

飞梭IOS-SDK接入文档

- 一、概述
- 二、SDK 项目部署
 - 2.1 开发环境
 - 2.2 集成方式
 - 方法一
 - 方法二
 - 2.3 注意事项
- 三、SDK 初始化及全局配置
 - 3.1 注意事项
 - 3.2 接口说明
 - 3.3 使用详情
- 四、开屏广告
 - 4.1 使用说明
 - 4.2 创建广告位对象、请求广告
 - 4.3 展示广告
 - 4.4 展示时机
 - 4.5 接收广告加载结果
- 五、插屏广告
 - 5.1 使用说明
 - 5.2 创建广告位对象、请求广告
 - 5.3 展示广告
 - 5.4 展示时机
 - 5.5 接收广告加载结果
 - 5.6 注意事项
- 六、信息流自渲染广告
 - 6.1 简介
 - 6.2 使用说明
 - 6.3 创建广告位对象、请求广告
 - 6.4 接收广告加载结果
 - 6.5 其他相关 API
 - 6.6 注意事项
- 七、信息流模板广告
 - 7.1 使用说明
 - 7.2 创建广告位对象、请求广告
 - 7.3 展示广告
 - 7.4 接收广告加载结果
- 八、激励视频广告
 - 8.1 使用说明
 - 8.2 创建广告位对象、请求广告
 - 8.3 展示广告
 - 8.4 展示时机
 - 8.5 接收广告加载结果

一、概述

FSUnionAdSDK 提供多种类型广告，该文档包含广告 SDK 接入的基础功能实现介绍。

二、SDK 项目部署

2.1 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：推荐Xcode 15.2及以上版本
- 开发语言：Swift5.0 & Objective-C
- 部署目标：iOS 11及以上版本
- 指令集架构：arm64 真机 和 arm64 / x86 模拟器

2.2 集成方式

方法一

使用 CocoaPods 方式集成

在 podfile 文件中加入以下代码即可接入成功。

```
pod 'FSUnionAdSDK'
```

备注：外部接入时，暂时不支持此方式

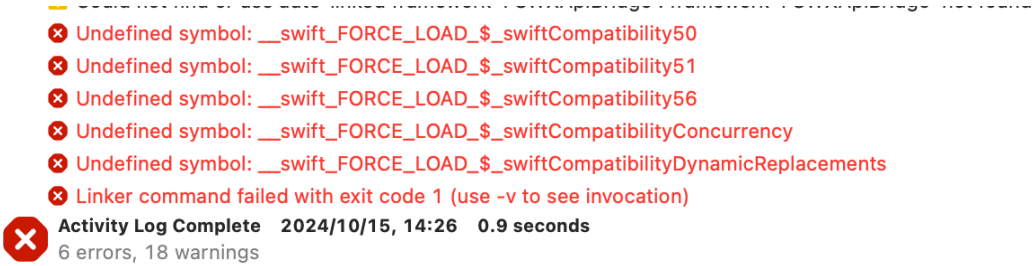
方法二

- 1. framework 文件: FSUnionAdSDK.framework
- 2. 资源文件: FSUnionAdSDK.bundle

将 framework 文件和资源文件直接拖入工程导入即可

2.3 注意事项

1. 使用手动方式集成时，由于 FSUnionAdSDK是使用 Swift 语言开发的项目，所以你的项目如果是一个纯 OC，没有 Swift 文件 的工程，你的项目在引入FSUnionAdSDK.framework 可能后会报错如下：



解决办法：在您的项目中新建一个新的空 Swift 文件即可

2. 使用手动集成时，添加framework 后，需要确保 Xcode工程 Build Settings 配置文件中，other link flags 标志中包含 -ObjC



三、 SDK 初始化及全局配置

3.1 注意事项

- FSAdSDKConfiguration 可以设置 SDK 的一些全局信息，提供单例进行设置和读取；
- FSAdSDKConfiguration 中 appId 为必填项，否则初始化无法成功，其他参数为选填；
- FSAdSDKManager 类 提供SDK 初始化，通过 FSAdSDKConfiguration 所设置的信息进行初始化；

