**QYSMobSDK\_iOS接入说明**

|  |  |  |
| --- | --- | --- |
| 版本号 | 更新时间 | 说明 |
| V1.2.0 | 2021-09-01 | 优化、修复了已知问题 |
| V1.1.0 | 2021-07-16 | 优化、修复了已知问题 |
| V1.0.0 | 2021-04-13 | iOS聚合SDK 广告类型：开屏、Banner、信息流、插屏、激励视频 |

**注意:**

当前聚合SDK版本：游可赢 YKY\_V2.3.0.220

请务必尽量使用上述版本，其他版本未经过测试

[一、概述 3](#_Toc2075369961)

[二、背景 3](#_Toc1313539953)

[1、运行环境 3](#_Toc534098911)

[2、注册开户 3](#_Toc118752717)

[3、术语介绍 3](#_Toc864606556)

[4、SDK初始化 4](#_Toc1568031090)

[三、 广告类型 4](#_Toc2126697293)

[四、SDK部署 5](#_Toc683582783)

[1、工程导入SDK文件 5](#_Toc2085806078)

[2、Xcode编译选项设置 6](#_Toc619699318)

[1) 添加“-ObjC”链接器标记 6](#_Toc2138233323)

[2) 添加HTTP权限 6](#_Toc1296110763)

[3) 添加定位权限 7](#_Toc1806962220)

[4) iOS14相关支持 7](#_Toc2047779313)

[5) 添加依赖库 8](#_Toc1453986769)

[6) 白名单 9](#_Toc939207370)

[7) 添加语言配置 11](#_Toc1253462140)

[8) 设置Bitcode 11](#_Toc123609910)

[五、 SDK接口类介绍与广告接入 12](#_Toc895070721)

[全局设置(QYSMobSDKConfig) 12](#_Toc330660612)

[1) 初始化 12](#_Toc1872711095)

[2) 获取当前SDK的版本号信息 12](#_Toc1135043233)

[1、开屏广告 -- QYSSplashAd 13](#_Toc574380730)

[1) QYSSplashAd接口说明 13](#_Toc677935845)

[2) QYSSplashAdDelegate回调说明 14](#_Toc1666999580)

[3) 开屏广告接入实例 15](#_Toc1190282298)

[2、横幅广告接入 -- QYSBannerView 17](#_Toc1264410681)

[1) QYSBannerView接口说明 17](#_Toc1599628502)

[2) QYSBannerViewDelegate 代理说明 18](#_Toc608456321)

[3) 横幅广告接入实例 18](#_Toc8260033)

[3、插屏广告接入 -- QYSInterstitialAd 19](#_Toc1387421223)

[1) QYSInterstitialAd接口说明 19](#_Toc1011055835)

[2) QYSInterstitialAdDelegate代理说明 20](#_Toc1924803781)

[3) 插屏广告接入实例 21](#_Toc483488859)

[4、信息流广告接入 -- QYSNativeExpressAd 22](#_Toc2066616612)

[1) QYSNativeExpressAd接口说明 22](#_Toc224891306)

[2) QYSNativeExpressAdDelegate代理说明 22](#_Toc176961222)

[3) 信息流广告接入实例 24](#_Toc2069890706)

[5、激励视频广告接入 -- QYSRewardVideoAd 25](#_Toc1565497989)

[1) QYSRewardVideoAd接口说明 25](#_Toc355058079)

[2) QYSRewardVideoAdDelegate代理说明 26](#_Toc1751562387)

[3) 激励视频广告接入实例 27](#_Toc803205233)

[六、 错误码 28](#_Toc388145989)

# 一、概述

本⽂档旨在帮助iOS应⽤开发者在程序中快速植⼊苏州纸影平台提供的⼴告。作为应⽤开发者，您只需要进⾏简单配置，就可以在您的应⽤中显示定制的⼴告。如果您是第⼀次接⼊我们的⼴告SDK，我们建议您利⽤Demo⼯程来了解本SDK的使⽤规范。关于 SDK 的具体使⽤⽅法，请仔细阅读下⾯的⽂档。

# 二、背景

## 1、运行环境

**开发工具：**推荐Xcode 11及以上版本

**部署目标：**iOS 9.0及以上版本

**支持架构：**arm64,armv7(只支持真机跑广告，若需要跑模拟器需替换KSAdSDK.framework)

## 2、注册开户

开发者需在苏州纸影平台上进⾏注册，在平台审核通过后，开发者就可以使用苏州纸影提供的广告功能了。请在苏州纸影变现平台上创建好应用ID和广告位ID。

## 3、术语介绍

AppID：应用ID，是您在苏州纸影平台创建应用时获得的ID，这个ID是我们在⼴告⽹络中识别您应⽤的唯⼀标识。

PlacementId: 广告位 ID，是您在苏州纸影平台为您的应⽤所创建的某种广告类型（开屏、Banner、插屏、信息流、激励视频等）的⼴告位置的ID。

## 4、SDK初始化

SDK 拉取⼴告之前，必须要进⾏初始化操作，传⼊在苏州纸影平台注册得到的AppId。⽅法如下：

#import <QYSMobSDK/QYSMobSDK.h>

-(BOOL)application:(UIApplication\*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {

// Override point for customization after application launch.

