExpressAdViewClose:(ADMobGenNativeExpressAdView
*)nativeExpressAdView;

/**
视频广告模板播放状态更换

@param nativeExpressAdView 广告模板
*/
- (void)admg_nativeExpressVideoAdView:(ADMobGenNativeExpressAdView
*)nativeExpressAdView stateDidChange:(ADMobGenPlayerState)playerState;

/**
广告模板被展示

@param nativeExpressAd 广告实例
@param nativeExpressAdView 广告模板
*/
- (void)admg_nativeExpressAdViewExposure:(ADMobGenNativeExpressAd
*)nativeExpressAd nativeExpressAdView:(ADMobGenNativeExpressAdView
*)nativeExpressAdView;

@end

```

信息流模板广告请求示例：

```

#import <ADMobGenSDK/ADMobGenNativeExpressAd.h>

if (!_expressAd) {
// 1 信息流请求对象的初始化，并声明为全局变量
ADMobGenNativeExpressAd *expressAd = [[ADMobGenNativeExpressAd alloc]
initWithSize:CGSizeMake([UIScreen mainScreen].bounds.size.width, 100)];
//ADMobGenNativeExpressAd *expressAd = [[ADMobGenNativeExpressAd alloc]
initWithSize:CGSizeMake([UIScreen mainScreen].bounds.size.width, 10)
withNativeAdType:ADMobGenNativeAdTypePic withFlowIndex:0];
expressAd.delegate = self;

```

```

// 2 设置信息流广告需要显示的控制器，保证和信息流展示的控制器是同一个
expressAd.controller = self;
// 3 设置信息流广告类型，默认为图文ADMobGenNativeAdTypeNormal，一共支持六种样式的信息流，
不同的信息流样式使用不同的实例对象，需要除normal以外其他的信息流样式可以联系ADMobile媒介或运营小姐姐
[expressAd setNativeAdType:ADMobGenNativeAdTypePic]; //信息流样式纯图片
// 4 隐藏信息流关闭按钮，头条和广点通的关闭按钮需要在其后台进行设置开关
[expressAd closeButtonHidden:YES];

_expressAd = expressAd;
}

// 5 拉取信息流模板广告
[_expressAd load:4];

// 6 代理回调
#pragma mark - ADMobGenNativeExpressAdDelegate
// 加载成功
- (void)admg_nativeExpressAdSucessToLoad:(ADMobGenNativeExpressAd
*)nativeExpressAd views:(NSArray<__kindof ADMobGenNativeExpressAdView *>
*)views {
//加入到数据源中，注意：要使ADMobGenNativeExpressAdView响应点击事件还需要调用它的
contentSize方法（请在此时添加到数据源中，否则可能出现白屏或者视频无法播放的情况）
[self.items addObjectsFromArray:views];
//成功回调中，直接返回views中的视图ADMobGenNativeExpressAdView，使用
ADMobGenNativeExpressAdView之前需要调用它的render方法进行渲染。
for (int index = 0; index < views.count; index++) {
[views[index] render];
}
[self.tableView reloadData];
}
// 加载失败回调
- (void)admg_nativeExpressAdFailToLoad:(ADMobGenNativeExpressAd
*)nativeExpressAd error:(NSError *)error {

}
// 渲染成功回调
- (void)admg_nativeExpressAdViewRenderSuccess:(ADMobGenNativeExpressAdView
*)nativeExpressAdView {
//刷新，此时webView加载可返回正确的高度
[self.tableView reloadData];

}
// 渲染失败
- (void)admg_nativeExpressAdViewRenderFail:(ADMobGenNativeExpressAdView
*)nativeExpressAdView {
//从临时数组中去除
[self.tempViewitems removeObject:nativeExpressAdView];
}

```

```
// 关闭回调
- (void)admg_nativeExpressAdViewClose:(ADMobGenNativeExpressAdView
*)nativeExpressAdView{

