## 四翼鸟广告投放平台 SDK iOS集成文档

## 1. 概述

本集成文档旨在帮助 iOS 开发者快速、便捷地完成 FBirdAdSDK 的接入，涵盖 SDK 导入、权限配置、广告请求与展示、点击/曝光事件上报、常见问题等方面的内容。

---

## 2. 前提条件

- Xcode 12.0 及以上

- iOS 11.0 及以上

- 使用 Objective-C / Swift 项目

- 已获取 FBirdAdSDK.framework 或已添加到项目的 CocoaPods 源

---

## 3. SDK 导入

### 3.1 手动导入

1. 将 `FBirdAdSDK.framework`,`FBirdAdSDKBundle.bundle` 拖入 Xcode 项目。

2. 在 Target 的 \*Build Phases\* 中确认已添加至 \*Link Binary With Libraries\*。

---

## 4. Info.plist 权限配置

为保证 SDK 的正常功能，需要在 `Info.plist` 中添加以下配置：

```xml

<key>NSUserTrackingUsageDescription</key>

<string>我们需要使用广告标识符以提供更精准的广告服务</string>

```

---

## 5. 初始化 SDK

无需额外初始化流程，直接调用 API 即可。

如有些敏感参数不获取，可以设置

```objective-c

[FBirdAdSDKManager setCanUseIDFAState:YES]; //是否获取IDFA 建议获取，如不获取，影响广告填充

[FBirdAdSDKManager setCanUseIDFVState:NO]; //是否获取IDFV

[FBirdAdSDKManager setCanUseLimitPersonalAdsState:NO]; //是否个性化推荐

```

---

## 6. 请求广告

### 6.1 基础请求

```objective-c

#import <FBirdAdSDK/FBirdAdSDK.h>

- (void)requestAd {

[FBirdAdSDKManager loadAdViewsWithTagID:tagID caid:@"" adtype:adType requestCount:1 completion:^(NSArray<FBirdAdSDKView \*> \* \_Nullable adViews, NSArray<NSError \*> \* \_Nullable errors) {

dispatch\_async(dispatch\_get\_main\_queue(), ^{

if (adViews.count > 0) {

self.loadedAdViews = adViews;

self.currentAdIndex = 0;

[self showAdViewAtIndex:self.currentAdIndex]; //广告展示绑定回调

} else {

NSLog(@"⚠️ 没有加载到广告");

}

if (errors.count > 0) {

NSLog(@"⚠️ 部分广告加载失败: %@", errors);

}

});

}];

}

```

### 6.2 广告类型

```objective-c

typedef NS\_ENUM(NSInteger, FBirdAdType) {

Splash = 1, ///< 开屏广告

Interstitial = 2, ///< 插屏广告

Feed = 4, ///< 信息流广告

RewardVideo = 7 ///< 激励视频广告

};

```

```objective-c

测试ID：开屏 “5”

插屏 “1631”

信息流 “128”

激励视频 “1346”

```

---

## 7. 广告展示及绑定回调

```objective-c

(void)showAdViewAtIndex:(NSInteger)index {

if (index < 0 || index >= self.loadedAdViews.count) return;

// 移除旧广告视图

for (UIView \*subview in self.adContainerView.subviews) {

[subview removeFromSuperview];

}

FBirdAdSDKView \*view = self.loadedAdViews[index];

view.frame = self.adContainerView.bounds;

view.userActionCallback = ^(FBirdAdUserActionType actionType) {

if ([actionType isEqualToString:FBirdAdUserActionTypeSwipeUp]) {

NSLog(@"🔼 用户上滑广告");

} else if ([actionType isEqualToString:FBirdAdUserActionTypeShake]) {

NSLog(@"🤳 用户晃动设备");

} else if ([actionType isEqualToString:FBirdAdUserActionTypeSkip]) {

NSLog(@"⏩ 用户跳过广告");

for (UIView \*subview in self.adContainerView.subviews) {

[subview removeFromSuperview];

}

} else if ([actionType isEqualToString:FBirdAdUserActionTypeClose]) {

NSLog(@"❎ 用户关闭广告");

} else if ([actionType isEqualToString:FBirdAdUserActionTypeConfirmExit]) {

NSLog(@"🚪 用户坚持退出广告");

} else if ([actionType isEqualToString:FBirdAdUserActionTypeClickDetail]) {

NSLog(@"📌 用户点击详情");

}

};

// 展示广告视图到容器上

[view showToView:self.adContainerView];

}

