

@[toc]

## 入门指南

本指南适用于希望借助 PTGAdSDK 通过 iOS 应用获利。

要展示广告和赚取收入，第一步是将移动广告 SDK 集成到应用中。集成 SDK 后，您可以选择一种广告样式，

### 前提条件

- 使用 Xcode 11.0 或更高版本
- 定位到 iOS 9.0 或更高版本
- 创建PTGAdSDK 帐号并注册应用'

### 导入 SDK

#### cocopods命令(推荐)

```
pod 'PTGAdFramework', '~> 1.3.9'
pod 'Bytedance-UnionAD', '3.3.6.1'
pod 'GDTMobSDK', '4.12.1'
```

#### 手动导入 SDK

直接下载并解压缩 SDK 框架，然后将以下框架导入您的 Xcode 项目中：

- PTGAdSDK.framework
  - PTGBundle.bundle
- 导入完成后，请确保 Build Phases > Copy Bundle Resources 中有

#### 需要添加其它 Framework

点击 Build Phases > Link Binary With Libraries，添加下面所需的 Framework:

- libz.tbd
- libbz2.tbd
- libxml2.tbd
- AdSupport.framework
- CoreMedia.framework
- CoreTelephony.framework
- CoreLocation.framework
- MediaPlayer.framework
- MobileCoreServices.framework
- CoreMotion.framework
- SystemConfiguration.framework
- WebKit.framework

头条还需要添加依赖库：

- libresolv.9.tbd
- MobileCoreServices.framework
- MediaPlayer.framework
- CoreMotion.framework
- Accelerate.framework
- libc++.tbd
- libsqlite3.tbd
- ImageIO.framework

## 全局配置

### 应用传输安全

应用传输安全(ATS) 是 iOS 9 中引入的隐私设置功能。默认情况下，系统会为新应用启用该功能，并强制实施安全连接。为确保您的广告不受 ATS 影响，请执行以下操作：

在您的应用的 Info.plist 文件中，添加 NSAllowsArbitraryLoads 以停用 ATS 限制。

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

### 关于 iOS 14 AppTrackingTransparency

iOS 14 开始，在应用调用 [App Tracking Transparency](#) framework 向用户请求应用跟踪权限之前，IDFA 将不可用。如果应用没有向用户发出请求，则 IDFA 将自动清零，这可能会导致广告收入出现重大损失。因为用户必须授权 IDFA。

本指南概述了支持 iOS 14 所需的更改。

### 请求 App Tracking Transparency 权限以访问 IDFA

应用开发者可以控制请求 App Tracking Transparency 框架来申请跟踪权限的时机。

要显示用于访问 IDFA 的 App Tracking Transparency 授权请求，请更新 Info.plist 以添加 NSUserTrackingUsageDescription 键，值为描述应用如何使用 IDFA 的自定义消息。以下是示例描述文字：

```
<key>NSUserTrackingUsageDescription</key>
<string>需要获取您设备的广告标识符，以为您提供更好的广告体验</string>
```

要显示授权请求对话框，请调用 [requestTrackingAuthorizationWithCompletionHandler](#)。我们建议您在授权回调之后再加载广告，以便如果用户允许跟踪权限，则 SDK 可以在广告请求中使用 IDFA，示例代码如下：

```

#import <AppTrackingTransparency/AppTrackingTransparency.h>
#import <AdSupport/AdSupport.h>

...

- (void)requestIDFA {
    [ATTrackingManager
requestTrackingAuthorizationWithCompletionHandler:^(ATTrackingManagerAuthorizat
ionStatus status) {
    // 授权结束，开始加载广告。注意：这不是主线程！
    // [self loadAd];
    }];
}

```

启用 SKAdNetwork 来跟踪转化

Apple 提供了 SKAdNetwork 用于进行转化跟踪，这意味着如果 Network SDK 支持 SKAdNetwork，那么即使 IDFA 不可用，也可以归因广告应用安装。

要启用此功能，您需要在 Info.plist 中添加 SKAdNetworkItems 键，并为各个 Network 添加对应的 SKAdNetworkIdentifier 键值对。为此，您需要查看下面的指南并更新