}
```

信息流模版广告注意事项：

- 信息流拉取成功后, 获得的 `ADMobGenNativeExpressAdView` 视图需要调用 `[adview render]` 方法, 否则无法进行广告的点击上报和展现上报。在渲染成功的回调中再调用 `render` 方法, 会再次调起渲染成功回调, 造成死循环。
- 如果同时使用到 `ADMobGenNativeAdTypeNormal` `ADMobGenNativeAdTypePic` 两种信息流类型, 是在同样的代理方法中返回 `ADMobGenNativeExpressAdView` 视图, 可调用 `[nativeExpressAdView getNativeAdType]` 来判断该视图是哪种类型
- `ADMobGenNativeExpressAd` 对象初始化传入的 `size`, 宽度会根据传入的`size`固定模板宽度, 高度会自适应, 调用方可以通过 `adview.contentSize` 获取当前信息流模板视图的详细尺寸, 如不使用 `adview.contentSize` 会出现无法点击的问题。
- 新增信息流模板广告加载状态, 强烈建议在广告加载完成之后处于闲置状态再进行下一次load。

4.5 激励视频广告 - ADMobGenRewardvodAd

激励视频广告是一种全新的广告形式，用户可选择观看视频广告以换取有价值物，例如虚拟货币、应用内物品和独家内容等等；这类广告的长度为 15-30 秒，不可跳过，且广告的结束画面会显示结束页面，引导用户进行后续动作。

1、头条19年11月28日自渲染代码位新建入口关闭，之后新增的激励视频广告位都需要设置 buRewardvodType为YES，不导入头条SDK请忽略

激励视频广告 - ADMobGenRewardvodAd：

```
@interface ADMobGenRewardvodAd : NSObject

/*
viewControllerForPresentingModalView
详解：[必选]开发者需传入用来弹出目标页的viewController，一般为当前viewController
*/
@property (nonatomic, weak) UIViewController *controller;

/**
代理回调
*/
@property (nonatomic, weak) id<ADMobGenRewardvodAdDelegate> delegate;

/**
加载激励视频广告
*/
- (void)loadVideoAd;

/**
加载激励视频广告

@param index 广告位序号，默认为0，不同的页面使用，可以传入不同的序号
*/
- (void)loadVideoAdIndex:(NSInteger)index;

/**
激励视频广告是否准备好
*/
- (BOOL)rewardvodAdIsReady;

/**
激励视频广告物料是否有效
*/
- (BOOL)rewardvodAdIsValid;

/**
展示激励视频广告
```



```

*/
- (void)showRewardvodAd;

/**
头条的激励视频模板广告

@param buRewardvodType 头条19年11月28日自渲染代码位新建入口关闭，之后新增的激励视频广告
位都需要设置为YES，不导入头条SDK请忽略
*/
- (void)setBuRewardvodType:(BOOL)buRewardvodType;

@end

```

激励视频广告代理回调 - ADMobGeRewardvodDelegate

```

@protocol ADMobGenRewardvodAdDelegate <NSObject>

/**
广告数据加载成功回调

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdLoadSuccessCallBack:(ADMobGenRewardvodAd
*)rewardvodAd;

/**
视频数据下载成功回调，已经下载过的视频会直接回调

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdVideoLoadSuccessCallBack:(ADMobGenRewardvodAd
*)rewardvodAd;

/**
视频播放页即将展示回调

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdWillVisibleCallBack:(ADMobGenRewardvodAd
*)rewardvodAd;

/**
视频广告曝光回调

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdDidVisibleCallBack:(ADMobGenRewardvodAd *)rewardvodAd;

/**
视频播放页关闭回调

@param rewardvodAd 广告实例

```

```

*/
- (void)admg_rewardvodAdDidCloseCallBack:(ADMobGenRewardvodAd *)rewardvodAd;
/**
视频广告信息点击回调

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdDidClickCallBack:(ADMobGenRewardvodAd *)rewardvodAd;
/**
视频广告视频播放完成

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdDidPlayFinishCallBack:(ADMobGenRewardvodAd
*)rewardvodAd;

/**
视频广告视频达到奖励条件

@param rewardvodAd 广告实例
*/
- (void)admg_rewardvodAdDidRewardEffective:(ADMobGenRewardvodAd *)rewardvodAd;

