适配i'pei﷽﷽﷽﷽﷽﷽﷽﷽适配i'pei﷽﷽﷽﷽﷽﷽﷽﷽适配i'pei﷽﷽﷽﷽﷽﷽﷽﷽

**非零无限UBK平台 iOS SDK开发者文档**

版本：1.0.2

**深圳市非零无限科技有限公司**

**版权所有 侵权必究**

**目录**

1. 集成准备 2

1.1获取APPID和APPKEY 2

1.2下载SDK 2

1.3导入SDK 2

1.4把Framework添加到‘Embedded Binaries’中 3

2. 设置工程属性 4

2.1配置Xcode7 Https请求 4

2.2 SDK初始化 4

2.3 调用展示接口 5

2.3.1 banner展示 5

2.3.2 推荐墙展示 6

2.3.3 自定义原生banner 7

2.3.5 自定义图文信息流广告 10

2.4 终止SDK 11

3. 设置调试模式 11

4. 广告展示回调 12

5. 联系方式 13

# 1. 集成准备

## 1.1 获取APPID和APPKEY

集成SDK之前，您首先需要到UBK[官网](http://www.feelwx.com/pc/index.html)（<http://www.feelwx.com/pc/index.html>）注册并且添加新应用，获得Appid和Appkey。



提醒：我们建议开发者在注册账号时使用公司企业邮箱，避免使用个人邮箱注册后由于个人离职带来的问题。

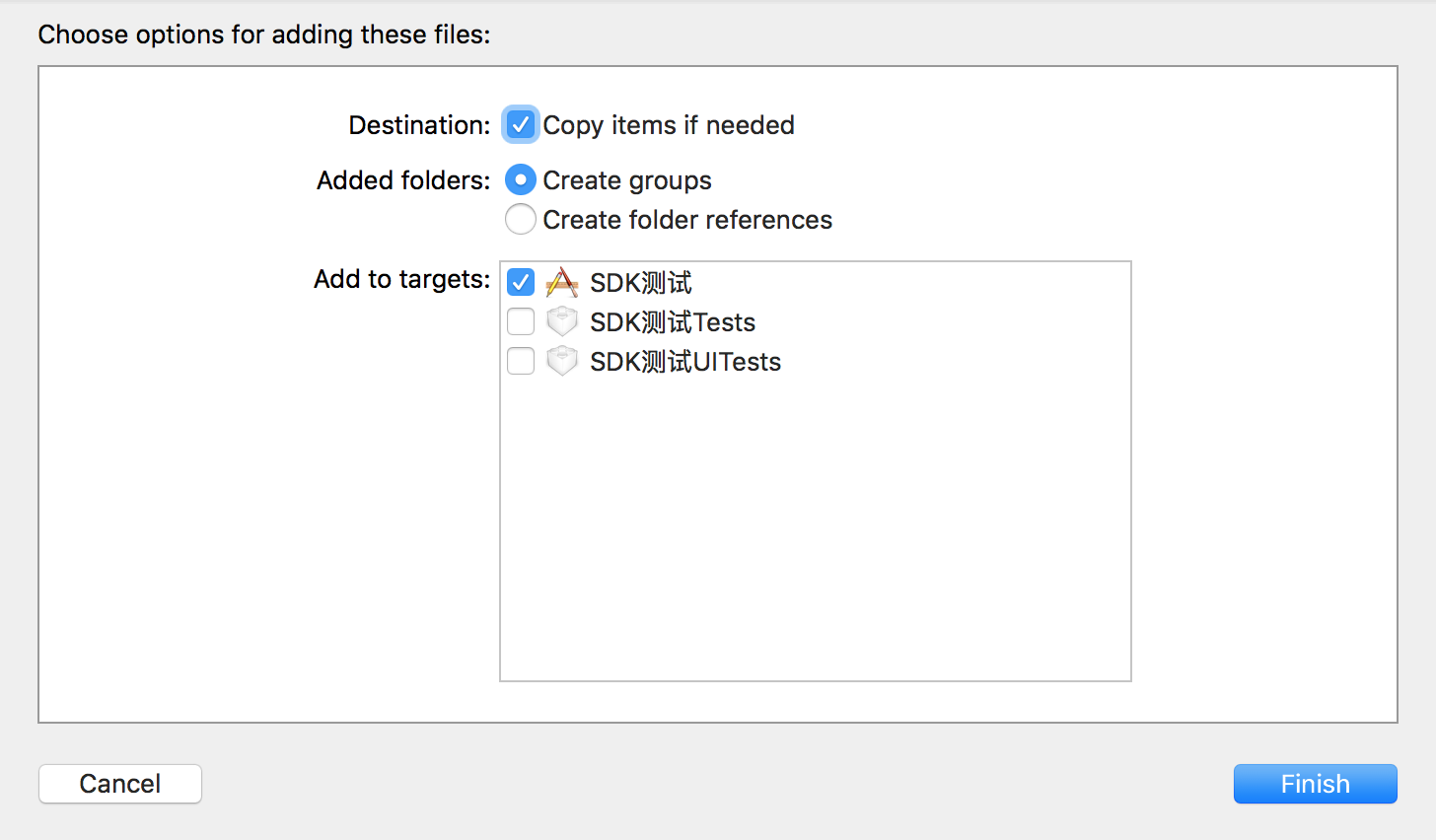
## 1.2 下载SDK

官方下载地址：

<http://www.feelwx.com/pc/sdk/download/UBKSDK_iOS.zip>

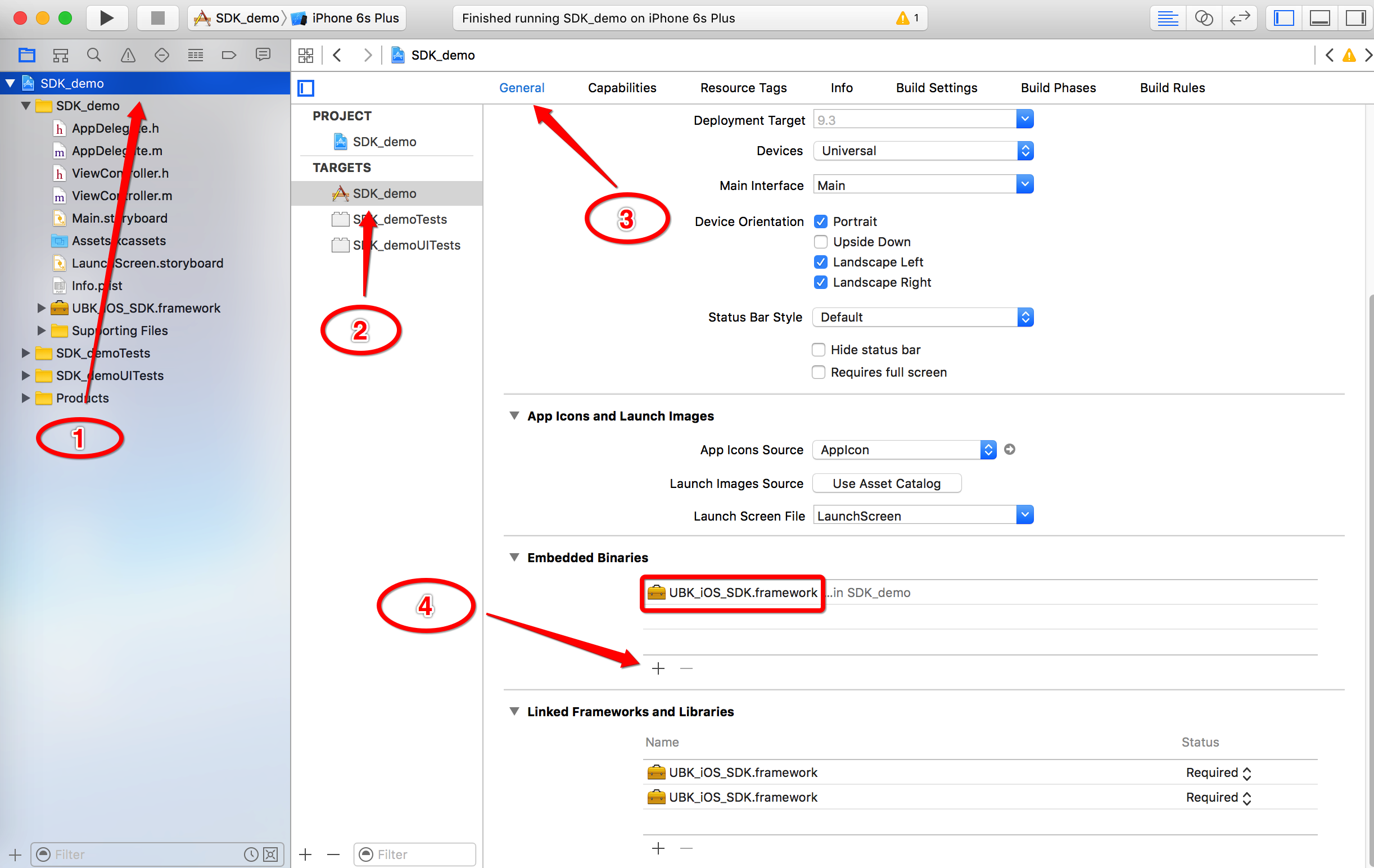
## 1.3 导入SDK

将解压后的文件(UBK\_iOS\_SDK.framework)拖入您的工程中，并勾选上Destination,如图所示：



## 1.4 把Framework添加到‘Embedded Binaries’中

步骤如下图所示：

****

注意：添加完毕后可能会在Linked Frameworks and Libraries中出现两个相同的 .framework ，属于正常现象，不影响使用。

# 设置工程属性

## 2.1 配置Xcode7 Https请求

特此声明：接入该SDK不需要在Xcode7中配置Https，可以直接使用。

建议使用Xcode7+，如果您仍使用Xcode6系列，在接入后如果SDK给您带来编译不通过等相关问题请及时联系我们。

