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

云蜻广告 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. 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>x2jnk7ly8j.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

- MapKit.framework
- AssetsLibrary.framework
- JavaScriptCore.framework
- libresolv.9.tbd
- libc++.tbd
- libz.tbd
- libbz2.tbd

- libxml2.tbd
- libiconv.tbd

  如果仍然报错，可继续添加以下依赖库，默认以下依赖库系统已自动添加！

  - 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4.5.0
pod 'YXLaunchAD' , '~> 4.5.0'
```

更多关于pod方式的接入请参考 gitthub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/YXLaunchAdManager.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.waitDataDuration = 5;
adManager.duration = 5;
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：

```
YXLaunchAdManager *adManager = [YXLaunchAdManager new];
adManager.waitDataDuration = 5;
adManager.duration = 5;
adManager.mediaId = @"运营分配的媒体位";
adManager.delegate = self;
adManager.showBackImage = YES;
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];
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. 资讯内容接入

2.1 准备工作

2.1.1 申请内容接入用户ID和对应的内容位ID

1. 申请账号：开发者从云蜻SDK后台运营人员处获取用户、密码后，登录云蜻内容运营后台。
2. 接入用户 ID 以及内容位ID： 开发者每创建一个应用后，系统会自动生成用户ID和内容位ID，可在云蜻SDK后台界面查看到已创建的应用以及对应的用户ID和内容位ID。

2.1.2 导入framework

获取 framework 文件后直接将 {YXLaunchAD.framework、XibAndPng.bundle}文件拖入工程即可。此 SDK 依赖第三方 MJRefresh与Weichat SDK,若工程已有，请勿重复导入

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。

2.2 全屏接入

```
SFInformationViewController *infoVC = [SFInformationViewController new];
infoVC.mediaId = @"1234";     //账号ID
infoVC.mLocationId = @"34";   //媒体内容位 ID
[self.navigationController pushViewController:infoVC animated:YES];
```

2.3 半屏接入

2.3.1 包含 tableView 半屏接入以及 scrollView 半屏接入，详情参考 Demo

新建自定义ScrollVIew或tableView继承自系统的UIScrollView或 UITableView，遵守代理，实现代理方法，让其允许多手势操作

```
- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
shouldRecognizeSimultaneouslyWithGestureRecognizer:(UIGestureRecognizer
*)otherGestureRecognizer
{
return YES;
}
```

2.3.2 主控制器操作

让自己的主控器以自定义 ScrollVIew 或 tableView 为底，遵守代理 ，实现代理方法，在viewDidLoad中添加监听，在dealloc中移除监听。 懒加载SFHalfPageViewController，让当前主控器添加子控制器，创建属性canScroll来控制ScrollView 的滑动，详情参考 Demo

a. scrollView 的 Demo

> **(isShowAllChannels： YES -> 所有频道 ; NO -> 只有一个推荐频道)**

```
- (void)viewDidLoad {
    [super viewDidLoad];
    self.canScroll = YES;
    // Do any additional setup after loading the view.
    self.scrollView = [[SFScrollerView alloc] initWithFrame:CGRectMake(0, 0,
self.view.bounds.size.width, self.view.bounds.size.height)];
    self.scrollView.contentSize = CGSizeMake(0,
self.view.bounds.size.height+300);
    self.scrollView.delegate = self;
    [self.view addSubview:self.scrollView];

    UIView *headView = [[UIView alloc] initWithFrame:CGRectMake(0, 0,
self.view.bounds.size.width, 300)];
    headView.backgroundColor = [UIColor purpleColor];
    [self.scrollView addSubview:headView];

    UIView *footView = [[UIView alloc] initWithFrame:CGRectMake(0, 300,
self.view.bounds.size.width, self.view.bounds.size.height)];
    footView.backgroundColor = [UIColor cyanColor];
    [self.scrollView addSubview:footView];
```

```
    [self addChildViewController:self.webVC];
    [footView addSubview:self.webVC.view];

    //添加请求数据的 HUD 开始请求推荐数据
    [self.webVC refreshNewsData];

    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(changeScrollStatus) name:LEAVETOPNOTIFITION object:nil];

}
- (void)changeScrollStatus//改变主视图的状态
{
    self.canScroll = YES;
    self.webVC.vcCanScroll = NO;
}
- (SFHalfPageViewController *)webVC{
    if (_webVC == nil) {
        _webVC = [[SFHalfPageViewController alloc] init];
        _webVC.mediaId = @"4";
        _webVC.mLocationId = @"3";
        _webVC.vcCanScroll = NO;
        _webVC.halfDelegate = self;
        _webVC.isShowAllChannels = self.isShowAll;
    }
    return _webVC;
}
- (void)scrollViewDidScroll:(UIScrollView *)scrollView{
    CGFloat offset = scrollView.contentOffset.y;
    CGFloat bottomCellOffset = 300 - StatusBarAndNavigationBarHeight;
    if (offset >= bottomCellOffset) {
        scrollView.contentOffset = CGPointMake(0, bottomCellOffset);
        if (self.canScroll) {
            self.canScroll = NO;
            self.webVC.vcCanScroll = YES;
        }
    }else{
        if (!self.canScroll) {//子视图没到顶部
            scrollView.contentOffset = CGPointMake(0, bottomCellOffset);
        }
    }
    self.scrollView.showsVerticalScrollIndicator = _canScroll?YES:NO;
}
#pragma mark - SFPageViewControllerDelegate
- (void)newsDataRefreshSuccess{
    NSLog(@"数据加载成功");
}
- (void)newsDataRefreshFail:(NSError *)error{
```

```
    NSLog(@"数据加载失败，error = %@",error);
}

- (void)dealloc{
    [[NSNotificationCenter defaultCenter] removeObserver:self];
    NSLog(@"%@ %@",[self class],NSStringFromSelector(_cmd));
}
```

