

# 云蜻广告 iOS SDK 接入说明

文档版本	修订日期	修订说明
v1.0	2018-3-23	创建文档，支持Banner，信息流广告，开屏广告
v1.1	2018-5-21	增加超时处理，超过30秒钟后会返回失败。
v2.0	2018-6-30	增加广告位icon
v2.1	2018-7-17	增加广告渠道，SDK中配置开屏显示逻辑
v2.2	2018-8-28	优化超时请求处理.优化开屏展示逻辑
v2.3	2018-9-2	增加广告内容，优化展示
v2.4	2018-9-25	【1】请求超时时间默认5秒 【2】去掉showEnterForeground 属性，此属性不再使用 【3】 demo中附带 icon 请求示例，启动页再次请求示例 【4】所有的广告windows不再使用keywindow 【5】 第三方framework更新
v2.5	2018-10-04	【1】 增加banner广告 【2】 优化开屏广告
v2.8	2018-11-10	【1】 增加广告点击跳转微信小程序功能。 【2】 微信开放平台新增了微信模块用户统计功能 【3】 增加原生广告。
v2.9	2018-12-16	【1】 优化轮播banner展示 【2】 添加pageController，默认显示
v3.0	2019-02-19	【1】 增加更多尺寸的banner显示 【2】 增加自定义大小的原生尺寸广告数据
v3.0.3	2019-03-19	增加多Icon样式。
v4.0.0	2019-04-12	【1】 优化轮播广告展示，增强用户自定义体验 【2】 增加数据安全性。
v4.0.1	2019-04-15	修复 BUG

v4.0.2	2019-04-16	更新 SDK 的轮播控件，增加自定义属性
v4.1.0	2019-04-16	修复 轮播图不能点击的 BUG
v4.1.1	2019-04-16	更新 Demo 稳定对接版本
v4.1.2	2019-04-18	增加插屏广告
v4.1.3	2019-04-24	修复Safari 浏览器不能关闭的问题
v4.2.0	2019-05-01	去掉UIWebView，替换为WKWebView，去掉微信SDK及其相关功能
v4.3.0	2019-07-21	集成广点通，穿山甲，快手等广告主，增加原生模板广告
v4.4.0	2020-03-03	增加激励视频， 增加Draw视频流广告
v4.5.0	2021-08-25	增加贴片广告，增加H5金币任务

## 云靖广告 iOS SDK 接入说明

### 1. 展示广告接入

#### 准备工作

#### 1.1 申请应用的媒体位ID

#### 1.2 iOS\_SDK导入framework

##### 方法一：直接拖入动态库

##### 1.2.1 工程设置导入framework

##### 1.2.2 Xcode编译选项设置

##### 1.2.2.1 添加权限

##### 1.2.2.2 运行环境配置

##### 1.2.2.3 添加依赖库

##### 方法二：使用CocoaPods

#### 1.3 SDK接口类介绍与广告接入

##### 1.3.0 全局设置

##### 1.3.1 开屏广告

##### 1.3.2 原生信息流广告

##### 1.3.3 icon广告

##### 1.3.4 banner广告

##### 1.3.5 轮播广告

##### 1.3.6 插屏广告

##### 1.3.7 原生模板广告

##### 1.3.8 激励视频

##### 1.3.9 draw视频流广告

1.3.10 视频贴片广告

2. H5 商城接入

2.1 准备工作

2.2 导入SDK包

2.3 权限申请

附录

错误码

FAQ

## 1. 展示广告接入

---

### 准备工作

#### 1.1 申请应用的媒体位ID

1. 申请账号：开发者从SDK运营人员处获取账号、密码后，登录SDK系统后台。
2. 媒体位id：开发者从SDK运营人员处获取对应的媒体位id。

#### 1.2 iOS\_SDK导入framework

##### 方法一：直接拖入动态库

##### 1.2.1 工程设置导入framework

获取 framework 文件后直接将 {YXLaunchAD}文件拖入工程即可。

拖入时请按以下方式选择：

拖入完请确保Copy Bundle Resources中有BUAdSDK.bundle,XibAndPng.bundle，否则可能出现icon图片加载不出来的情况。

##### 1.2.2 Xcode编译选项设置

##### 1.2.2.1 添加权限

注意要添加的系统库

- 工程plist文件设置，点击右边的information Property List后边的 "+" 展开

添加 App Transport Security Settings，先点击左侧展开箭头，再点右侧加号，Allow Arbitrary Loads 选项自动加入，修改值为 YES。SDK API 已经全部支持HTTPS，但是广告主素材存在非HTTPS情况。