3.2 接口说明

目前接口提供一下几个方法：

```

public class FSAdSDKConfiguration: NSObject {
    ///
    public static let configuration = FSAdSDKConfiguration()

    public var debug: Bool = false

    /// ID
    public var appID: String?

    ///
    public var appName: String?

    /// ID
    public var customDeviceID: String?

    ///
    public var extraUserData: [String: Any]?

    /// false
    public var personalRecommend: Bool = false

    /// false
    public var teenagerModel = false

    /// "AppStore"
    public var channel: String = "AppStore"

    ///
    public let version: String = "1.0.0.0"

    /// build number
    public let buildNumber: Int8 = 0

    /// false
    public var supportWXApi: Bool = false

    private override init() {}
}

public class FSAdSDKManager: NSObject {
    /// SDK
    public static func startWithCompletionHandler(_ completionHandler: ((Bool)->Void)?)
}

```

3.3 使用详情

建议在 **AppDelegate** 中的 **didFinishLaunchingWithOptions** 回调中初始化 SDK ；

传入正确的 **AppID** 后，**SDK** 会立刻初始化成功回调，**SDK** 内部各个模块的初始化会在子线程异步进行。

Swift版本

```

import UIKit
/// FSUnionAdSDK
import FSUnionAdSDK

@main
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.

        // FSAdSDKManager.logEnable(true)

        let configuration = FSAdSDKConfiguration.configuration
        configuration.appID = "xxx"
        FSAdSDKManager.startWithCompletionHandler { success in
            if (!success) {
                print("appID")
            }
        }

        return true
    }
}

```

OC 版本

```

#import "AppDelegate.h"
/// FSUnionAdSDK
#import <FSUnionAdSDK/FSUnionAdSDK-Swift.h>
// @import FSUnionAdSDK;

@implementation AppDelegate

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

    // [FSAdSDKManager logEnable:YES];

    FSAdSDKConfiguration *configuration = FSAdSDKConfiguration.configuration;
    configuration.appID = @"XXX";
    [FSAdSDKManager startWithCompletionHandler:^(BOOL success) {
        if (!success) {
            NSLog(@"appID");
        }
    }];

    return YES;
}

@end

```

四、开屏广告

4.1 使用说明

开屏使用 **FSSplashAD**对象调用**loadAd()**请求广告，使用**FSSplashAD**对象调用**showSplashWithRootController:bottomView:**展示广告，通过设置**FSSplashAdDelegate**代理，获取广告、展示、点击、关闭等回调。

4.2 创建广告位对象、请求广告

FSSplashAd

请求广告时必须传入广告位对象

字段定义	是否必传	字段名称	字段类型	备注
slotID	是	代码位	String	代码位ID
type	是	广告类型	Enum: Int	FSAdTypeSplash 表示开屏
timeout	否	超时时间	Double	
requestID	否	请求 ID		

使用FSSplashAd创建广告对象，使用FSSplashAd调用loadAd() 请求广告

```
// FSAdSlot
FSAdSlot *slot = [[FSAdSlot alloc] initWithSlotID:@"XXX" type:FSAdTypeSplash];
slot.timeout = 5.0;

// FSSplashAD
FSSplashAD *ad = [[FSSplashAD alloc] initWithSlot:slot];
// delegate
ad.delegate = self;

//
[ad loadAd];

// ad
self.splashAd = ad;
```

4.3 展示广告

调用showSplash:bottomView:展示广告，此处需要传入当前展示的面，作为rootViewController，供展示广告和跳转落地页使用。

bottomView 为可选参数

```
[self.splashAd showSplashIn:UIApplication.sharedApplication.keyWindow.rootViewController bottomView:self.bottomView];
```

4.4 展示时机

广告请求成功后，即可展示广告

```
- (void)fs_splashAdLoadSuccess:(FSSplashAd *)ad {
    [self.splashAd showSplashIn:UIApplication.sharedApplication.keyWindow.rootViewController bottomView:self.bottomView];
}
```

但是建议在收到物料加载成功后再展示广告

```
- (void)fs_splashAdMaterialDownloadSuccess:(FSSplashAd *)ad {
    [self.splashAd showSplashIn:UIApplication.sharedApplication.keyWindow.rootViewController bottomView:self.bottomView];
}
```