```

## 8. 数据模型参考

```objective-c

#pragma mark - 视频模型

@interface VideoContent : NSObject

@property (nonatomic, copy) NSString \*url;

@property (nonatomic, copy) NSString \*cover;

@property (nonatomic, assign) NSInteger duration;

@end

#pragma mark - 图片模型

@interface NativeImage : NSObject

@property (nonatomic, assign) NSInteger h;

@property (nonatomic, assign) NSInteger w;

@property (nonatomic, assign) NSInteger type;

@property (nonatomic, copy) NSString \*url;

@end

#pragma mark - 标题模型

@interface NativeTitle : NSObject

@property (nonatomic, copy) NSString \*text;

@end

#pragma mark - 数据模型

@interface NativeData : NSObject

@property (nonatomic, copy) NSString \*value;

@end

#pragma mark - 资源单元

@interface NativeAsset : NSObject

@property (nonatomic, assign) NSInteger assetId;

@property (nonatomic, assign) BOOL isrequired;

@property (nonatomic, strong) NativeImage \*img;

@property (nonatomic, strong) NativeTitle \*title;

@property (nonatomic, strong) VideoContent \*video;

@property (nonatomic, strong) NativeData \*data;

@end

#pragma mark - 原生广告内容

@interface NativeContent : NSObject

@property (nonatomic, strong) NSArray<NativeAsset \*> \*assets;

@end

#pragma mark - 广告对象

@interface AdmObject : NSObject

@property (nonatomic, strong) VideoContent \*video;

@property (nonatomic, strong) NativeContent \*native;

@end

#pragma mark - 事件跟踪

@interface TrackingEvents : NSObject

@property (nonatomic, strong) NSArray<NSString \*> \*imp\_urls;

@property (nonatomic, strong) NSArray<NSString \*> \*click\_urls;

@property (nonatomic, strong) NSArray<NSString \*> \*deeplink\_urls;

@end

#pragma mark - 竞价单元

@interface Bid : NSObject

@property (nonatomic, copy) NSString \*target;

@property (nonatomic, copy) NSString \*deeplink;

@property (nonatomic, copy) NSString \*action;

@property (nonatomic, copy) NSString \*bidId;

@property (nonatomic, copy) NSString \*crid; // 修改为NSString类型

@property (nonatomic, assign) NSInteger price;

@property (nonatomic, copy) NSString \*impid;

@property (nonatomic, strong) AdmObject \*admobject;

@property (nonatomic, copy) NSString \*adid;

@property (nonatomic, copy) NSString \*nurl;

@property (nonatomic, strong) TrackingEvents \*events;

@end

#pragma mark - 席位竞价

@interface SeatBid : NSObject

@property (nonatomic, strong) NSArray<Bid \*> \*bid;

@end

#pragma mark - 完整响应

@interface AdResponseModel : NSObject

@property (nonatomic, copy) NSString \*responseId;

@property (nonatomic, copy) NSString \*bidid;

@property (nonatomic, strong) SeatBid \*seatbid;

- (instancetype)initWithDictionary:(NSDictionary \*)dict;

@end

#pragma mark - 请求视频模型

@interface VideoInfo : NSObject

@property (nonatomic, assign) NSInteger width;

@property (nonatomic, assign) NSInteger height;

@property (nonatomic, assign) NSInteger type;

@property (nonatomic, strong) NSArray<NSString \*> \*mimes;

- (NSDictionary \*)toDictionary;

@end

```

---

## 10. 常见问题

- \*\*广告无返回\*\*：检查网络状态及 `tagID` 是否正确。

- \*\*点击/曝光无回调\*\*：确认 `view` 是否可交互，`clickPoint` 是否在 `view` 范围内。

- \*\*权限弹窗不弹出\*\*：检查 `Info.plist` 中权限配置是否完整。

---