//SDK初始化

[QYSMobSDKConfig configWithAppId:@"Your appId"];

return YES;

}

注：建议在didFinishLaunchingWithOptions⽅法中，在项⽬AppDelegate.window成为**keyWindow**之后⽴即调⽤。

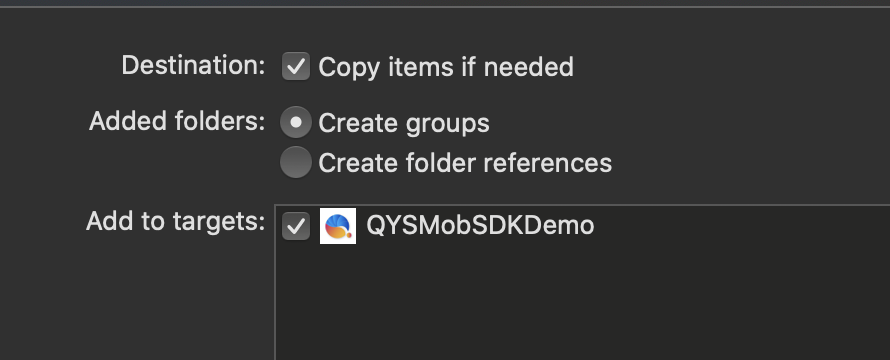
# 广告类型

|  |  |  |
| --- | --- | --- |
| **广告类型** | **简介** | **适用场景** |
| 开屏广告 | 开屏广告以App启动作为曝光时机，提供5s的可感知广告展示。 | app启动时，常会使用开屏广告 |
| [Banner广告](https://developers.adnet.qq.com/doc/ios/union/union_banner2_0) | 位于app顶部、中部、底部任意一处，横向贯穿整个app页面 | 常出现在文章页末尾，详情页面底部，信息流顶部等 |
| 插屏广告 | 插屏广告是移动广告的一种常见形式，在应用开流程中弹出。当应用展示插页式广告时，用户可以选择点按广告，访问其目标网址，也可以将其关闭，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。希望广告展示时保留应用原有元素可使用插屏广告  插屏广告分为图片插屏和视频插屏。 |
| 信息流广告 | 契合应用原生环境的广告样式，提供浑然一体的用户体验，助您最大程度获取收益。 | 常见场景：资讯页、视频详情页、信息列表页等  支持格式：文字+小图、文字+大图、文字+三图、文字+多图、文字+视频 |
| 激励视频广告 | 通过应用内奖励吸引用户主动选择观看的全屏视频形式广告，用户获得应用内奖励的同时，开发者获得丰厚变现收益。 | 常见场景：常见展示于获取应用内奖励及权益体系的所有应用及场景  支持格式：15～30秒竖版视频、横版视频 |

# 四、SDK部署

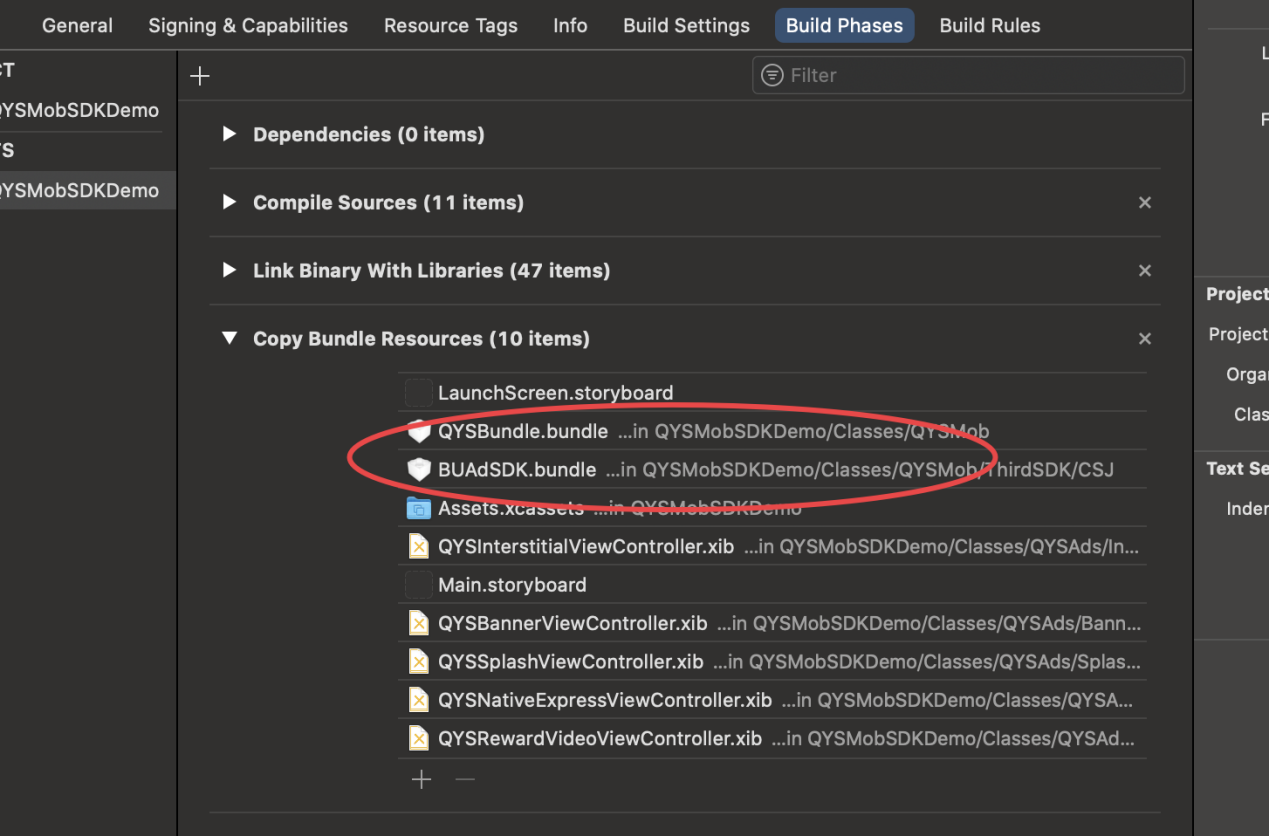
## 1、工程导入SDK文件

直接将 {QYSMobSDK.framework、QYSBundle.bundle、支持的第三方SDK库文件}用”Add Files to xxxxx”方式添加到工程中即可。（或者直接将QYSMob文件夹拖入工程）