对于Bytedance-UnionA在3.2.5.0以上的版本需要：将 SKAdNetwork ID 添加到 info.plist 中，以保证 SKAdNetwork 的正确运行

```

<key>SKAdNetworkItems</key>
<array>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>238da6jt44.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>22mmun2rn5.skadnetwork</string>
    </dict>
</array>

```

权限请求窗口调用方法：requestTrackingAuthorization(completionHandler:)

## 添加 -ObjC

将 -ObjC 链接器标记添加到项目的 Build Settings 下的 Other Linker Flags 中：

▼ <b>Linking</b>	
Setting	PTGAdSDK_Example
Dynamic Library Install Name	
Dynamic Library Install Name Base	
Link With Standard Libraries	Yes ↕
Mach-O Type	Executable ↕
Other Librarian Flags	
> <b>Other Linker Flags</b>	<b>-ObjC -I"GDTMobSDK" -I"IFLYAdLib" -I"bz2" -I"c++"...</b>
Quote Linker Arguments	Yes ↕
Separately Edit Symbols	No ↕

## 集合SDK的初始化

加载广告之前，请先使用 PTGAdSDK 应用 ID 进行初始化，此操作仅需执行一次，最好是在应用启动时执行。

### 初始化

SDK 为接入方提供了开屏广告，开屏广告建议为用户在进入 App 时展示的全屏广告。开屏广告为一个 View，宽高默认为 match\_parent,注意开屏广告 view: width >=70%屏幕宽；height >=50%屏幕高，否则会影响计费。

```
@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    //Ptg后台创建的媒体广告位ID AppKey
    //Ptg后台创建的媒体广告位密钥 appSecret
    [PTGAdSDKManager setAppKey:@"45227" appSecret:@"1r8hOksXStGASHrp"
    success:^(BOOL result) {
        //初始化成功以后进行开屏页面的加载
    }];
    return YES;
}

@end
```

### 设置日志是否开启

```
[PTGAdSDKManager setLogEnable:false];
```

### 获取PTGAdSDK版本号

```
NSString *sdkVersion = [PTGAdSDKManager getSDKVersion];
```

现已导入移动广告 SDK 并完成了初始化，您可以实现广告了。PTGAdSDK 提供了多种广告样式，您可以从中选择最适合您应用的样式。

# 开屏广告

开屏广告用于当用户进入您的应用展示，一般会展示 5 秒（部分可以跳过），倒计时结束会自动关闭。与插屏广告不同，开屏广告可以轻松地展示应用品牌（Logo 和名称），以便用户了解他们看到广告的环境。

本指南向您介绍如何将 PTGAdSDK 开屏广告集成到 iOS 应用中。

## 前提条件

[导入 PTGAdSDK.framework]

## 创建开屏广告

开屏广告由 PTGAdSDK 对象来请求和展示。要使用此对象，第一步是对其实例化并设置其广告单元 ID。例如，以下示例演示了如何在 AppDelegate 的 application:didFinishLaunchingWithOptions: 方法中创建：

```
@interface AppDelegate () <PTGSplashAdDelegate>
@property (nonatomic, strong) PTGSplashA *splashAd;
@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {
    // Init TaurusXAds
    ...

    // Init SplashAd
    UIView *bottomView = [[UIView alloc] initWithFrame:CGRectMake(0, 0,
[[UIScreen mainScreen] bounds].size.width, 80)];

    UIImageView *logo = [[UIImageView alloc] initWithImage:[UIImage
imageName:@"SplashLogo"]];
    logo.accessibilityIdentifier = @"splash_logo";
    logo.frame = CGRectMake(0, 0, 311, 47);
    logo.center = bottomView.center;
    [bottomView addSubview:logo];
    self.nativeExpressAd = [[PTGSplashA alloc] initWithPlacementId:@"989"
bottomView:bottomView];
    self.nativeExpressAd.delegate = self;
    self.nativeExpressAd.hideSkipButton = false;
    [self.nativeExpressAd loadAd];

    return YES;
}
@end
```