4.5 接收广告加载结果

FSSplashAdDelegate回调说明

回调方法	注释	备注
<code>fs_splashAdLoadSuccess(_ ad: FSSplashAd)</code>	开屏广告请求成功回调	请求成功后，可以通过 <code>ad.price</code> 获取价格
<code>fs_splashAdLoadFailed(_ ad: FSSplashAd, error: NSError)</code>	开屏广告请求失败回调	
<code>fs_splashAdMaterialDownloadSuccess(_ ad: FSSplashAd)</code>	开屏广告物料加载成功回调	
<code>fs_splashAdMaterialDownloadFailed(_ ad: FSSplashAd, error: NSError)</code>	开屏广告物料加载失败回调	
<code>fs_splashAdWillPresent(_ ad: FSSplashAd)</code>	开屏广告将要展示回调	
<code>fs_splashAdDidVisible(_ ad: FSSplashAd)</code>	开屏广告曝光回调	
<code>fs_splashAdFailedToPresent(_ ad: FSSplashAd, error: NSError)</code>	开屏广告展示失败回调	
<code>fs_splashAdDidClick(_ ad: FSSplashAd)</code>	开屏广告点击回调	
<code>fs_splashAdDidClose(_ ad: FSSplashAd, closeType: FSSplashAdCloseType)</code>	开屏广告关闭回调	<pre>enum FSSplashAdCloseType: Int { case unknown // 未知 case clickSkip // 点击跳过 case countdownToZero // 倒计时结束 case clickJump // 点击跳转 }</pre>

五、插屏广告

5.1 使用说明

插屏使用 `FSInterstitialAd` 对象调用 `loadAdData()` 请求广告，使用 `FSInterstitialAd` 对象调用 `showAdFromRootViewController(:)` 展示广告，通过设置 `FSInterstitialAdDelegate` 代理，获取广告、展示、点击、关闭等回调。

5.2 创建广告位对象、请求广告

`FSInterstitialAd`

请求广告时必须传入广告位ID

字段定义	是否必传	字段名称	字段类型	备注
<code>slotID</code>	是	代码位	String	代码位ID
<code>type</code>	是	广告类型	Enum: Int	<code>FSAdTypeInterstitial</code> 表示插屏

使用 `FSInterstitialAd` 创建广告对象，使用 `FSInterstitialAd` 调用 `loadAdData()` 请求广告

```
// FSAdSlot
FSAdSlot *slot = [[FSAdSlot alloc] initWithSlotID:@"XXX" type:FSAdTypeInterstitial];
slot.timeout = 5.0;

// FSInterstitialAd
FSInterstitialAd *ad = [[FSInterstitialAd alloc] initWithSlot:slot];
//
ad.videoMuted = YES;
// delegate
ad.delegate = self;

//
[ad loadAdData];

// ad
self.ad = ad;
```

5.3 展示广告

调用`showAdFromViewController(:)`展示广告，此处需要传入当前展示的页面，作为`rootViewController`，供展示广告和跳转落地页使用。

```
ad.showAdFromRootViewController(self)
```

5.4 展示时机

广告请求成功后，即可展示广告（**SDK 内部有 bug，不支持此时展示，有概率展示不出来广告**）

```
func fs_interstitialAdSuccessToLoadAd(_ ad: FSInterstitialAd) {
    ad.showAdFromRootViewController(self)
}
```

但是建议在收到物料加载成功后再展示广告

```
func fs_interstitialAdSuccessToDwonlaodMaterial(_ ad: FSInterstitialAd) {
    ad.showAdFromRootViewController(self)
}
```

