

目录

目录

版本修订

1.简介

2.接入说明

2.1 支持版本

2.2 SDK 文件

2.3 依赖库

2.4 编译参数设置

2.5 适配iOS9

2.5.1plist中添加支持HTTP请求权限

2.5.2关闭bitcode支持

3 开发示例

3.0 设置参数

3.1 开屏广告

3.2 Banner 广告

3.3 Native广告

版本修订

版本号	更新内容	备注
V1.0.0	初始化版本	
V1.0.3	变更Banner广告和Native广告代理接口，回传itemspaceid，便于app区分	
V1.0.5	优化落地页	
V1.0.6	调整Banner广告缩放模式	详见3.2
V1.0.7	调整落地页更好支持AppStore	
V1.0.8	Native广告调整上报策略	

1.简介

移动广告联盟是搜狐推出的针对开发者的广告变现平台,移动广告联盟SDK用于简化开发者接入难度,降低开发周期。 本文档用于描述接入SDK所需设置，推荐下载SDK及DEMO以便获得更多信息。

2.接入说明

2.1 支持版本

iOS 7.0 +

2.2 SDK 文件

- libSHADMob.a //SDK静态库文件
- STADAsset.bundle //资源文件
- SHADMobConfig.h //配置头文件
- SHADMobOpen.h //开屏广告头文件
- SHADMobBanner.h //Banner广告头文件

- SHADMobNative.h //原生广告头文件

libSHADMob.a 文件包含模拟器，所以体积稍大，但是编译运行后只会对app增加约2M大小

2.3 依赖库

- WebKit.framework
- StoreKit.framework
- libc++.tbd
- libxml2.tbd
- libz.tbd
- libsqlite3.0.tbd
- CoreLocation.framework
- AdSupport.framework
- CoreTelephony.framework
- SystemConfiguration.framework
- Foundation.framework
- UIKit.framework

2.4 编译参数设置

targets->BuildSettings->Other Linker Flags ->添加 -all_load -ObjC

▼ Linking

Setting	SHADMobDEMO
Link With Standard Libraries	Yes ▾
Other Linker Flags	-all_load -ObjC
Quote Linker Arguments	Yes ▾

2.5 适配iOS9

2.5.1plist中添加支持HTTP请求权限

在NSAppTransportSecurity下设置NSAllowsArbitraryLoads为true

```
<key>NSAppTransportSecurity</key><dict>

<key>NSAllowsArbitraryLoads</key>

<true/></dict>
```

2.5.2 关闭bitcode支持

ENABLE_BITCODE=NO. SDK不支持BITCODE

Basic All Combined Levels +

Q bitcode

▼ Build Options

Setting SHADMobDEMO

Enable Bitcode No

3 开发示例

3.0 设置参数

```
[SHADMobConfig getVersion]; //获取SDK版本号

/*
*设置是否调试模式
*调试模式使用测试服务器地址，并输出log信息，app正式上线时，请切换至正式模式，设置为NO
*/
[SHADMobConfig setDebugMode:NO];
```

3.1 开屏广告

在AppDelegate.m中添加代理对象

```
@interface AppDelegate () <SHADMobOpenDelegate>

@end
```

添加开屏广告初始化代码

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    SHADMobOpen *open = [[SHADMobOpen alloc] initWithAppkey:@"12345"
developerId:@"67890" itemspaceId:@"12848"]; // 设置应用id,开发者id,广告位id

    open.delegate = self; // 设置代理

    open.canClick = YES ; //开屏广告是否可点击, 推荐开启 (可选)

    open.backgroundColor = [UIColor blackColor]; //根据app主题设置背景色
(可选)

    [open loadAndDisplayUsingKeyWindow:self.window]; //设置keyWindow

    return YES;
}
```

开屏广告包含3秒自动倒计时, 可点击跳转落地页, 3秒后自动消隐。

添加delegate回调函数

```
#pragma mark - SHADMobOpenDelegate

- (void)openPresentSuccess{
    NSLog(@"%s",__FUNCTION__);
}

- (void)openPresentFail:(SHADMobOpenErrorType)error{
    NSLog(@"%s code=%ld",__FUNCTION__,(long)error);
}

- (void)openDidDismissScreen{
    NSLog(@"%s",__FUNCTION__);
}

- (void)openImpression{
    NSLog(@"%s",__FUNCTION__);
}

- (void)openClicked{
    NSLog(@"%s",__FUNCTION__);
}

- (void)openLoadingPageDidDismissScreen{
    NSLog(@"%s",__FUNCTION__);
}
```

3.2 Banner 广告

添加代理对象

```
@interface ViewController () <SHADMobBannerViewDelegate>
{
    SHADMobBannerView *bannerView;
}
@end
```

添加Banner初始化代码

```

bannerView = [[SHADMobBannerView
alloc]initWithFrame:SHAD_BANNER_SIZE_640_100 appkey:@"00000"
developerId:@"67890" itemspaceId:@"12849"];
// 设置应用id,开发者id,广告位id
bannerView.delegate = self; // 设置代理
bannerView.currentViewController = self; //设置落地页展示ViewController
bannerView.showCloseBtn = NO; //是否显示关闭按钮
[bannerView requestAndShow];//发起请求并展示广告

```

添加delegate回调函数

注意：**sdk**返回的广告 **UIView**对象，应添加到当前**app**中一个独立的指定大小的 **UIView**父窗口中，父窗口的展示尺寸要符合广告位比例，并且满足**app**展示美观需要，广告**sdk**将根据父窗口尺寸和广告位尺寸适当缩放大小；推荐在 **bannerViewDidPresentScreenWithItemspaceid:**函数回调时将广告对象添加到父窗口。