## 底部区域设置

部分 Network 可以设置开屏广告底部展示的内容。

```
// 设置开屏底部显示的内容
UILabel *bottomView = [[UILabel alloc] initWithFrame:CGRectMake(0,0,[UIScreen
 mainScreen].bounds.size.width, 50)];
bottomView.backgroundColor = UIColor.whiteColor;
bottomView.text = @"App Name";
self.splashAd.bottomView = bottomView;
```

## 隐藏设置

如果你想控制开屏中控制跳过按钮的隐藏，可以在 SplashAd 对象上调用 hideSkipButton: 方法。如果不调用，默认显示。

## 展示广告

开屏广告加载完毕后会自动展示。

## 广告事件

通过使用 PTGSplashAdDelegate，您可以监听广告生命周期事件，例如，广告何时关闭、用户何时离开应用等。

## 注册开屏广告事件

要注册开屏广告事件，请将 PTGSplashA 上的 delegate 属性设置为实现 PTGSplashAdDelegate 协议的对象。一般情况下，实现开屏广告的类也充当代理类，在这种情况下，可将 delegate 属性设为 self，如下所示：

```
@interface AppDelegate () <PTGSplashAdDelegate>

@property (nonatomic, strong) PTGSplashA *splashAd;

@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    // Init 注册
    ...
    // Init SplashAd
    ...
    self.splashAd.delegate = self;

    return YES;
```

```
}

@end
```

## 实现开屏广告事件

PTGSplashAdDelegate 中的每个方法都是可选方法，因此您只需实现所需的方法即可。以下示例实现了每个方法并将消息记录到控制台：

```
- (void)splashAdSuccessPresentScreen:(NSObject *)splashAd {
    // 加载成功
    // 获取加载成功的 LineItem 信息
}

- (void)splashAdFailToPresent:(NSObject *)splashAd withError:(NSError *)error {
    // 加载失败，error 为错误信息
}

- (void)splashAdClicked:(NSObject *)splashAd{
    // 广告点击
}

- (void)splashAdClosed:(NSObject *)splashAd {
    // 广告关闭
    NSLog(@"txAdSplashDidDismissScreen");
}

- (void)splashAdDidCloseOtherController:(NSObject *)splashAd{
/**
 * 此方法在splash详情广告即将关闭时调用
 */
}
```

## 信息流广告和Draw信息流广告

信息流广告是通过平台原本就有的界面组件向用户呈现的广告素材资源。这种广告使用您在构建布局时已经采用的同类视图进行展示，而且能以和周围视觉设计相称的形式呈现，让用户有浑然一体的使用体验。从代码编写的角度来说，这意味着当信息流广告加载时，由应用负责展示它们了。

本指南向您介绍如何将信息流广告集成到 iOS 应用中。

### 创建广告

信息流广告和Draw信息流广告通过 PTGNativeExpressAd 类加载，要使用此对象，第一步是对其实例化并设置其广告单元 ID。例如，以下示例演示了如何在 UIViewController 的 viewDidLoad 方法中创建 PTGNativeExpressAd：

```
@import PTGNativeExpressAd;
```

```

#import UIKit;

@interface ViewController ()

@property(n nonatomic, strong) PTGNativeExpressAd *nativeExpressAd;

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    if (self.nativeExpressAd == nil) {
        // 信息流广告type=1; Draw信息流广告type=5;
        self.nativeExpressAd = [[PTGNativeExpressAd alloc]
initWithPlacementId:placementId type:5
adSize:CGSizeMake(self.view.frame.size.width , 0)];
        self.nativeExpressAd.delegate = self;
        [self.nativeExpressAd dataCorrectionHandler:^(BOOL result, NSArray *
_Nonnull views) {
            }];//数据成功回调 可以不用
        }
        [self.nativeExpressAd loadAdData];//数据加载
    }
}

@end

```

## 注意事项

建议高度给0,保证自适应布局

## 广告事件

通过使用 PTGNativeExpressAdDelegete, 您可以监听广告生命周期事件, 例如, 广告何时加载、用户何时离开应用等。

## 实现广告事件

PTGNativeExpressAdDelegete 中的每个方法都是可选方法, 因此您只需实现所需的方法即可。以下示例实现了每个方法, 并将消息记录到控制台:

```

- (void)nativeExpressAdSuccessToLoad:(NSObject *_Nullable)nativeExpressAd
views:(NSArray<__kindof UIView *> *_Nonnull)views{
    // 加载成功
    // 获取加载成功的 LineItem 信息
    //这里做渲染操作
    NSLog(@"%s",__FUNCTION__);
    self.expressAdViews = [NSMutableArray arrayWithArray:views];
    if (self.expressAdViews.count) {