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

具体操作如图：

- Build Settings中Other Linker Flags 增加参数-ObjC

具体操作如图：

- 在项目设置中，选择Build Phases,点击左上角+号，添加Embed Frameworks和Run Script,如图所示：

- 工程plist文件设置，点击右边的information Property List后边的 "+" 展开 更新您的 Info.plist

添加 App Transport Security Settings，先点击左侧展开箭头，再点右侧加号，Allow Arbitrary Loads 选项自动加入，修改值为 YES。SDK API 已经全部支持HTTPS，但是广告主素材存在非HTTPS情况。

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

具体操作如图：

- SDK中包含获取IDFA的权限,所以需要在info.plist中添加IDFA权限,如图所示：

```
<key>NSUserTrackingUsageDescription</key>
<string>该标识符将用于向您投放个性化广告</string>
```

- 将SDK的 SKAdNetwork ID 添加到 info.plist 中，以保证 SKAdNetwork 的正确运行,如图所示：

```
<key>SKAdNetworkItems</key>
<array>
  <dict>
```

```
<key>SKAdNetworkIdentifier</key>
<string>r3y5dwb26t.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>238da6jt44.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>x2jnk71y8j.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>22mmun2rn5.skadnetwork</string>
</dict>
  <dict>
    <key>SKAdNetworkIdentifier</key>
    <string>f7s53z58qe.skadnetwork</string>
  </dict>
</array>
```

### 1.2.2.2 运行环境配置

- 支持系统 iOS 9.0 及以上;
- SDK编译环境 Xcode 9.3;
- 支持架构: i386, x86-64, armv7, arm64

### 1.2.2.3 添加依赖库

工程需要在TARGETS -> Build Phases中找到Link Binary With Libraries, 点击“+”, 依次添加下列依赖库

- libresolv.9.tbd
- libc++.tbd
- libc++abi.tbd
- libz.tbd
- libbz2.tbd
- libxml2.tbd
- libiconv.tbd
- libsqlite3.tbd
- StoreKit.framework
- MobileCoreServices.framework
- WebKit.framework
- MediaPlayer.framework
- CoreMedia.framework
- AVFoundation.framework

- CoreLocation.framework
- CoreTelephony.framework
- SystemConfiguration.framework
- AdSupport.framework
- CoreMotion.framework
- Security.framework
- QuartzCore.framework
- CoreGraphics.framework
- SafariServices.framework
- UIKit.framework
- Foundation.framework
- JavaScriptCore.framework
- MapKit.framework
- AssetsLibrary.framework
- AppTrackingTransparency.framework

具体操作如图所示：

## 方法二：使用CocoaPods

SDK3.0版本以后支持pod方式接入，只需配置pod环境，在podfile文件中加入以下代码即可接入成功。不用在添加任何依赖库。

```
# 建议pod到最新版本 当前最新版本为4.5.2
pod 'YXLaunchAD' , '~> 4.5.2'
```

更多关于pod方式的接入请参考 [github地址](#)

## 1.3 SDK接口类介绍与广告接入

### 1.3.0 全局设置

SDK的开屏广告建议在 AppDelegate 的方法 `-(BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions` 里

当window指定root控制器之后最先进行初始化

### 1.3.1 开屏广告

- **类型说明：**开屏广告主要是 APP 启动时展示的全屏广告视图，开发只要按照接入标准就能够展示设计好的视图**(注意：开屏接入代码必须放在window指定rootViewController之后)**。具体可参考Demo中YXLaunchScreenViewController部分示例代码

#### 1. 导入

```
#import <YXLaunchAds/YXLaunchAdapterManager.h>
```

## 2. 遵循代理

<YXLaunchAdManagerDelegate>

3. 广告页面呈现在一个不是keywindow的windows上，建议开屏广告的初始化放在window指定root控制器之后。

- 全屏示例：

```
YXLaunchAdManager *adManager = [YXLaunchAdManager new];
adManager.mediaId = @"运营分配的媒体位";
adManager.delegate = self;
adManager.showBackImage = YES;
[adManager loadLaunchAdWithShowAdwindow:[UIApplication
sharedApplication].delegate.window];
```

- 非全屏示例：

```
//含 APP Logo 的bottomView
UIView *bottom = [[UIView alloc] initWithFrame:CGRectMake(0, [UIScreen
 mainScreen].bounds.size.height * 0.8, [UIScreen mainScreen].bounds.size.width,
 [UIScreen mainScreen].bounds.size.height * 0.2)];
bottom.backgroundColor = [UIColor whiteColor];
UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage
 imageNamed:@"APP的Logo"]];
image.center = CGPointMake(bottom.bounds.size.width/2,
 bottom.bounds.size.height/2);
[bottom addSubview:image];

YXLaunchAdManager *adManager = [YXLaunchAdManager new];
adManager.mediaId = @"运营分配的媒体位";
adManager.delegate = self;
adManager.showBackImage = YES;
adManager.bottomView = bottom;
[adManager loadLaunchAdWithShowAdwindow:[UIApplication sharedApplication]
keywindow]]];
```