注：\*\*升级SDK必须同时更新framework和bundle文件，否则可能出现部分页面无法展示的问题\*\*

拖入完请确保Copy Bundle Resources中有QYSBundle.bundle和BUAdSDK.bundle（不集成CSJ则忽略BUAdSDK.bundle），否则可能出现部分页面无法展示的情况。



注：若有引入快手SDK则需配置Target->General->Frameworks, Libraries, and Embedded Content设置KSAdSDK.framework 的 Embed 状态为 Embed&Sign。

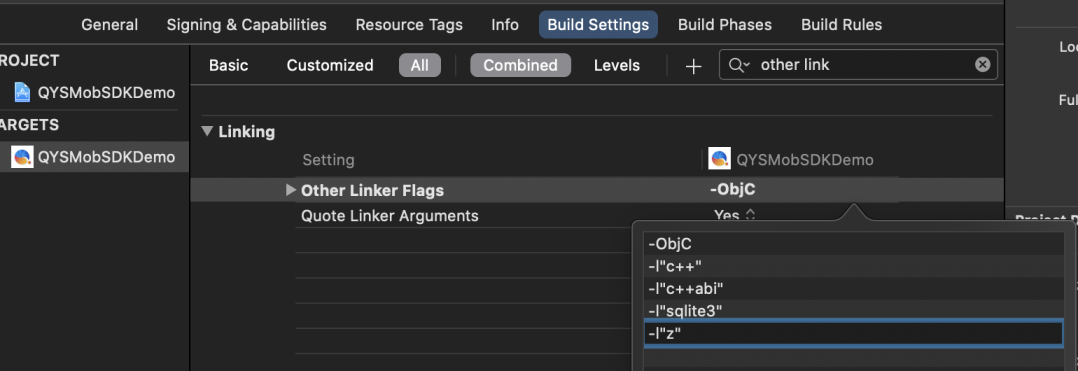
ps:为了方便模拟器开发，KSAdSDK带有x86\_ 64, i386架构。在打发布到AppStore的安装包时需要移除这两个架构(CocoaPods 方式接入会自动移除),当前集成默认是移除状态，接入者可自行替换。

## 

## 2、Xcode编译选项设置

### 1) 添加“-ObjC”链接器标记

在Xcode中选择项目的Targets->Build Settings，配置Other Link Flags 增加 \*\*-ObjC\*\*。注意大小写。新增 -l"c++"、 -l"c++abi" 、-l"sqlite3"、-l"z" 。



### 2) 添加HTTP权限

应⽤的 Info.plist 添加相应配置信息，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。

\*\*注意：\*\* \*\*Allow Arbitrary Loads in Web Content\*\* key存在时会忽略Allow Arbitrary Loads的设置（iOS10之后）

```

<key>NSAppTransportSecurity</key>

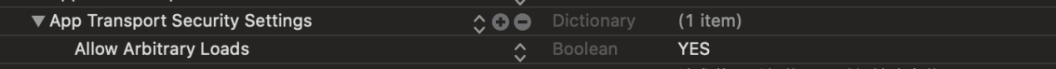
<dict>

<key>NSAllowsArbitraryLoads</key>

<true/>

</dict>

```



### 3) 添加定位权限

SDK 不会主动获取应⽤位置权限，当应⽤本身有获取位置权限逻辑时，需要在应⽤的 Info.plist 添加相应配置信息，避免 App Store 审核被拒。

工程Info.plist文件设置，点击右边的Information Property List后边的 "+" 展开

应⽤根据实际情况配置

```

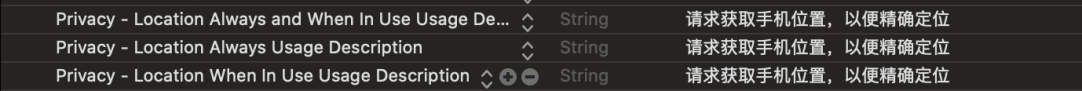
Privacy - Location When In Use Usage Description

Privacy - Location Always and When In Use Usage Description

Privacy - Location Always Usage Description

Privacy - Location Usage Description

```



### 4) iOS14相关支持

(1)应用编译环境升级至 Xcode 12.0 及以上版本

(2)从 iOS 14 开始，在应用程序调用 App Tracking Transparency 向用户提跟踪授权请求之前，IDFA 将不可用。 如果应用未提出此请求，应用获取到的 IDFA 将自动清零，可能会导致您的广告收入的降低。

要获取 App Tracking Transparency 权限，请更新您的 Info.plist，添加 NSUserTrackingUsageDescription 字段和自定义文案描述。代码示例：

```

<key>NSUserTrackingUsageDescription</key>

<string>获取设备信息用以精准推送您喜欢的内容。</string>

```



### 5) 添加依赖库

在Xcode中选中工程名，在Target->Build Phases->Link Binary With Libraries中点击“+”,在弹出窗口输入库名称，出现后点击“Add”则将库引入到系统中。