```



```

        [self.expressAdViews enumerateObjectsUsingBlock:^(id _Nonnull obj,
        NSUInteger idx, BOOL * _Nonnull stop) {
            [self.nativeExpressAd render:obj controller:self];

        }];
    }
    [self.tableView reloadData];
}

- (void)nativeExpressAdFailToLoad:(NSObject * _Nonnull)nativeExpressAd error:
(NSError * _Nullable)error {
    // 加载失败, error 为错误信息
}

- (void)nativeExpressAdViewRenderSuccess:(UIView * _Nonnull)nativeExpressAdView{
    //这里做渲染成功
    //进行刷新

}

- (void)nativeExpressAdViewClicked:(UIView * _Nonnull)nativeExpressAdView{
    // 广告点击
    NSLog(@"nativeExpressAdViewClicked");
}

- (void)nativeExpressAdViewClosed:(UIView * _Nonnull)nativeExpressAdView {
    // 广告关闭
}

```

## 插屏广告

插屏广告是移动广告的一种常见形式，在应用流程中弹出，当应用展示插屏广告时，用户可以选择点击广告，访问其目标网址，也可以将其关闭并返回应用。在应用执行流程的自然停顿点，适合投放这类广告。

插屏广告 - PTGInterstitialAd

广告请求示例：

```

@interface PTGInterstitialAd : NSObject

/**
    代理回调
*/
@property (nonatomic, weak) id<PTGInterstitialAdDelegate> delegate;
/**
    Initializes interstitial ad.
    @param placementId : The unique identifier of interstitial ad.
    @param expectSize : custom size, default 600px * 400px
    @return PTGInterstitialAd
*/

```

```

- (instancetype)initWithPlacementId:(NSString *)placementId size:
(PTGProposalSize)expectSize;

/**
 Display interstitial ad.
 @param rootViewController : root view controller for displaying ad.
 */
- (void)showRootViewController:(UIViewController *)rootViewController;

/**
 Load interstitial ad datas.
 */
- (void)loadAdData;
@end

```

## 插屏广告代理回调 - PTGInterstitialAdDelegate

```

@protocol PTGInterstitialAdDelegate <NSObject>
@optional
/**
 This method is called when interstitial ad material loaded successfully.
 加载成功以后记得调取showRootViewController
 */
- (void)interstitialAdDidLoad:(NSObject *)interstitialAd;

/**
 This method is called when interstitial ad material failed to load.
 @param error : the reason of error
 */
- (void)interstitialAd:(NSObject *)interstitialAd didFailWithError:(NSError *
_Nullable)error;

/**
 This method is called when interstitial ad is clicked.
 */
- (void)interstitialAdDidClick:(NSObject *)interstitialAd;

/**
 This method is called when interstitial ad slot will be showing.
 */
- (void)interstitialAdWillVisible:(NSObject *)interstitialAd;

/**
 This method is called when interstitial ad is closed.
 */
- (void)interstitialAdDidClose:(NSObject *)interstitialAd;