建议等待时间设置为5秒，展示时间设置为5秒。App在从后台5分钟后到前台时 建议也加上开屏广告。

### 1.3.2 原生信息流广告

- **类型说明：** 广告原生广告即一般广告样式，形式分为图文和视频。
- **使用说明：** 在SDK里只需要使用 YXFeedAdManager 就可以获取原生广告，YXFeedAdManager 类提供了原生广告的数据类型等各种信息，在数据获取后可以在属性 data (YXFeedAdData) 里面获取广告数据信息。具体可参考Demo中YXFeedAdViewController部分示例代码。

### 1.3.3 icon广告

- **类型说明：** Icon广告主要是 APP 中展示一个小图标，用户点击可跳到对应的广告业或者小程序。

- **使用说明：** SDK可提供单Icon与多Icon样式。具体可参考Demo中YXIconViewController部分示例代码。

### 1.3.4 banner广告

- **类型说明：** 原生banner广告是为满足媒体多元化需求而开发的一种广告。
- **使用说明：** SDK可提供数据绑定、点击事件的上报、响应回调，开发者进行自渲染，接入方式同原生广告相同。具体可参考Demo中YXBannerViewController部分示例代码

### 1.3.5 轮播广告

- **类型说明：** 原生轮播广告主要是在APP 中展示的广告轮播视图，开发只要按照接入标准就能够展示设计好的视图。具体可参考Demo中YXScrollerBannerViewController部分示例代码

### 1.3.6 插屏广告

- **类型说明：** 原生插屏广告是为满足媒体多元化需求而开发的一种广告。
- **使用说明：** SDK可提供数据绑定、点击事件的上报、响应回调，开发者进行自渲染，接入方式同原生广告相同。具体可参考Demo中YXInterstitialViewController部分示例代码

### 1.3.7 原生模板广告

- **类型说明：** 模板广告即视图广告,SDK会返回已渲染完成的广告视图,开发只需展示即可,避免了接入方的大量工作量。
- **使用说明：** 在SDK里只需要使用 SFNativeExpressAdManager 就可以获取模板广告，SFNativeExpressAdManager 类提供了模板广告的各种信息，具体可参考Demo中YXNativeExpressAdController的部分示例代码。

### 1.3.8 激励视频

- **类型说明：** 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。具体可参考Demo中YXMotivationVideoViewController部分示例代码。

#### 1. 导入

```
#import <YXLaunchAds/YXMotivationVideoManager.h>
```

#### 2. 遵循代理

```
<YXMotivationDelegate>
```

#### 3. 使用示例



```
self.motivationVideo = [YXMotivationVideoManager new];
self.motivationVideo.delegate = self;
self.motivationVideo.showAdController = self;
self.motivationVideo.mediaId = @"beta_ios_video";//使用申请得到的媒体位
[self.motivationVideo loadVideoPlacement];
```

### 1.3.9 draw视频流广告

- **类型说明：**draw视频流广告是为满足媒体视频流多元化需求而开发的一种广告。
- **使用说明：**SDK可提供数据绑定、点击事件的上报、响应回调，开发者进行自渲染，接入方式同原生广告类似。具体可参考Demo中YXDrawVideoViewController部分示例代码

### 1.3.10 视频贴片广告

- **类型说明：**视频贴片广告是为满足媒体视频流多元化需求而开发的一种广告。
- **使用说明：**SDK提供数据视图。具体可参考Demo中 YXPasterVideoViewController 部分示例代码

## 2. H5 商城接入

### 2.1 准备工作

申请账号：开发者从SDK后台运营人员处获取Channel ID。

在 AppDelegate 文件中的 - (BOOL)application:(UIApplication \*)application  
didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions 方法下初始化渠道ID的赋值：  
[YXAdSDKManager defaultManager].channelID = @"您的渠道ID,比如beta-ios";

在用户登录后,为vuid赋值:

[YXAdSDKManager defaultManager].vuid = @"前缀+用户的userID(比如1),vuid即为beta\_1,建议对  
userID加密,类似beta\_c4ca4238a0b923820dcc509a6f75849b";

在用户退出登录后,为vuid赋值为空字符串:

[YXAdSDKManager defaultManager].vuid = @"";