AppTrackingTransparency.framework    iOS 14+，Optional

Accelerate.framework

AddressBook.framework

AdSupport.framework

AssetsLibrary.framework

AudioToolbox.framework

AVFoundation.framework

AVKit.framework

CoreData.framework

CoreGraphics.framework

CoreImage.framework

CoreLocation.framework

CoreMedia.framework

CoreMotion.framework

CoreServices.framework

CoreTelephony.framework

CoreText.framework

Foundation.framework

ImageIO.framework

JavaScriptCore.framework

MapKit.framework

MediaPlayer.framework

MessageUI.framework

MobileCoreServices.framework

Photos.framework

QuartzCore.framework

QuickLook.framework

SafariServices.framework

Security.framework

StoreKit.framework

SystemConfiguration.framework

UIKit.framework

WebKit.framework

libbz2.tbd

libc++.tbd

libc++abi.tbd

libiconv.tbd

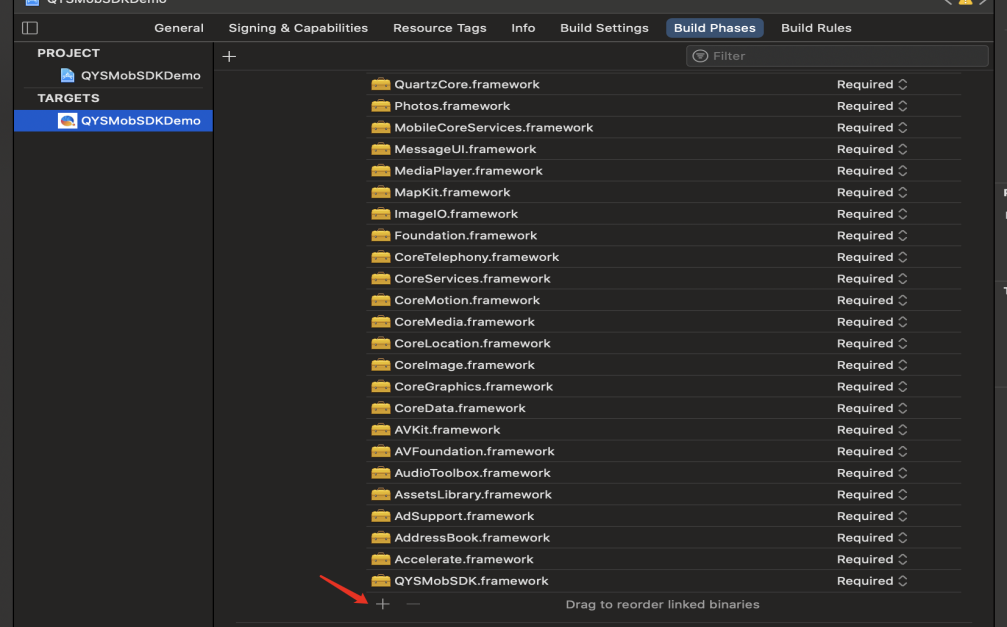
libresolv.9.tbd

libresolv.tbd

libsqlite3.tbd

libxml2.tbd

libz.tbd



6) 白名单

为了支持应用间的跳转，工程需在Info.plist添加常用的URL Scheme 白名单，以下仅供参考：

```

<key>LSApplicationQueriesSchemes</key>

    <array>

        <!-- 微信 URL Scheme 白名单-->

        <string>wechat</string>

        <string>weixin</string>

        <!-- 新浪微博 URL Scheme 白名单-->

        <string>sinaweibohd</string>

        <string>sinaweibo</string>

        <string>sinaweibosso</string>

        <string>weibosdk</string>

        <string>weibosdk2.5</string>

        <!-- QQ、Qzone URL Scheme 白名单-->

        <string>mqqapi</string>

        <string>mqq</string>

        <string>mqqOpensdkSSoLogin</string>

        <string>mqqconnect</string>

        <string>mqqopensdkdataline</string>

        <string>mqqopensdkgrouptribeshare</string>

        <string>mqqopensdkfriend</string>

        <string>mqqopensdkapi</string>

        <string>mqqopensdkapiV2</string>

        <string>mqqopensdkapiV3</string>

        <string>mqzoneopensdk</string>

        <string>wtloginmqq</string>

        <string>wtloginmqq2</string>

        <string>mqqwpa</string>

        <string>mqzone</string>

        <string>mqzonev2</string>

        <string>mqzoneshare</string>

        <string>wtloginqzone</string>

        <string>mqzonewx</string>

        <string>mqzoneopensdkapiV2</string>

        <string>mqzoneopensdkapi19</string>

        <string>mqzoneopensdkapi</string>

        <string>mqzoneopensdk</string>

        <!-- 支付宝 淘宝  URL Scheme 白名单-->

        <string>alipay</string>

        <string>alipayshare</string>

        <string>taobao</string>

        <string>itaobao</string>

<string>tbopen</string>

        <!-- 天猫  URL Scheme 白名单-->

        <string>tmall</string>

        <string>itmall</string>

        <!-- 京东 Scheme 白名单-->

<string>jd</string>

        <string>openApp.jdMobile</string>

        <!-- 大众点评 Scheme 白名单-->

        <string>dianping</string>

         <!-- 聚美优品 Scheme 白名单-->

        <string>JuMei</string>

        <string>jumeimall</string>

         <!-- 美团 Scheme 白名单-->

        <string>meituan0000</string>

        <string>imeituan</string>

<string>meituanwaimai</string>

<!-- 苏宁 Scheme 白名单-->

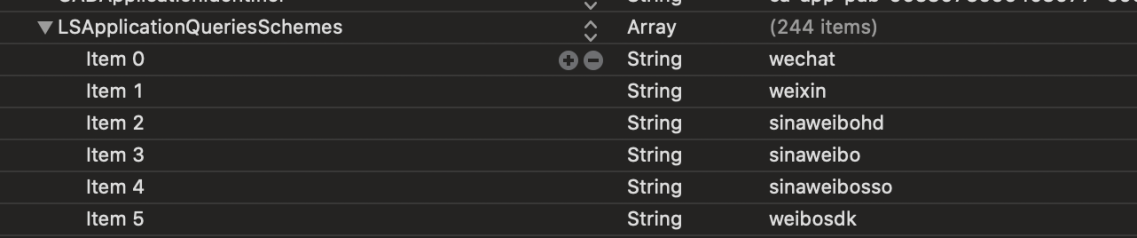
        <string>suning</string>

         <!--  唯品会 Scheme 白名单-->

         <string>vipshop</string>

</array>