```

```
@end
```

插屏代码示例：

```
PTGProposalSize尺寸:  
PTGProposalSize_Interstitial600_400,  
PTGProposalSize_Interstitial600_600,  
PTGProposalSize_Interstitial600_900,
```

```
#import <PTGAdSDK.h>  
- (void)loadAndShowWithBUProposalSize:(PTGProposalSize)proposalSize {  
    self.interstitialAd = [[PTGInterstitialAd alloc] initWithPlacementId:@"650"  
size:proposalSize];  
    self.interstitialAd.delegate = self;  
    [self.interstitialAd loadAdData];  
}  
  
#pragma mark - PTGInterstitialAdDelegate  
/**  
    IntertitialAd请求成功回调  
  
    @param interstitialAd 插屏广告实例对象  
    */  
- (void)interstitialAdDidLoad:(NSObject *)interstitialAd {  
    [self.interstitialAd showRootViewController:self.navigationController];  
}  
/**  
    interstitialAd请求失败回调  
  
    @param interstitialAd 插屏广告实例对象  
    @param error 失败原因  
    */  
- (void)interstitialAd:(NSObject *)interstitialAd didFailWithError:(NSError  
*)error {  
    //内存回收  
    NSLog(@"error code : %ld , error message : %@",  
(long)error.code,error.description);  
    _intertitialAd = nil;  
}  
/**  
    关闭回调  
    @param interstitialAd 插屏广告实例对象  
    */  
- (void)interstitialAdDidClose:(NSObject *)interstitialAd {
```

```
// 内存回收
_intertitialAd = nil;
}
```

## Banner横幅广告

Banner广告(横幅广告)位于app顶部、中部、底部任意一处，横向贯穿整个app页面；当用户与app互动时，Banner广告会停留在屏幕上，并可在一段时间后自动刷新。

横幅广告 - PTGBannerView:

```
@interface PTGBannerView : UIView

/**
 广告生命周期代理
 */
@property (nonatomic, weak) id<PTGExpressBannerViewDelegate> delegate;

/**
 * 构造方法
 * 详解: frame - banner 展示的位置和大小
 *        placementId - 广告位 ID
 *        viewController - 视图控制器
 */
- (instancetype)initWithFrame:(CGRect)frame
                        placementId:(NSString *)placementId
                        viewController:(UIViewController *)viewController;

/**
 * 拉取并展示广告
 */
- (void)loadAdData;

@end
```

横幅广告 - PTGExpressBannerViewDelegate

```
@protocol PTGExpressBannerViewDelegate <NSObject>

/**
  This method is called when bannerAdView ad slot loaded successfully.
  @param bannerAdView : view for bannerAdView
 */
- (void)nativeExpressBannerAdViewDidLoad:(UIView *)bannerAdView;

/**
  This method is called when bannerAdView ad slot failed to load.
  @param error : the reason of error
 */
```

```

    */
- (void)nativeExpressBannerAdView:(UIView *)bannerAdView didLoadFailWithError:
(NSError *_Nullable)error;

/**
    This method is called when rendering a nativeExpressAdView succeeded.
    */
- (void)nativeExpressBannerAdViewRenderSuccess:(UIView *)bannerAdView;

/**
    This method is called when bannerAdView is clicked.
    */
- (void)nativeExpressBannerAdViewDidClick:(UIView *)bannerAdView;

/**
    This method is called when closed.
    */
- (void)nativeExpressBannerAdViewDidClose:(UIView *)bannerAdView;

@end

```

请求横幅广告请求示例:

```

#import <PTGAdSDK.h>
@property (nonatomic, strong) PTGBannerView *bannerView;

- (PTGBannerView *)bannerView
{
    if (!_bannerView) {
        CGRect rect = {CGPointMake(0, 0), CGSizeMake(375, 60)};
        _bannerView = [[PTGBannerView alloc]
                        initWithFrame:rect
                        placementId:self.placementIdText.text.length > 0 ?
self.placementIdText.text: self.placementIdText.placeholder
                        viewController:self];
        _bannerView.accessibilityIdentifier = @"banner_ad";
        _bannerView.delegate = self;
    }
    return _bannerView;
}

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

```

```

}

// 代理回调
#pragma mark - PTGExpressBannerViewDelegate
/**
    广告获取成功

    @param bannerView banner实例
    */
- (void)nativeExpressBannerAdViewDidLoad:(UIView *)bannerAdView{

}

/**
    广告拉取失败

    @param bannerView banner实例
    @param errorModel 错误描述
    */
- (void)nativeExpressBannerAdView:(UIView *)bannerAdView didLoadFailWithError:
(NSError *_Nullable)error{
    NSLog(@"nativeExpressBannerAdView:%@, %@",errorModel.errorDescription,
errorModel.errorDetailDict);
    [_bannerView removeFromSuperview];
    _bannerView = nil;
}

/**
    广告点击

    @param bannerView 广告实例
    */
- (void)nativeExpressBannerAdViewDidClick:(UIView *)bannerAdView{

}

/**
    广告关闭

    @param bannerView 广告实例
    */
- (void)nativeExpressBannerAdViewDidClose:(UIView *)bannerAdView{
    _bannerView = nil;
}

/**
    广告展示

    @param bannerView 广告实例