5.5 接收广告加载结果

FSInterstitialAdDelegate回调说明

回调方法	注释	备注
<code>fs_interstitialAdSuccessToLoadAd(_ ad: FSInterstitialAd)</code>	广告请求成功回调	请求成功后，可以通过 <code>ad.price</code> 获取价格
<code>fs_interstitialAdFailedToLoadAd(_ ad: FSInterstitialAd, error: Error)</code>	广告请求失败回调	
<code>fs_interstitialAdSuccessToDwonlaodMaterial(_ ad: FSInterstitialAd)</code>	广告物料加载成功回调	
<code>fs_interstitialAdFailedToDwonlaodMaterial(_ ad: FSInterstitialAd, error: Error)</code>	广告物料加载失败回调	
<code>fs_interstitialAdRenderSuccess(_ ad: FSInterstitialAd)</code>	渲染成功回调	
<code>fs_interstitialAdRenderFail(_ ad: FSInterstitialAd, error: Error)</code>	渲染失败回调	
<code>fs_interstitialAdDidVisible(_ ad: FSInterstitialAd)</code>	广告曝光回调	
<code>fs_interstitialAdDidClick(_ ad: FSInterstitialAd)</code>	广告点击回调	
<code>fs_interstitialAdDidClose(_ ad: FSInterstitialAd)</code>	广告关闭回调	1.0.1.7 版本后 插屏广告不再自动关闭
<code>fs_interstitialAdDidCloseOtherController(_ ad: FSInterstitialAd, interactionType: FSAdInteractionType)</code>	此回调在广告跳转到其他控制器时，该控制器被关闭时调用 <code>interactionType</code> ：此参数可区分是打开的appstore/网页等	

5.6 注意事项

- 必须要设置`rootViewController`，用来处理广告跳转。插屏广告的跳转采用`present`的方式，请确保传入的`rootViewController`不能为空且没有`present`其他的控制器，否则会出现`presentedViewController`已经存在而导致`present`失败。

六、信息流自渲染广告

6.1 简介

自渲染是对原有类型的优化和升级，通过自渲染的API，您可自行定义广告布局样式和展示场景

6.2 使用说明

自渲染广告使用**FSNativeAdsManager**对象调用**loadAdData**：请求广告，通过设置**FSNativeAdsManagerDelegate**、**FSNativeAdDelegate**代理，获取广告相关素材，以及点击、曝光等回调。

6.3 创建广告位对象、请求广告

通过 **FSNativeAdsManager** 对象进项广告数据请求

```
FSAdSlot *slot = [[FSAdSlot alloc] initWithSlotID:self.slotTextField.text type:FSAdTypeNative];
FSNativeAdsManager *loadManager = [[FSNativeAdsManager alloc] initWithSlot:slot];
loadManager.delegate = self;
[loadManager loadAd];
self.loadManager = loadManager;
```

通过**FSNativeAdsManagerDelegate** 获取，广告请求成功 & 失败回调

```
- (void)fs_nativeAdsManagerDidFail:(FSNativeAdsManager * _Nonnull)adsManager error:(NSError * _Nullable)
error {
    NSLog(@"%s",__FUNCTION__);
}

- (void)fs_nativeAdsManagerSuccessToLoad:(FSNativeAdsManager * _Nonnull)adsManager nativeAds:
(NSArray<FSNativeAd *> * _Nonnull)nativeAds {
    if (nativeAds.count <= 0) {
        return;
    }
    self.nativeAd = nativeAds.firstObject;
}
```

6.4 接收广告加载结果

FSNativeAdsManagerDelegate 回调说明

回调方法	注释	备注
fs_nativeAdsManagerSuccessToLoad(_:nativeAds:)	加载成功回调	请求成功后，可以通过 ad.price获取价格
fs_nativeAdsManagerDidFail(_:error:)	加载失败回调	

FSNativeAdDelegate 回调说明

回调方法	注释	
fs_nativeAdDidBecomeVisible (nativeAd:)	广告显示回调	

fs_nativeAdDidCloseOtherController(nativeAd:interactionType:)	此回调在广告跳转到其他控制器时，该控制器被关闭时调用 interactionType: 此参数可区分是打开的appstore/网页/视频广告详情页面	<pre>@objc public enum FSAdInteractionType: Int { case WfNativeAdTypeUnspecified = 0 ///未定义 case WfNativeAdTypeRedirect = 1 ///落地页 case WfNativeAdTypeDownload = 2 ///下载 case WfNativeAdTypeDeepLink = 3 ///拉起deeplink case WfNativeAdTypeMiniApp = 4 ///小程序 case WfNativeAdTypeMarket = 5 ///应用市场 }</pre>
fs_nativeAdDidClick(nativeAd:containerView:)	广告点击回调	