```

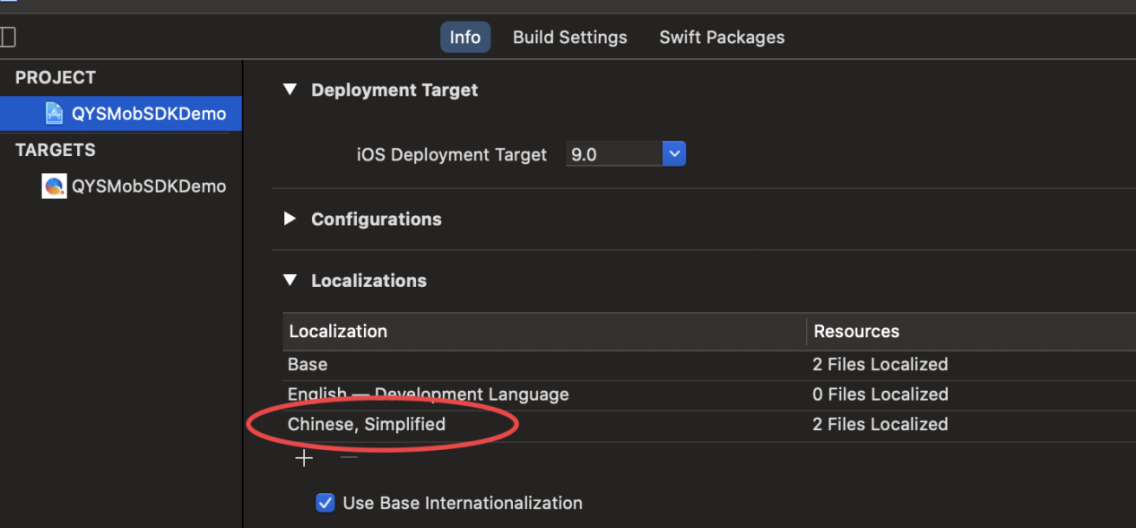


### 7) 添加语言配置

注意 : 开发者须在这里设置所支持的语言,否则会有语言显示的问题.

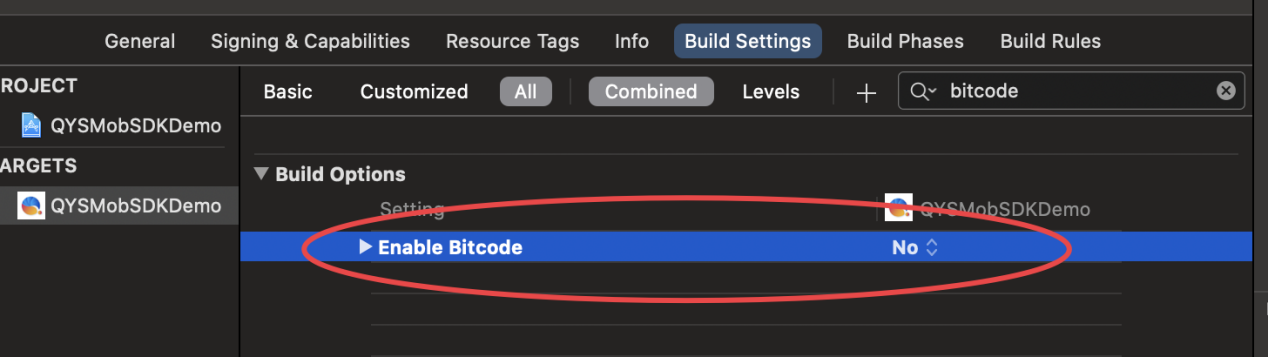
\*\*例如 : 支持中文 添加 Chinese\*\*

工程需要在PROJECT -> Info中找到Localization，点击“+”，选择 Chinese，Simplified



### 8) 设置Bitcode

如果工程报'~XXX.o'does not contain bitcode. You must rebuild it with bitcode enabled (Xcode setting ENABLE\_BITCODE)的错误，则在Target->Build Phases->Enable Bitcode 设置为NO。



# SDK接口类介绍与广告接入

详细接入实例请查看Demo工程，使用时导入头文件：

**#import <QYSMobSDK/QYSMobSDK.h>**

## 全局设置(QYSMobSDKConfig)

QYSMobSDKConfig 类是整个 SDK 设置的入口和接口，可以设置 SDK 的一些全局信息，提供类方法获取设置结果。

### 1) 初始化

SDK 拉取⼴告之前，必须要进⾏初始化操作，传⼊在苏州纸影平台注册得到的AppId。

SDK 需要在 AppDelegate 的方法 ```- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions``` 里进行初始化。

注：建议在didFinishLaunchingWithOptions⽅法中，在项⽬ AppDelegate.window 成为keyWindow之后⽴即调⽤。

```

[QYSMobSDKConfig configWithAppId:@"Your appId"];

```

### 2) 获取当前SDK的版本号信息

```

[QYSMobSDKConfig **sdkVersion];**

```

## 1、开屏广告 -- QYSSplashAd

类型说明：开屏广告主要是APP 启动加载时的全屏或半屏广告；展现形式为静态图片、动态视频。提供5s的可感知⼴告展示

### 1) **QYSSplashAd**接口说明

```

@interface QYSSplashAd : NSObject

/\*\*

\* 委托对象

\*/

@property (nonatomic, weak) id <QYSSplashAdDelegate>delegate;

/\*\*

\* 构造方法

\* 详解：placementId - 广告位 ID

\*/

- (instancetype)initWithPlacementId:(NSString \*)placementId;

/\*\*

\* 拉取广告超时时间，默认为5秒

\*详解：拉取广告超时时间，开发者调用loadAd方法以后会立即展示backgroundImage，然后在该超时时间内，如果广告拉取成功，则立马展示开屏广告，否则放弃此次广告展示机会。

\*/

@property (nonatomic, assign) CGFloat fetchDelay;

/\*\*

\* 开屏广告的背景图片

\* 可以设置背景图片作为开屏加载时的默认背景（部分广告支持）

\*/

@property (nonatomic, strong) UIImage \*backgroundImage;

/\*\*

\* 自定义底部视图

\* 高度上限：屏幕高度\*25%

\*/

@property (nonatomic, strong) UIView \*bottomView;

/\*\*

\* 返回广告是否可展示

\* @return 当广告已经加载完成且未曝光时，为YES，否则为NO

\*/

- (BOOL)isAdValid;

/\*\*

\* 拉取并展示广告

\* window 展示开屏的容器

\*/

- (void)loadAdAndShowInWindow:(UIWindow \*)window;

@end