```

```
 */
- (void)nativeExpressBannerAdViewRenderSuccess:(UIView *)bannerAdView{

}
```

## 激励视频广告

将短视频融入到APP场景当中，用户观看短视频广告后可以给予一些应用内奖励。常出现在游戏的复活、任务等位置，或者网服类APP的一些增值服务场景。

广告请求示例：

```
@interface PTGRewardedVideoAd : NSObject

/**
构造方法
@param placementId 广告位ID

/**
 *
 * 数据加载的时候使用
 *
 */
- (void)loadAdData;
*/
- (instancetype)initWithPlacementId:(NSString *)placementId;
/**
在合适的时间进行展示视频 （用户自己控制）
Display interstitial ad.
@param rootViewController : root view controller for displaying ad.
 */
- (void)showRootViewController:(UIViewController *)rootViewController;

@end
```

激励视频广告代理回调 - PTGRewardedVideoAdDelegate

```
@protocol PTGRewardedVideoAdDelegate <NSObject>

/**
广告数据加载成功回调

@param rewardedVideoAd NSObject 实例
 */
- (void)rewardVideoAdDidLoad:(NSObject *)rewardedVideoAd;
```

```

/**
// 视频数据下载成功回调，已经下载过的视频会直接回调
//
// @param rewardedVideoAd NSObject 实例
// */
- (void)rewardVideoAdVideoDidLoad:(NSObject *)rewardedVideoAd;

/**
    视频播放页即将展示回调

    @param rewardedVideoAd NSObject 实例
    */
- (void)rewardVideoAdWillVisible:(NSObject *)rewardedVideoAd;

/**
    视频播放页关闭回调

    @param rewardedVideoAd NSObject 实例
    */
- (void)rewardVideoAdDidClose:(NSObject *)rewardedVideoAd;

/**
    视频广告信息点击回调

    @param rewardedVideoAd NSObject 实例
    */
- (void)rewardVideoAdDidClicked:(NSObject *)rewardedVideoAd;

/**
    视频广告各种错误信息回调

    @param rewardedVideoAd NSObject 实例
    @param error 具体错误信息
    */
- (void)rewardVideoAd:(NSObject *)rewardedVideoAd didFailWithError:(NSError
*)error;

/**
    视频广告视频播放完成

    @param rewardedVideoAd NSObject 实例
    */
- (void)rewardVideoAdDidPlayFinish:(NSObject *)rewardedVideoAd;

@end

```

激励视频代码示例：



```

#import <PTGAdSDK.h>

- (void)loadRewardvodAd{
    // 1、初始化激励视频广告
    self.rewardVideoAd = [[PTGRewardedVideoAd alloc]
initWithPlacementId:placementId];
    self.rewardVideoAd.delegate = self;
    // 2、加载激励视频广告
    [self.rewardVideoAd loadAdData];
}

/**
    激励视频广告准备好被播放

    @param rewardvodAd 广告实例
    */
- (void)rewardVideoAdDidLoad:(NSObject *)rewardedVideoAd{
    //
    NSLog(@"%s",__FUNCTION__);

    self.statusLabel.text = [NSString stringWithFormat:@" 广告数据加载成功"];
    //在这里可以进行视频的加载 用户也可以在其他时机进行展示
}

/**
    视频播放页关闭回调

    @param rewardvodAd 广告实例
    */
- (void)rewardVideoAdDidClose:(NSObject *)rewardedVideoAd;

    // 4、广告内存回收
    rewardedVideoAd = nil;
}

/**
    视频广告请求失败回调

    @param rewardedVideoAd 广告实例
    @param error 具体错误信息
    */
- (void)rewardVideoAd:(NSObject *)rewardedVideoAd didFailWithError:(NSError
*)error;
*)errorModel{
    // 4、广告内存回收
    rewardedVideoAd = nil;
}

```