/**
视频广告各种错误信息回调

@param rewardvodAd 广告实例
@param error 具体错误信息
*/
- (void)admg_rewardvodAdFailToLoadCallBack:(ADMobGenRewardvodAd *)rewardvodAd
error:(NSError *)error;

@end

```

激励视频广告 - ADMobGenRewardvodAd请求示例：

```

#import <ADMobGenSDK/ADMobGenRewardvodAd.h>

// 1、初始化激励视频广告对象，每次请求数据 需要重新创建
_rewardvodAd = [[ADMobGenRewardvodAd alloc] init];
_rewardvodAd.delegate = self;
_rewardvodAd.controller = self;

// 2、加载激励视频广告，播放一次之后需要重新拉取广告
[_rewardvodAd loadVideoAd];
// [_rewardvodAd loadVideoAdIndex:0];

```

//3、展示激励视频广告，可在广告数据加载成功回调中或者视频数据下载成功回调，强烈建议在广告数据加载成功回调中展示广告

```
- (void)admg_rewardvodAdLoadSuccessCallBack:(ADMobGenRewardvodAd
*)rewardvodAd{
[self show];
_rewardvodButton.selected = NO;
}
```

//4、展示激励视频广告时，判断广告是否过期或者广告是否准备好

```
- (void)showAd{
if (![ _rewardvodAd rewardvodAdIsValid]) {
NSLog(@"物料已过期，请重新加载");
return;
}
```

```
if ([ _rewardvodAd rewardvodAdIsReady]) {
//广告已经准备好了
[_rewardvodAd showRewardvodAd];
}
}
```

4.6 插屏广告 - ADMobGenInterstitialAd

插屏广告是移动广告的一种常见形式，在应用开流程中弹出，当应用展示插页式广告时，用户可以选择点按广告，访问其目标网址，也可以将其关闭，返回应用。

插屏广告 - ADMobGenInterstitialAd:

```
@interface ADMobGenInterstitialAd : NSObject

/**
 代理回调
*/
@property (nonatomic, weak) id<ADMobGenInterstitialAdDelegate> delegate;

- (id)new NS_UNAVAILABLE;
- (id)init NS_UNAVAILABLE;

/*
 初始化ADMobGenInterstitialAd
  @param viewController 开发者需传入用来弹出目标页的ViewController, 一般为当前
  ViewController
  @param index 请求的广告位序号
*/
- (instancetype)initWithViewController:(nullable UIViewController
*)viewController index:(NSInteger)index;

/*
 初始化ADMobGenInterstitialAd
  @param viewController 开发者需传入用来弹出目标页的ViewController, 一般为当前
  ViewController
  @param index 请求的广告位序号
  @param adSize 期望尺寸
  @param ratio 设置广告的大小, 广告比例(只影响穿山甲广告) 默认为1:1, size 300x300
*/
- (instancetype)initWithViewController:(nullable UIViewController
*)viewController index:(NSInteger)index adSize:(CGSize)adSize aspectRatio:
(ADMobGenInterstitialAdAspectRatio)ratio NS_DESIGNATED_INITIALIZER;

/**
 加载广告
*/
- (void)load;

/**
 展示广告
*/
- (void)show;
```

```
@end
```

插屏广告代理回调 - ADMobGenInterstitialAdDelegate

```
@protocol ADMobGenInterstitialAdDelegate <NSObject>

/**
    ADMobGenInterstitialAd请求成功回调

    @param interstitialAd 插屏广告实例对象
 */
- (void)admg_interstitialSuccedToLoad:(ADMobGenInterstitialAd *)interstitialAd;

/**
    ADMobGenInterstitialAd请求失败回调

    @param interstitialAd 插屏广告实例对象
    @param error 失败原因
 */
- (void)admg_interstitialFailedToLoad:(ADMobGenInterstitialAd *)interstitialAd
error:(NSError * _Nullable)error;

/**
    ADMobGenInterstitialAd展示在屏幕内回调

    @param interstitialAd 插屏广告实例对象
 */
- (void)admg_interstitialDidPresent:(ADMobGenInterstitialAd *)interstitialAd;