## 2.2 SDK初始化

引入相关头文件 #import "UBK\_iOS\_SDK/UBK\_iOS\_SDK.h"（写其他方式不能正确编译）

在工程的 AppDelegate 中的以下方法中，调用 SDK 初始化方法：

#import "UBK\_iOS\_SDK/UBK\_iOS\_SDK.h"

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

{

UBKAD \*ubkad = [[UBKAD alloc] initWithChannel:@"AppStroe" app\_id:00000 app\_key:@"ABCDEFGHIJKLMNOP"];

return YES;

}

在应用的Application初始化时调用。注意不要在其他 .m文件中调用，避免重复初始化。

SDK的初始化方法参数说明：

-(instancetype)initWithChannel:(NSString\*)channel app\_id:(NSInteger )app\_id app\_key:(NSString \*)app\_key;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| channel | NSString | 渠道号（可为nil默认为AppStore） |
| app\_id | | NSInteger | 官网申请的APP ID |
| app\_key | | NSString | 官网申请的APP KEYY |

## 2.3 调用展示接口

注意：此版本不支持横屏展示，建议不要在可以横竖屏切换的ViewController里面调用展示接口。

### 2.3.1 banner展示

①初始化请求类型为banner的UBKADRequest对象

UBKADRequest \*adRequest = [[UBKADRequest alloc] initWithBannerType:BannerOnTop marginTop:20 marginBottom:0 onView:nil];

②请求banner，参数为UBKADRequest对象

[UBKAD requestADWithADRequest:adRequest];

参数说明：