### 2.2 导入SDK包

获取 framework 文件后直接将 {YXLaunchAD.framework、XibAndPng.bundle}文件拖入工程即可。

拖入时请按以下方式选择：

拖入完请确保Copy Bundle Resources中有XibAndPng.bundle，否则可能出现icon图片加载不出来的情况。

## 2.3 权限申请

依赖广告权限IDFA

获取入口素材代码示例：

```
- (void)getH5TaskWebViewWithPostID:(NSString *)postID {
    [YXAdSDKManager getMaterialWithPostID:postID Success:^(id _Nonnull json)
    {
        if (json) {
            dispatch_async(dispatch_get_main_queue(), ^{
                NSLog(@"接口请求成功 data=%@", json);
                NSDictionary *dataDict = (NSDictionary *)json;
                NSDictionary *dict = dataDict[@"data"];
                if (dict[@"material_path"]) {
                    //活动入口 标题与logo
                    //展示到页面
                }
            });
        }
    } fail:^(NSError * _Nonnull error) {
        NSLog(@"接口请求失败 error=%@", error);
    }];
}
```

调用任务活动

```
[YXAdSDKManager defaultManager].channelID = @"您的渠道ID,比如beta-ios";
[YXAdSDKManager defaultManager].vuid = @"前缀+用户的userID(比如1),vuid即为beta_1,
建议对userID加密,类似beta_c4ca4238a0b923820dcc509a6f75849b";
//调用素材接口获取入口素材icon以及title -> 详情参考demo
SFTaskWebViewController *taskVC = [SFTaskWebViewController new];
taskVC.urlStr = @"在素材接口中获取的page_url";
taskVC.posId = @"入口位置标记, 如: banner-v1";
[self.navigationController pushViewController:taskVC animated:YES];
```

## 附录

### 错误码

下面是各种ErrorCode的值

```
ErrorCode      = 404,      // 网络请求失败
ErrorCode      = 403,      // 解析的数据没有广告
ErrorCode      = 402,      // 解析失败
ErrorCode      = 401,      // 请求配置失败
ErrorCode      = 10001,    // 参数错误
ErrorCode      = 10002,
ErrorCode      = 20000,
ErrorCode      = 20001,    // 没有广告
ErrorCode      = 40000,    // http content_type错误
ErrorCode      = 40001,    // http request pb错误
ErrorCode      = 40002,    // 请求app不能为空
ErrorCode      = 40003,    // 请求wap不能为空
ErrorCode      = 40004,    // 缺少广告位描述
ErrorCode      = 40005,    // 广告位尺寸 不合法
ErrorCode      = 40006,    // 广告位 ID 不合法
ErrorCode      = 40007,    // 请求广告数量 错误
ErrorCode      = 50001     // 广告服务器错误
服务器错误码
ErrorCode      = 40008 //没有填写素材尺寸，或者素材尺寸大于 10000
ErrorCode      = 40009 //媒体是空，或者没有运行
ErrorCode      = 40015 //如果字段非法,则不返回广告
ErrorCode      = 40016 //请求的 appid 与媒体平台的 appid 不一致
ErrorCode      = 40018 //SDK包名与广告配置包名不一致

ErrorCode      = 205001 //后台数据错误
ErrorCode      = 205002 //视频素材下载错误
ErrorCode      = 205003 //视频素材播放错误
ErrorCode      = 205004 //没匹配的广告，禁止重试，否则影响流量变现效果
ErrorCode      = 205005 //广告请求量或者消耗等超过日限额，请第二天再请求广告
ErrorCode      = 205006 //包名校验非法
ErrorCode      = 205009 //广告请求量或者消耗等超过小时限额，请一小时后再请求广告
ErrorCode      = 205010 //广告样式校验失败，请检查广告位与接口使用是否一致
ErrorCode      = 205012 //广告过期，请重新拉取
ErrorCode      = 205013 //广告拉取过于频繁，请稍后再试
```

## FAQ

1. 为什么demo可以运行，接入项目会出错？

答：接入SDK需要很多的配置工作，请按照文档说明配置齐全，保证没有遗漏！

2. 媒体位、内容位、渠道号等在哪获取？

答：在我们的对接群里，请联系我们的运营人员获取所需要的媒体位、内容位或者渠道号等！

### 3. 广告对接成功，但是没有收益是怎么回事？

答：广告收益一般在第二天会在后台系统显示，节假日顺延，如果还是没看到，请确保媒体位是否使用正确，如果误用测试媒体位，这个是没有广告收益的！

### 3. iOS集成的包大小是多少？

答： 根据我们demo打包后的计算为5M左右。但是具体大小会根据导入的功能有所差别。实际情况以集成后的包大小为主。