/**
    ADMobGenInterstitialAd点击回调

    @param interstitialAd 插屏广告实例对象
 */
- (void)admg_interstitialDidClick:(ADMobGenInterstitialAd *)interstitialAd;

/**
    ADMobGenInterstitialAd关闭回调

    @param interstitialAd 插屏广告实例对象
 */
- (void)admg_interstitialDidClose:(ADMobGenInterstitialAd *)interstitialAd;

/**
    ADMobGenInterstitialAd展示回调

    @param interstitialAd 广告实例
 */
```

```
*/  
- (void)admg_interstitialExposure:(ADMobGenInterstitialAd *)interstitialAd;  
  
@end
```

插屏广告 - ADMobGenInterstitialAd请求示例：

```
#import <ADMobGenSDK/ADMobGenInterstitialAd.h>  
  
// 1、插屏广告广告初始化  
self.interstitialAd = [[ADMobGenInterstitialAd  
alloc]initWithViewController:self index:0 adSize:CGSizeMake(kADMGScreenWidth,  
kADMGScreenHeight) aspectRatio:ADMobGenInterstitialAdAspectRatioDefault];  
self.interstitialAd.delegate = self;  
  
// 2、加载插屏广告  
[self.interstitialAd load];  
  
//3、推荐在admg_interstitialSuccedToLoad回调中，展示插屏广告  
/**  
    ADMobGenInterstitialAd请求成功回调  
  
    @param interstitialAd 插屏广告实例对象  
*/  
- (void)admg_interstitialSuccedToLoad:(ADMobGenInterstitialAd  
*)interstitialAd{  
    // 展示插屏广告  
    [self.interstitialAd show];  
}
```

4.7 全屏视频广告 - ADMobGenFullscreenVideoAd

全屏视频广告是移动广告的一种常见形式，在应用开流程中弹出，当应用展示视频式广告时，用户可以选择点按广告，访问其目标网址，也可以将其关闭，返回应用。

插屏广告 - ADMobGenFullscreenVideoAd：

```
@interface ADMobGenFullscreenVideoAd : NSObject

/**
 代理回调
*/
@property (nonatomic, weak) id<ADMobGenFullscreenVideoAdDelegate> delegate;

- (id)new NS_UNAVAILABLE;
- (id)init NS_UNAVAILABLE;

/**
 初始化ADMobGenFullscreenVideoAd
  @param viewController 开发者需传入用来弹出目标页的ViewController，一般为当前
  ViewController
  @param index 请求的广告位序号
*/
- (instancetype)initWithViewController:(UIViewController *)viewController
index:(NSInteger)index;

/**
 加载广告
*/
- (void)load;

/**
 展示广告
*/
- (void)show;

@end
```

全屏视频广告代理回调 - ADMobGenFullscreenVideoAdDelegate

```
@protocol ADMobGenFullscreenVideoAdDelegate <NSObject>

/**
  ADMobGenFullscreenVideoAd请求成功回调

  @param fullscreenVideoAd 全屏视频广告实例对象
*/
```

```

- (void)admg_fullscreenVideoSuccedToLoad:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd;

/**
    ADMobGenFullscreenVideoAd视频加载成功回调

    @param fullscreenVideoAd 全屏视频广告实例对象
*/
- (void)admg_fullscreenSuccessToLoadVideo:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd;

/**
    ADMobGenFullscreenVideoAd请求失败回调

    @param fullscreenVideoAd 全屏视频广告实例对象
    @param error 失败原因
*/
- (void)admg_fullscreenVideoFailedToLoad:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd error:(NSError *_Nullable)error;

/**
    ADMobGenFullscreenVideoAd展示在屏幕内回调

    @param fullscreenVideoAd 全屏视频广告实例对象
*/
- (void)admg_fullscreenVideoDidPresent:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd;

/**
    ADMobGenFullscreenVideoAd展示失败回调

    @param fullscreenVideoAd 全屏视频广告实例对象
    @param error 失败原因
*/
- (void)admg_fullscreenVideoFailToPresent:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd error:(NSError *)error;