b. tableView 的 Demo

> **(isShowAllChannels： YES -> 所有频道 ; NO -> 只有一个推荐频道)**

```
- (void)viewDidLoad {
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor whiteColor];
    self.canScroll = YES;
    // Do any additional setup after loading the view.
    [self addChildViewController:self.webVC];
    [self.view addSubview:self.tableView];

    //添加请求数据的 HUD 开始请求推荐数据
    [self.webVC refreshNewsData];

    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(changeScrollStatus) name:LEAVETOPNOTIFITION object:nil];

}
- (void)changeScrollStatus//改变主视图的状态
{
    self.canScroll = YES;
    self.webVC.vcCanScroll = NO;
}
- (SFHalfPageViewController *)webVC{
    if (_webVC == nil) {
        _webVC = [[SFHalfPageViewController alloc] init];
        _webVC.mediaId = @"4";
        _webVC.mLocationId = @"3";
        _webVC.vcCanScroll = NO;
        _webVC.halfDelegate = self;
        _webVC.isShowAllChannels = self.isShowAll;
    }
    return _webVC;
}
- (void)scrollViewDidScroll:(UIScrollView *)scrollView{
    CGFloat offset = scrollView.contentOffset.y;
    CGFloat bottomCellOffset = 300 - StatusBarAndNavigationBarHeight;
    if (offset >= bottomCellOffset) {
        self.tableView.contentOffset = CGPointMake(0, bottomCellOffset);
        if (self.canScroll) {
```

```
            self.canScroll = NO;
            self.webVC.vcCanScroll = YES;
        }
    }else{
        if (!self.canScroll) {//子视图没到顶部
            scrollView.contentOffset = CGPointMake(0, bottomCellOffset);
        }
    }
    self.tableView.showsVerticalScrollIndicator = _canScroll?YES:NO;
}
- (UIView *)tableViewHeaderView{
    UIView *headView = [[UIView alloc] initWithFrame:CGRectMake(0, 0,
self.view.bounds.size.width, 300)];
    UILabel *label = [[UILabel alloc] initWithFrame:headView.bounds];
    label.text = @"这是header~~~~~~~~~~~~~~";
    label.textAlignment = NSTextAlignmentCenter;
    label.textColor = [UIColor redColor];
    [headView addSubview:label];
    headView.backgroundColor = [UIColor greenColor];
    return headView;
}

- (SFTableView *)tableView{
    if(!_tableView){
        _tableView = [[SFTableView alloc] initWithFrame:CGRectMake(0, 0,
self.view.bounds.size.width, self.view.bounds.size.height)
style:UITableViewStylePlain];
        _tableView.delegate = self;
        _tableView.dataSource = self;
        _tableView.rowHeight = self.view.bounds.size.height;
        _tableView.tableHeaderView = [self tableViewHeaderView];
        [_tableView registerNib:[UINib nibWithNibName:@"YXFeedAdTableViewCell"
bundle:nil] forCellReuseIdentifier:@"YXFeedAdTableViewCell"];
    }
    return _tableView;
}
#pragma mark - tableViewDelegate
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView{
    return 1;
}
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
(NSInteger)section{
    return 1;
}
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
(NSIndexPath *)indexPath {
    YXFeedAdTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"YXFeedAdTableViewCell"
forIndexPath:indexPath];
```

```
    [cell.subviews makeObjectsPerformSelector:@selector(removeFromSuperview)];
    self.webVC.view.frame = CGRectMake(0, 0, self.tableView.bounds.size.width,
self.tableView.bounds.size.height);
    cell.costomView = self.webVC.view;
    return cell;
}
#pragma mark - SFPageViewControllerDelegate
- (void)newsDataRefreshSuccess{
    NSLog(@"数据加载成功");
}
- (void)newsDataRefreshFail:(NSError *)error{
    NSLog(@"数据加载失败，error = %@",error);
}
- (void)dealloc{
    [[NSNotificationCenter defaultCenter] removeObserver:self];
    NSLog(@"%@ %@",[self class],NSStringFromSelector(_cmd));
}
```

3. H5 商城接入

3.1 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 = @"";

3.2 导入SDK包

将SDK拖入主工程，在项目设置中，选择Build Phases,点击左上角+号，添加Embed Frameworks，在destination类别中，选择Frameworks, 在下方添加YXLaunchAD.framework。

3.3 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 conent_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.