-(instancetype)initWithBannerType:(BannerTypeName)bannerType marginTop:(float)marginTop marginBottom:(float)marginBottom onView:(UIView \*)adContainer;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| bannerType | BannerTypeName | banner的三种类型：  BannerOnTop = 1//顶部  BannerOnMiddle,//正中间  BannerOnBottom //底部 |
| marginTop | float | banner距view顶部高度，如果banner类型不为BannerOnTop，此值无效 |
| marginBottom | float | banner距view底部高度，如果banner类型不为BannerOnBottom，此值无效 |
| adContainer | UIView | Banner展示的位置（可选） |

### 2.3.2 推荐墙展示

①初始化请求类型为推荐墙的UBKADRequest对象

UBKADRequest \*adRequest = [[UBKADRequest alloc] initWithADType:AdTypeRecommend];

②请求推荐墙，参数为UBKADRequest对象

[UBKAD requestADWithADRequest:adRequest];

参数说明：

-(instancetype)initWithADType:(ADTypeName)adType;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| adType | ADTypeName | 推荐应用的展示类型(必填):  AdTypeHalfScreen = 2,//半屏插屏  AdTypeNotify, //通知  AdTypeRecommend, //推荐墙  AdTypeFullScreen //全屏插屏 |

### 2.3.3 自定义原生banner

调用下面代码就可以返回banner的view数组，开发者只需在block中对view进行操作即可。

[[UBKAdLoader new] loadAdWithType:AdTypeCustomBanner count:1 size:CGSizeMake(KWIDTH, 120) block:^(NSArray \*adViewArray) {

if (adViewArray.count) {

[adViewArray[0] setFrame:CGRectMake(0, 150, KWIDTH, 120)];

[self.view addSubview:adViewArray[0]];

}

}];