/**
    ADMobGenFullscreenVideoAd点击回调

    @param fullscreenVideoAd 全屏视频广告实例对象
*/
- (void)admg_fullscreenVideoDidClick:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd;

/**
    ADMobGenFullscreenVideoAd关闭回调

    @param fullscreenVideoAd 全屏视频广告实例对象

```



```

*/
- (void)admg_fullscreenVideoDidClose:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd;

/**
    ADMobGenFullscreenVideoAd展示回调

    @param fullscreenVideoAd 广告实例
*/
- (void)admg_fullscreenVideoExposure:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd;

/**
    ADMobGenFullscreenVideoAd播放完成回调

    @param fullscreenVideoAd 广告实例
    @param error 失败原因（如果为nil，则是单纯的播放完成）
*/
- (void)admg_fullscreenVideoPlayComplete:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd didFailed:(NSError *_Nullable)error;

@end

```

全屏视频广告 - ADMobGenFullscreenVideoAd请求示例：

```

#import <ADMobGenSDK/ADMobGenFullscreenVideoAd.h>

// 1、全屏视频广告初始化
_fullscreenVideoAd = [[ADMobGenFullscreenVideoAd alloc]
initWithViewController:self index:0];
_fullscreenVideoAd.delegate = self;
// 2、加载全屏视频广告
[_fullscreenVideoAd load];

//3、推荐在admg_fullscreenVideoSuccedToLoad和admg_fullscreenSuccessToLoadVideo回
调中，展示插屏广告
/**
    ADMobGenFullscreenVideoAd请求成功回调

    @param fullscreenVideoAd 全屏视频广告实例对象
*/
- (void)admg_fullscreenVideoSuccedToLoad:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd {
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 *
NSEC_PER_SEC)), dispatch_get_main_queue(), ^{
        [fullscreenVideoAd show];
    });
}

```

```
/**
    ADMobGenFullscreenVideoAd视频加载成功回调

    @param fullscreenVideoAd 全屏视频广告实例对象
*/
- (void)admg_fullscreenSuccessToLoadVideo:(ADMobGenFullscreenVideoAd
*)fullscreenVideoAd {
    [fullscreenVideoAd show];
}
```

4.8 原生自渲染广告 - ADMobGenUnifiedNativeAd

自渲染是对原有类型的优化和升级，使用自渲染的API，您可以为您的应用打造定制式体验。

点击事件的响应需要调用unifiedNativeAdView类中ADMobUnifiedNativeAdViewProtocol协议中的admg_regist方法

原生自渲染广告 - ADMobGenUnifiedNativeAd：

```
#import <ADMobGenSDK/ADMobGenUnifiedNativeAd.h>

@interface ADMobGenUnifiedNativeAd : NSObject

- (id)new NS_UNAVAILABLE;
- (id)init NS_UNAVAILABLE;

/**
    ADMobGenUnifiedNativeAd初始化接口

    @param viewController 开发者需传入用来弹出目标页的ViewController，一般为当前
    ViewController
    @param index 广告位序号，默认为0
 */
- (instancetype)initWithViewController:(nullable UIViewController
*)viewController index:(NSInteger)index NS_DESIGNATED_INITIALIZER;

/**
    加载广告

    @param count 拉取几条广告,建议区间 1~4，超过可能无法拉取到
 */
- (void)load:(int)count;

/**
    代理回调
 */
@property (nonatomic, weak) id<ADMobGenUnifiedNativeAdDelegate> delegate;

@end
```

原生自渲染广告代理回调 - ADMobGenUnifiedNativeAdDelegate

```
@protocol ADMobGenUnifiedNativeAdDelegate <NSObject>

/**
    ADMobGenUnifiedNativeAd请求成功

    @param unifiedNativeAd 自渲染广告实例对象
```

```

@param nativeAdViewAry 自渲染广告视图数组
*/
- (void)admg_unifiedNativeSuccessToLoad:(ADMobGenUnifiedNativeAd
*)unifiedNativeAd viewArray:
(NSArray<UIView<ADMobUnifiedNativeAdViewProtocol>*> *)nativeAdViewAry;