```

### 2) **QYSSplashAdDelegate**回调说明

```

@protocol QYSSplashAdDelegate <NSObject>

@optional

/\*\*

\* 广告请求成功回调

\*/

- (void)quys\_SplashRequestSuccess:(QYSSplashAd \*)splashAd;

/\*\*

\* 广告请求失败回调

\*/

- (void)quys\_SplashRequestFail:(QYSSplashAd \*)splashAd error:(NSError\*)error;

/\*\*

\* 广告曝光回调

\*/

- (void)quys\_SplashExposured:(QYSSplashAd \*)splashAd;

/\*\*

\* 广告点击回调

\*/

- (void)quys\_SplashClicked:(QYSSplashAd \*)splashAd;

/\*\*

\* 广告关闭回调

\*/

- (void)quys\_SplashClosed:(QYSSplashAd \*)splashAd;

```

### 3) 开屏广告接入实例

备注: \*\*目前开屏广告只针对 iPhone 设备在垂直方向上展示。\*\*

开屏广告提供全屏开屏接口和半屏展示接口。

开屏广告支持开发者自定义设置开屏底部的界面，用以展示应用 Logo 等。嵌入代码如下:

```

#import <QYSMobSDK/QYSMobSDK.h>

@interface AppDelegate ()<QYSSplashAdDelegate>

@property (nonatomic,strong) QYSSplashAd \*splashAd;

@end

...

-(BOOL)application:(UIApplication\*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {

// Override point for customization after application launch.

//1.SDK初始化

[QYSMobSDKConfig configWithAppId:@"Your appId"];

//设置开屏底部自定义LogoView，展示半屏开屏广告，logo的高度建议不要过高，以免影响广告的展示效果。

UIView \*bottomView;

{

bottomView = [[UIView alloc] initWithFrame:CGRectMake(0, 0, [UIScreen mainScreen].bounds.size.width, 100)];

UIImage \*image = [UIImage imageNamed:@"xxx\_logo"];

UIImageView \*logoView = [[UIImageView alloc] initWithImage:image];

logoView.contentMode = UIViewContentModeScaleAspectFill;

logoView.frame = bottomView.bounds;

[bottomView addSubview:logoView];

}

//2. 开屏广告初始化并展示代码

self.splashAd = [[QYSSplashAd alloc]initWithPlacementId:@"Your 广告位ID"];

self.splashAd.delegate = self;

if (bottomView) {

self.splashAd.bottomView = bottomView;

}

self.splashAd.backgroundImage = [UIImage imageNamed:@"Your LaunchImage"];

self.splashAd.fetchDelay = 5.0;

UIWindow \*window = [[UIApplication sharedApplication] keyWindow];

[self.splashAd loadAdAndShowInWindow:window];

return YES;

}

#pragma mark -- QYSSplashAdDelegate --

/\*\*

\* 广告请求成功回调

\*/

- (void)quys\_SplashRequestSuccess:(QYSSplashAd \*)splashAd{

}

/\*\*

\* 广告请求失败回调

\*/

- (void)quys\_SplashRequestFail:(QYSSplashAd \*)splashAd error:(NSError\*)error{

}

/\*\*

\* 广告曝光回调

\*/

- (void)quys\_SplashExposured:(QYSSplashAd \*)splashAd{

}

/\*\*

\* 广告点击回调

\*/

- (void)quys\_SplashClicked:(QYSSplashAd \*)splashAd{

}

/\*\*

\* 广告关闭回调

\*/

- (void)quys\_SplashClosed:(QYSSplashAd \*)splashAd{

self.splashAd = nil;

}

```

## 2、**横幅广告接入 -- QYSBannerView**

类型说明：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；并可在一段时间后自动刷新。

BannerView宽高比为固定640:100，开发者在嵌入Banner时，可以手动设置Banner条的宽度用来满足场景需求，根据宽高比动态调整高度，以此保证显示效果。

### 1) **QYSBannerView接口说明**

```

@interface QYSBannerView: NSObject

/\*\*

\* 委托对象

\*/

@property (nonatomic, weak) id <QYSBannerViewDelegate>delegate;

/\*\*

\* 构造方法

\* 详解：placementId - 广告位 ID

\* controller - 视图控制器

\* Banner 比例 640:100

\*/

- (instancetype)initWithFrame:(CGRect)frame placementId:(NSString \*)placementId rootViewController:(UIViewController\*)controller;

/\*\*

\* 拉取并展示广告

\*/

- (void)loadAdAndShow;

@end

```

### 2) **QYSBannerViewDelegate 代理说明**

```

@protocol QYSBannerViewDelegate <NSObject>

@optional

//广告请求成功回调

- (void)quys\_BannerRequestSuccess:(QYSBannerView \*)bannerView;

//广告请求失败回调

- (void)quys\_BannerRequestFail:(QYSBannerView \*)bannerView error:(NSError\*)error;

//广告曝光回调

- (void)quys\_BannerExposured:(QYSBannerView \*)bannerView;

//广告点击回调

- (void)quys\_BannerClicked:(QYSBannerView \*)bannerView;

//广告关闭回调

- (void)quys\_BannerClosed:(QYSBannerView \*)bannerView;

@end

```

### 3) 横幅广告接入实例

```

#import <QYSMobSDK/QYSMobSDK.h>

@interface AppDelegate ()<QYSBannerViewDelegate>

@property (nonatomic,strong) QYSBannerView \*bannerView;

@end

//加载广告

self.bannerView = [[QYSBannerAd alloc]initWithFrame:CGRectMake(0, 0, [UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.width \* 100 / 640) placementId:@"Your 横幅广告位ID" rootViewController:self];

self.bannerView .delegate = self;

[self.view addSubview:self.bannerView ];

[self.bannerView loadAdAndShow];