6.5 其他相关 API

通过FSNativeRelatedView，可以直接获取和使用播放器、logo 等相关视图

FSNativeAdRelatedView 说明

api	注释	备注
<code>public var logoImageView: UIImageView = UIImageView()</code>	logo 图片视图	
<code>public var videoAdView: FSVideoAdView?</code>	视频广告对应的播放器	
<code>public func refreshData(_ nativeAd:FSNativeAd)</code>	刷新数据源	注意：在使用FSNativeRelatedView 内的属性前，必须先调用 refreshData 方法

如果希望监听videoAdView 播放器的回调，需要设置delegate: FSVideoAdViewDelegate

```
self.nativeAdRelatedView.delegate = self;
```

FSVideoAdViewDelegate 回调说明

回调	注释	备注
<code>func fs_videoAdViewReadyToPlay(_ videoAdView: FSVideoAdView)</code>	播放器加载完成，准备播放	
<code>func fs_videoAdViewDidPlayFinish(_ videoAdView: FSVideoAdView)</code>		
<code>func fs_videoAdView(_ videoAdView: FSVideoAdView, didLoadFailedWithError error: NSError?)</code>		
<code>func fs_videoAdView(_ videoAdView: FSVideoAdView, playStateDidChange playState: FSPlayerPlayState)</code>		<pre>@objc public enum FSPlayerPlayState: Int { case none = 0 case buffering case playing case stopped case pause case failed }</pre>

6.6 注意事项

- 1、在物料加载成功方法里获取相关广告信息赋值后，需调用 `registerContainer:withClickableViews:clickableViews`注册绑定点击的View并刷新数据源`refreshData:`。
- 2、每次获取物料信息后需要刷新调用`refreshData:`方法
- 3、如果期望获取播放视频宽高可以使用：`FSNative.material.videoWidth & .videoHeight`，此属性在`fs_videoAdViewReadyToPlay(_ videoAdView: FSVideoAdView)` 回调之后生效，否则为 0

七、信息流模板广告

7.1 使用说明

信息流模板广告使用`FSNativeExpressFeedsAd`对象调用`loadAdData()`请求广告，使用`FSNativeExpressFeedsAd`对象调用`showInView:`展示广告，通过设置`FSNativeExpressFeedsAdDelegate`代理，获取广告、展示、点击、关闭等回调。

注意：信息流模板广告分两种类型，一种是大图模板 (`FSNativeExpressType.banner`)，一种是全屏模板 (`FSNativeExpressType.brand`)，两者的唯一区别在于 UI 样式上的差异，可通过`FSNativeExpressFeedsAd`的`setExpressType()`方法进行设置，默认是 `banner` 大图模板类型。

7.2 创建广告位对象、请求广告

FSNativeExpressFeedsAd

请求广告时必须传入广告位ID

字段定义	是否必传	字段名称	字段类型	备注
slotID	是	代码位	String	代码位ID

使用`FSNativeExpressFeedsAd`创建广告对象，调用`loadAdData()`请求广告

```
FSNativeExpressFeedsAd *ad = [[FSNativeExpressFeedBannerAd alloc] initWithSlotID:slotID];
ad.delegate = self;
ad.rootViewController = self; // SDK rootViewController
[ad loadAdData];

self.feedBannerAd = ad;
```

7.3 展示广告

调用`showInView:`展示广告，该方法必须在`fs_expressFeedAdLoadSuccess(_ ad:FSNativeExpressFeedsAd)`回调后调用

```
[self.feedBannerAd showInView:self.adCell.contentView];
```