/**
    ADMobGenUnifiedNativeAd请求失败

    @param unifiedNativeAd 自渲染广告实例对象
    @param error 失败原因
    */
- (void)admg_unifiedNativeFailToLoad:(ADMobGenUnifiedNativeAd
*)unifiedNativeAd error:(NSError *)error;

/**
    ADMobGenUnifiedNativeAd被点击

    @param unifiedNativeAdView 被点击的原生广告
    */
- (void)admg_unifiedNativeClicked:(ADMobGenUnifiedNativeAd *)unifiedNativeAd
view:(UIView<ADMobUnifiedNativeAdViewProtocol>*)unifiedNativeAdView;

/**
    ADMobGenUnifiedNativeAd展示

    @param unifiedNativeAd 广告实例
    @param unifiedNativeAdView 广告模板
    */
- (void)admg_unifiedNativeExposure:(ADMobGenUnifiedNativeAd *)unifiedNativeAd
view:(UIView<ADMobUnifiedNativeAdViewProtocol>*)unifiedNativeAdView;

@end

```

原生自渲染广告 - ADMobGenUnifiedNativeAd请求示例：

```

// 1、初始化信息流自渲染广告
_unifiedNativeAd = [[ADMobGenUnifiedNativeAd alloc]
initWithViewController:self index:0];
_unifiedNativeAd.delegate = self;
// 2、加载信息流自渲染广告
[self.unifiedNativeAd load:1];
.....
// 3、信息流自渲染广告加载成功
#pragma mark - ADMobGenUnifiedNativeAdDelegate
- (void)admg_unifiedNativeSuccessToLoad:(ADMobGenUnifiedNativeAd
*)unifiedNativeAd viewArray:(NSArray<UIView<ADMobUnifiedNativeAdViewProtocol>
*> *)nativeAdViewAry {

```

```

    for (UIView<AdMobUnifiedNativeAdViewProtocol>* nativeAdView in
nativeAdViewArray) {
        // 4、信息流自渲染广告视图由开发者渲染
        nativeAdView.frame = CGRectMake(0, 0, self.view.frame.size.width,
self.view.frame.size.width / 16.0 * 9 + 50 + 50);

        UIImageView *iconImageView = [UIImageView new];
        [nativeAdView addSubview:iconImageView];
        iconImageView.frame = CGRectMake(12, 3, 44, 44);
        NSString *iconUrlStr = nativeAdView.admg_dataObject.iconImageUrl;

        if(iconUrlStr) {

            dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND,
0), ^{
                UIImage *image = [UIImage imageWithData:[NSData
dataWithContentsOfURL:[NSURL URLWithString:iconUrlStr]]];
                dispatch_async(dispatch_get_main_queue(), ^{
                    iconImageView.image = image;
                });
            });
        }
        if(nativeAdView.admg_dataObject.iconImage)
            iconImageView.image = nativeAdView.admg_dataObject.iconImage;
        iconImageView.clipsToBounds = YES;
        iconImageView.layer.cornerRadius = 22;
        iconImageView.contentMode = UIViewContentModeScaleAspectFill;

        UILabel *titleLabel = [UILabel new];
        [nativeAdView addSubview:titleLabel];
        titleLabel.frame = CGRectMake(12 + 44 + 12, 0, 300, 50);
        titleLabel.text = nativeAdView.admg_dataObject.title;

        UIImageView *imageView = [UIImageView new];
        [nativeAdView addSubview:imageView];
        imageView.frame = CGRectMake(0, 50, self.view.frame.size.width,
self.view.frame.size.width / 16.0 * 9);
        NSString *urlStr = nativeAdView.admg_dataObject.imageUrl;

        if(urlStr.length > 0) {

            dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND,
0), ^{
                UIImage *image = [UIImage imageWithData:[NSData
dataWithContentsOfURL:[NSURL URLWithString:urlStr]]];
                dispatch_async(dispatch_get_main_queue(), ^{
                    imageView.image = image;
                });
            });
        }
    }
}

```