```

## 3、插屏广告接入 -- QYSInterstitialAd

类型说明：插屏广告是移动广告的一种常见形式，在应用开流程中弹出，当应用展示插页式广告时，用户可以选择点按广告，访问其目标网址，也可以将其关闭，返回应用。

插屏广告分为图片插屏和视频插屏。

### 1) QYSInterstitialAd接口说明

```

@interface QYSInterstitialAd : NSObject

/\*\*

\* 委托对象

\*/

@property (nonatomic, weak) id <QYSInterstitialAdDelegate>delegate;

/\*\*

\* 构造方法

\* 详解：placementId - 广告位 ID

\*/

- (instancetype)initWithPlacementId:(NSString \*)placementId;

/\*\*

\* 返回广告是否可展示

\* @return 当广告已经加载完成且未曝光时，为YES，否则为NO

\*/

- (BOOL)isAdValid;

/\*\*

\* 加载广告方法，只拉取不展示

\*/

- (void)loadAd;

/\*\*

\* 展示广告方法

\* 详解：rootViewController 必须传入用于显示插播广告的UIViewController

\*/

- (void)showAdFromRootViewController:(UIViewController \*)rootViewController;

@end

```

### 2) QYSInterstitialAdDelegate代理说明

```

@protocol QYSInterstitialAdDelegate<NSObject>

@optional

/\*\*

\* 广告请求成功回调

\*/

- (void)quys\_InterstitialRequestSuccess:(QYSInterstitialAd \*)interstitialAd;

/\*\*

\* 广告请求失败回调

\*/

- (void)quys\_InterstitialRequestFail:(QYSInterstitialAd \*)interstitialAd error:(NSError\*)error;

/\*\*

\* 广告曝光回调

\*/

- (void)quys\_InterstitialExposured:(QYSInterstitialAd \*)interstitialAd;

/\*\*

\* 广告点击回调

\*/

- (void)quys\_InterstitialClicked:(QYSInterstitialAd \*)interstitialAd;

/\*\*

\* 广告关闭回调

\*/

- (void)quys\_InterstitialClosed:(QYSInterstitialAd \*)interstitialAd;

@end

```

### 3) 插屏广告接入实例

```

#import <QYSMobSDK/QYSMobSDK.h>

@interface AppDelegate ()<QYSInterstitialAdDelegate>

@property (nonatomic,strong) QYSInterstitialAd \*interstitialAd;

@end

//加载广告

self.interstitialAd = [[QYSInterstitialAd alloc]initWithPlacementId:@"Your 插屏广告位ID"];

self.interstitialAd.delegate = self;

[self.interstitialAd loadAd];

//展示广告

if ([self.interstitialAd isAdValid]) {

[self.interstitialAd showAdFromRootViewController:self];

}

```

## 4、信息流广告接入 -- QYSNativeExpressAd

类型说明：信息流广告是指当用户在浏览、阅读内容时安插的广告。主要包括文字+小图、文字+大图、文字+三图、文字+多图、文字+视频等展现形式。

### 1) QYSNativeExpressAd接口说明

```

@interface QYSNativeExpressAd : NSObject

/\*\*

\* 委托对象

\*/

@property (nonatomic, weak) id <QYSNativeExpressAdDelegate>delegate;

/\*\*

\* 构造方法

\* 详解：placementId - 广告位 ID

\* adSize - 广告展示的宽高 当 height = 0，自动根据 width 算高；

\*/

- (instancetype)initWithPlacementId:(NSString \*)placementId adSize:(CGSize)size;

/\*\*

\* 加载广告方法，只拉取不展示

\* count 一次性加载的广告数量 (实际拿到的广告数量有机率小于count)

\*/

- (void)loadAd:(NSInteger)count;

@end

```

### 2) QYSNativeExpressAdDelegate代理说明

```

@protocol QYSNativeExpressAdDelegate<NSObject>

@optional

/\*\*

\* 信息流广告请求成功回调

\*/

- (void)quys\_nativeExpressRequestSuccess:(QYSNativeExpressAd \*)nativeExpressAd views:(NSArray<\_\_kindof QYSNativeExpressAdView \*> \*)views;

/\*\*

\* 信息流广告请求失败回调

\*/

- (void)quys\_nativeExpressRequestFail:(QYSNativeExpressAd \*)nativeExpressAd error:(NSError\*)error;

#pragma mark -- QYSNativeExpressAdView 相关 --

/\*\*

\* 信息流广告渲染成功, 此时的 nativeExpressAdView.size.height 根据 size.width 完成了动态更新。

\*/

-(void)quys\_nativeExpressAdViewRenderSuccess:(QYSNativeExpressAdView \*)nativeExpressAdView;

/\*\*

\* 信息流广告渲染失败

\*/

-(void)quys\_nativeExpressAdViewRenderFail:(QYSNativeExpressAdView \*)nativeExpressAdView;

/\*\*

\* 信息流广告曝光回调

\*/

-(void)quys\_nativeExpressAdViewExposured:(QYSNativeExpressAdView \*)nativeExpressAdView;

/\*\*

\* 信息流广告点击回调

\*/

-(void)quys\_nativeExpressAdViewClicked:(QYSNativeExpressAdView \*)nativeExpressAdView;

/\*\*

\* 信息流广告关闭回调

\*/

-(void)quys\_nativeExpressAdViewClosed:(QYSNativeExpressAdView \*)nativeExpressAdView;

@end