7.4 接收广告加载结果

FSNativeExpressFeedsAdDelegate回调说明

回调方法	注释	备注
<code>fs_expressFeedAdLoadSuccess(_ ad:FSNativeExpressFeedsAd)</code>	广告资源加载成功回调	<ul style="list-style-type: none">▪ 此方法回调后才可以调用广告的 <code>show</code> 方法，否则无法正常展示▪ 请求成功后，可以通过 <code>ad.price()</code> 获取价格单位分
<code>fs_expressFeedAdLoadFailed(_ ad: FSNativeExpressFeedBannerAd, withError error: NSError)</code>	广告请求失败回调	

fs_expressFeedAdShowSuccess(_ ad: FSNativeExpressFeedBannerAd)	广告展示成功回调	
fs_expressFeedAdShowFailed(_ ad: FSNativeExpressFeedBannerAd, withError error: NSError)	广告展示失败回调	
fs_expressFeedAdDidClosed(_ ad: FSNativeExpressFeedBannerAd)	广告关闭回调	
fs_expressFeedAdDidClicked(_ ad: FSNativeExpressFeedBannerAd)	广告点击回调	

八、激励视频广告

8.1 使用说明

激励视频广告使用FSrewardedAd对象调用loadAdData()请求广告，使用FSrewardedAd对象调用showAdFromRootViewController()展示广告，通过设置FSRewardedAdDelegate代理，获取广告、展示、点击、关闭等回调。

8.2 创建广告位对象、请求广告

FSRewardedAd

请求广告时必须传入广告位ID

字段定义	是否必传	字段名称	字段类型	备注
slotID	是	代码位	String	代码位ID
type	是	广告类型	Enum: Int	FSAdTypeReward 表示激励视频

使用FSRewardedAd创建广告对象，使用FSInterstitialAd调用loadAdData()请求广告

```
// FSAdSlot
FSAdSlot *slot = [[FSAdSlot alloc] initWithSlotID:self.textField.text type:FSAdTypeReward];
// FSRewardedAd
FSRewardedAd *rewardedAd = [[FSRewardedAd alloc] initWithSlot:slot];
// delegate
rewardedAd.delegate = self;

//
[rewardedAd loadAdData];

// ad
self.rewardedAd = rewardedAd;
```

8.3 展示广告

调用showAdFromRootViewController()展示广告，此处需要传入当前展示的页面，作为rootViewController，供展示广告和跳转落地页使用。

```
ad.showAdFromRootViewController(self)
```

8.4 展示时机

广告请求成功后，即可展示广告

```
- (void)fs_rewardedAdDidLoadSuccess:(FSRewardedAd *)ad {
    [self.rewardedAd showAdFromRootViewController:self];
}
```

但是建议在收到物料加载成功后再展示广告

```

- (void)fs_rewardedAdDidDownloadMaterial:(FSRewardedAd *)ad {
    [self.rewardedAd showAdFromRootViewController:self];
}

```

8.5 接收广告加载结果

FSRewardedAdDelegate回调说明

回调方法	注释	备注
@objc optional func fs_rewardedAdDidLoadSuccess(_ ad: FSRewardedAd)	广告请求成功回调	请求成功后，可以通过 ad.price获取价格
@objc optional func fs_rewardedAd(_ ad: FSRewardedAd, didLoadFailedWithError error: Error)	广告请求失败回调	
@objc optional func fs_rewardedAdDidDownloadMaterial(_ ad:FSRewardedAd)	广告物料加载成功回调	
@objc optional func fs_rewardedAd(_ ad:FSRewardedAd, didRenderFailedWithError error: Error)	广告物料加载失败回调	
@objc optional func fs_rewardedAdWillPresent(_ ad: FSRewardedAd)	广告即将展示回调	
@objc optional func fs_rewardedAdDidVisible(_ ad: FSRewardedAd)	广告曝光回调	
@objc optional func fs_rewardedAdDidSucceed(_ ad: FSRewardedAd)	广告奖励发放成功回调	
@objc optional func fs_rewardedAdDidClose(_ ad: FSRewardedAd)	广告关闭回调	
@objc optional func fs_rewardedAdDidClick(_ ad: FSRewardedAd)	广告点击回调	
@objc optional func fs_rewardedAdDidSkip(_ ad: FSRewardedAd)	广告点击跳过回调	
@objc optional func fs_rewardedAdVideoDidPlayFinish(_ ad: FSRewardedAd)	视频广告播放完成	