```

    }

    UIView *mediaView = [nativeAdView
    admg_mediaViewForWidth:imageView.frame.size.width];
    [nativeAdView addSubview:mediaView];
    mediaView.frame = imageView.frame;
    mediaView.hidden = !nativeAdView.admg_dataObject.shouldShowMediaView;

    UIImage *logoImage = [nativeAdView logoImageWithText:YES];
    UIImageView *logoImageView = [UIImageView new];
    [nativeAdView addSubview:logoImageView];
    logoImageView.frame = CGRectMake(0, self.view.frame.size.width / 16.0
    * 9 + 50, self.view.frame.size.width, 50);
    logoImageView.contentMode = UIViewContentModeBottomRight;
    logoImageView.image = logoImage;

    UILabel *descLabel = [UILabel new];
    descLabel.text = nativeAdView.admg_dataObject.desc;
    [nativeAdView addSubview:descLabel];
    descLabel.frame = CGRectMake(12, self.view.frame.size.width / 16.0 * 9
    + 50, self.view.frame.size.width, 50);
    // 重要
    [nativeAdView admg_regist];

    [self.items addObject:nativeAdView];
}
[self.tableView reloadData];
}
.....
// 5、信息流渲染广告内存管理，取消注册
for (UIView<ADMobUnifiedNativeAdViewProtocol> *item in _items) {
    [item admg_unregist];
}

```

问题排查

用户日志输出等级设置之后会在控制台打印各类信息，或者在广告请求的错误回调中也会带有各个类型的错误，错误代码说明如下：

错误代码	错误说明
1001	SDK初始化失败
1002	appId为空
1003	包名和appId不对应
1004	请求配置为空
-9995	请求超时（请检查网络或者是广告无填充，假设一直如此请问ADMobile媒介是否配置广告）
-9996	获取广告失败（可能是网络或者填充不够，属于正常情况）
-9997	未获取到加载广告的平台（第一次启动未获取到配置信息或者运营未配置广告）
-9998	未获取到需要展示的广告位ID（请问ADMobile媒介是否配置广告位id，和序号也有关系）
-9999	横屏不加载开屏广告

如碰到其他非常见问题，请查阅下表或者反馈问题至ADMob方技术人员

- Q：开屏广告有时候出现落地页无法关闭，或者落地页关闭之后广告视图不消失？

请检查是否在开屏广告的点击回调中将开屏广告的实例对象置空。开屏广告的实例对象置空会导致开屏广告的生命周期中断，开屏广告正常的生命周期为‘请求-展现-点击-落地页-关闭’，请在开屏广告的关闭回调中置空它。

- Q：请求开屏广告时在控制台出现请求失败的错误信息却没有执行错误回调

开屏广告目前设置超时时间为3s，请检查是否开屏广告的代理对象已经被置空了，导致没有执行错误回调

- Q：请求信息流模版广告时，点击之后落地页无法关闭

请检查是否是信息流广告的实例对象的控制器设置问题，保证和信息流展示的控制器是同一个

- Q：请求信息流模版广告时，广告无法点击

请检查是否调用信息流模版视图的contentSize方法

- Q：请求原生自渲染广告时，广告无法点击

请检查是否调用unifiedNativeAdView类中ADMobUnifiedNativeAdViewProtocol协议中的admg_regist方法

- Q: 请求信息流模版广告或者banner广告时, 出现信息流和banner显示不全或者文字会重复

请检查是否调用信息流模版视图的render方法, 已经调用的话可以参照demo中的WKWebView分类

- Q: pod导入有部分不是最新的库怎么办?

1、pod 删除记录命令sudo rm -fr ~/Library/Caches/CocoaPods/和rm
~/Library/Caches/CocoaPods/search_index.json

2、pod setup

3、pod中查找该SDK名称, 查看版本是否和文档中的一致

- Q: 展示ADMoblie广告落地页时, 导航栏按钮或者文字颜色有问题

ADMobileAdSDK.h文件中有属性barTintColor、barItemTintColor可以自定义, barItemTintColor这个是设置关闭文字颜色的, barTintColor这个是设置导航栏颜色的。

作者

sangshen@admobile.top, huacai@admobile.top

商务合作

tomato@admobile.top