```

### 3) 信息流广告接入实例

详细接入实例请查看Demo工程

```

#import <QYSMobSDK/QYSMobSDK.h>

@interface AppDelegate ()<QYSNativeExpressAdDelegate>

@property (nonatomic,strong) QYSNativeExpressAd \*nativeExpressAd;

@property (nonatomic, strong) NSMutableArray \*expressAdViews;//获取到的信息流广告时图数组

@end

//加载广告

self.nativeExpressAd = [[QYSNativeExpressAd alloc]initWithPlacementId:@"Your 信息流广告位ID" adSize:CGSizeMake([UIScreen mainScreen].bounds.size.width, 0)];

self.nativeExpressAd.delegate = self;

/\*\*

\* 信息流广告请求成功回调

\*/

- (void)quys\_nativeExpressRequestSuccess:(QYSNativeExpressAd \*)nativeExpressAd views:(NSArray<\_\_kindof QYSNativeExpressAdView \*> \*)views{

// [self.expressAdViews removeAllObjects];//【重要】不能保存太多view，需要在合适的时机手动释放不用的，否则内存会过大

[self.expressAdViews addObjectsFromArray:views];

if (views.count) {

[views enumerateObjectsUsingBlock:^(id \_Nonnull obj, NSUInteger idx, BOOL \* \_Nonnull stop) {

QYSNativeExpressAdView \*expressView = (QYSNativeExpressAdView \*)obj;

expressView.controller = self;

[expressView render];

}];

}

}

//展示广告

获取到 self.expressAdViews 数组里的 QYSNativeExpressAdView 添加展示

```

## 5、激励视频广告接入 -- QYSRewardVideoAd

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应用内虚拟货币或者应用内物品和内容等等；

这类广告的长度为 15-30秒，默认为不可跳过，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。支持竖屏和横屏展示。

⽤户观看短视频⼴告后可以得到⼀些应⽤内奖励。

### 1) QYSRewardVideoAd接口说明

```

@interface QYSRewardVideoAd : NSObject

/\*\*

\* 委托对象

\*/

@property (nonatomic, weak) id <QYSRewardVideoAdDelegate>delegate;

/\*\*

\* 构造方法

\* 详解：placementId - 广告位 ID

\*/

- (instancetype)initWithPlacementId:(NSString \*)placementId;

/\*\*

\*  播放视频是否静音 默认为 NO

    videoMuted 静音为 YES 扬声 NO

\*/

@property (nonatomic, assign) BOOL videoMuted;

/\*\*

\*  视频是否为横屏展示 默认为 NO

    isHorizontalVideo 横屏为 YES 竖屏为NO

\*/

@property (nonatomic, assign) BOOL isHorizontalVideo;

/\*\*

\* 返回广告是否可展示

\* @return 当广告已经加载完成且未曝光时，为YES，否则为NO

\*/

- (BOOL)isAdValid;

/\*\*

\* 加载广告方法，只拉取不展示

\*/

- (void)loadAd;

/\*\*

\* 展示广告方法

\* 详解：rootViewController 必须传入用于显示插播广告的UIViewController

\*/

- (void)showAdFromRootViewController:(UIViewController \*)rootViewController;

@end

```

### 2) QYSRewardVideoAdDelegate代理说明

```

@protocol QYSRewardVideoAdDelegate<NSObject>

@optional

/\*\*

\* 广告请求成功回调

\*/

- (void)quys\_rewardVideoRequestSuccess:(QYSRewardVideoAd \*)rewardVideoAd;

/\*\*

视频数据下载成功回调，已经下载过的视频会直接回调

\*/

- (void)quys\_rewardVideoAdVideoDidLoad:(QYSRewardVideoAd \*)rewardVideoAd;

/\*\*

\* 广告各种错误信息回调

\*/

- (void)quys\_rewardVideoDidFail:(QYSRewardVideoAd\*)rewardVideoAd error:(NSError\*)error;

/\*\*

\* 广告曝光回调

\*/

- (void)quys\_rewardVideoExposured:(QYSRewardVideoAd \*)rewardVideoAd;

/\*\*

\* 广告点击回调

\*/

- (void)quys\_rewardVideoClicked:(QYSRewardVideoAd \*)rewardVideoAd;

/\*\*

\* 广告关闭回调

\*/

- (void)quys\_rewardVideoClosed:(QYSRewardVideoAd \*)rewardVideoAd;

/\*\*

\* 广告开始播放回调

\*/

- (void)quys\_rewardVideoPlaystart:(QYSRewardVideoAd\*)rewardVideoAd;

/\*\*

\* 广告播放结束回调

\*/

- (void)quys\_rewardVideoPlayEnd:(QYSRewardVideoAd\*)rewardVideoAd;

/\*\*

\* 广告播放达到激励条件回调

\*/

- (void)quys\_rewardVideoDidRewardEffective:(QYSRewardVideoAd\*)rewardVideoAd;

@end

```

### 3) 激励视频广告接入实例

```

#import <QYSMobSDK/QYSMobSDK.h>

@interface AppDelegate ()<QYSRewardVideoAdDelegate>

@property (nonatomic,strong) QYSRewardVideoAd \*rewardVideoAd;

@end

//加载广告

self.rewardVideoAd = [[QYSRewardVideoAd alloc]initWithPlacementId:@"Your 激励视频广告位ID"];

self.rewardVideoAd.delegate = self;

[self.rewardVideoAd loadAd];

//展示广告

if (self.rewardVideoAd.isAdValid) {

[self.rewardVideoAd showAdFromRootViewController:self];

}

```

# 错误码

|  |  |
| --- | --- |
| 错误码 | **错误码说明** |
| 2001 | **初始化失败** |
| 200101 | **包名校验错误** |
| 200102 | **广告位错误** |
| 200103 | **网络异常** |
| 200104 | **请求超时** |
| 2002 | **无广告配置** |
| 200201 | **无广告填充** |
| 200202 | **服务端数据错误** |
| 200203 | **展示次数达到上限** |
| 200204 | **请求参数错误** |
| 3001 | **广告展示异常** |
| 300101 | **广告物料下载错误** |
| 300102 | **图片加载错误** |
| 300103 | **视频加载错误** |
| 300104 | **视频播放错误** |
| 99999 | **未知错误，请联系苏州纸影技术人员** |
|  |  |