参数说明：

-(void)loadAdWithType:(ADType)adType count:(NSInteger)count size:(CGSize)size block:(ADViewBlock )adViewBlock;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| Type | ADType | 自定义广告的三种类型：  AdTypeCustomBanner = 7,  //自定义原生banner  AdTypeListWord = 8,  //自定义文案信息流广告  AdTypeListBanner = 9  //自定义图文信息流广告 |
| count | NSInteger | 要请求的广告数量，返回的广告数量会小于或等于该数 |
| size | CGSize | 想要生成的banner view的宽高 |
| adViewArray | NSArray | 返回的广告数组，在调用前最好判断数组的个数，再对数组中的view进行操作，首先要设置view的frame值，其次添加view即可。 |

原生banner最常用的有两种：①用于view中 ②用于scrollView中，具体代码实现及效果展示请参考demo。

**2.3.4** **自定义文案信息流广告**



具体效果和代码可运行demo查看，调用：

[[UBKAdLoader new] loadAdWithType:AdTypeListWord count:1 size:CGSizeMake(KWIDTH, 70) block:^(NSArray \*adViewArray) {

if (adViewArray.count) {

[adViewArray[0] setFrame:CGRectMake(0, 100, KWIDTH, 70)];

[self.view addSubview:adViewArray[0]];

}

}];

参数说明：

-(void)loadAdWithType:(ADType)adType count:(NSInteger)count size:(CGSize)size block:(ADViewBlock )adViewBlock;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| Type | ADType | 自定义广告的三种类型：  AdTypeCustomBanner = 7,  //自定义原生banner  AdTypeListWord = 8,  //自定义文案信息流广告  AdTypeListBanner = 9  //自定义图文信息流广告 |
| count | NSInteger | 要请求的广告数量，返回的广告数量会小于或等于该数 |
| size | CGSize | 想要生成的banner view的宽高 |
| adViewArray | NSArray | 返回的广告数组，在调用前最好判断数组的个数，再对数组中的view进行操作，首先要设置view的frame值，其次添加view即可。 |

### 2.3.5 自定义图文信息流广告



[[UBKAdLoader new] loadAdWithType:AdTypeListBanner count:1 size:CGSizeMake(KWIDTH, 160) block:^(NSArray \*adViewArray) {

if (adViewArray.count) {

[adViewArray[0] setFrame:CGRectMake(0, 100, KWIDTH, 160)];

[self.view addSubview:adViewArray[0]];

}

}];

参数说明：

-(void)loadAdWithType:(ADType)adType count:(NSInteger)count size:(CGSize)size block:(ADViewBlock )adViewBlock;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| Type | ADType | 自定义广告的三种类型：  AdTypeCustomBanner = 7,  //自定义原生banner  AdTypeListWord = 8,  //自定义文案信息流广告  AdTypeListBanner = 9  //自定义图文信息流广告 |
| count | NSInteger | 要请求的广告数量，返回的广告数量会小于或等于该数 |
| size | CGSize | 想要生成的banner view的宽高 |
| adViewArray | NSArray | 返回的广告数组，在调用前最好判断数组的个数，再对数组中的view进行操作，首先要设置view的frame值，其次添加view即可。 |

## 2.4 终止SDK

在工程的 AppDelegate 中的以下方法中，终止SDK：

- (void)applicationDidEnterBackground:(UIApplication \*)application {

[UBKAD sdkTerminate];

}

# 3. 设置调试模式

+(void)setDebug:(BOOL)debug;

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **描述** |
| debug | BOOL | YES –使用debug模式，输出日志  NO –使用release模式，没有日志输出 |

在工程的 AppDelegate 中的以下方法中设置，必须在UBKAD初始化之前设置调试模式：

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

{

[UBKAD setDebug:YES];

UBKAD \*ubkad = [[UBKAD alloc] initWithChannel:@"App Stroe" app\_id:00000 app\_key:@" ABCDEFGHIJKLMNOP "];

return YES;

}

# 4. 广告展示回调

广告展示和关闭等操作时，SDK可以通过回调的方式通知调用者，前提是调用者必须遵守协议（<UBKADDelegate>），并设置代理（[UBKAD setADDelegate:self]）然后再实现回调方法。

回调方法如下：

//没有广告

-(void)onNoAD:(NSInteger )errorCode;

//广告被加载

-(void)onADLoaded;

//广告被点击

-(void)onADClicked;

//广告被展示

-(void)onADShowedWithADRequest:(UBKADRequest \*)adRequest adWidth:(float)width adHeight:(float)height;

//广告被关闭

-(void)onADClosedWithADRequest:(UBKADRequest \*)adRequest adWidth:(float)width adHeight:(float)height;

# 5. 联系方式

您在接入过程中遇到任何问题，请即时联系我们：

李伟鹏

QQ ：275138984

手机：15989054826

邮箱：gary@feelwx.com