```

#pragma mark - Banner Delegate

- (void)bannerViewDataDidReceivedWithItemspaceid:(NSString
*)itemspaceId{
    NSLog(@"%s itemspaceId = %@",__FUNCTION__,itemspaceId);
    [rootView addSubview:bannerView];//向展示广告对象添加返回的Banner广告
    View对象
}

/**
 * banner广告展示成功回调
 * 推荐在此时将返回的广告对象View添加到app中容纳广告的父窗口对象中，广告sdk将
    根据广告位尺寸和父窗口尺寸调整广告大小
 * 推荐父窗口大小与广告位大小等比例，以避免拉伸或压缩
 *
 * @param itemspaceId 广告位id
 */
- (void)bannerViewDidPresentScreenWithItemspaceid:(NSString
*)itemspaceId{
    NSLog(@"%s itemspaceId = %@",__FUNCTION__,itemspaceId);
    [rootView addSubview:bannerView];
}

- (void)bannerViewFail:(SHADMobBannerErrorType)error andItemspaceid:
(NSString*)itemspaceId{

```

```

        NSLog(@"%s error=%ld, itemspaceid= %@", __FUNCTION__,
(long)error, itemspaceId);
    }

    - (void)bannerViewDidDismissScreenWithItemspaceid:(NSString
*)itemspaceId{
        NSLog(@"%s itemspaceId = %@", __FUNCTION__, itemspaceId);
    }

    - (void)bannerViewImpressionWithItemspaceid:(NSString *)itemspaceId{
        NSLog(@"%s itemspaceId = %@", __FUNCTION__, itemspaceId);
    }

    - (void)bannerViewClickedWithItemspaceid:(NSString *)itemspaceId{
        NSLog(@"%s itemspaceId = %@", __FUNCTION__, itemspaceId);
    }

    - (void)bannerViewWillCloseWithItemspaceid:(NSString *)itemspaceId{
        NSLog(@"%s itemspaceId = %@", __FUNCTION__, itemspaceId);
    }

```

3.3 Native广告

添加代理对象

```

@interface ViewController () <SHADMobNativeDelegate>
{
    SHADMobNative *nativeObj;
}
@end

```

添加Native广告初始化代码

```

        nativeObj = [[SHADMobNative
alloc] initWithFrame:SHAD_NATIVE_SIZE_360_234 appId:@"100"
developerId:@"1000" itemspaceId:@"12873"];
// 设置应用id,开发者id,广告位id

        nativeObj.delegate = self; // 设置代理
        nativeObj.currentViewController = self; // 设置落地页展示ViewController
        [nativeObj requestDataWithCount:1]; // 发起请求并传入请求广告数量

```

添加delegate回调函数

```
#pragma mark - NativeAd Delegate
- (void)nativeAdLoadSuccess:(NSArray *)nativeAddataArray andItemspaceId:
(NSString *)itemspaceId{
    //请求成功后返回广告对象数组
    if (nativeAddataArray && nativeAddataArray.count>0) {
        NSLog(@"nativeAD itemspaceId=%@",itemspaceId);
        //如需展示多个广告，请遍历数组对象，此处仅演示获取第一个对象内容
        id currentAdObj = nativeAddataArray[0];
        //转换成SHADMobNativeData对象
        if ([currentAdObj isKindOfClass:[SHADMobNativeData class]]) {
            data = currentAdObj;
            //获取字典内容
            nativeTitle.text = data.properties[@"title"];
            nativeImagePath.text = data.properties[@"image"];
            //简单展示
            NSURL *url = [NSURL
URLWithString:data.properties[@"image"]];
            UIImage *image = [UIImage imageWithData:[NSData
dataWithContentsOfURL:url]];
            nativeImage.image = image;
            nativeImage.contentMode = UIViewContentModeScaleAspectFit;
        }
    }
}

- (void)nativeAdLoadFail:(SHADMobNativeErrorType)type andItemspaceid:
(NSString *)itemspaceId{
    NSLog(@"%s code=%ld,itemspaceId=%@",__FUNCTION__,
(long)type,itemspaceId);
}

- (void)nativeAdLoadingPageWillPresentSrceenWithItemspaceid:(NSString
*)itemspaceId{
    NSLog(@"%s itemspaceId= %@",__FUNCTION__,itemspaceId);
}

- (void)nativeAdLoadingPageClosedWithItemspaceid:(NSString
*)itemspaceId{
    NSLog(@"%s itemspaceId= %@",__FUNCTION__,itemspaceId);
}
```

添加点击和曝光处理函数

```
/**
 * native广告被点击后需要调用sdk方法
 *
 * @param sender UIButton对象
 */
- (IBAction)nativeAdClicked:(UIButton *)sender {
    [nativeObj nativeAdClick:data];
}

/**
 * native广告展示完成后需要调用sdk方法
 *
 * @param sender UIButton对象
 */
- (IBAction)nativeAdImpression:(UIButton *)sender {
    [nativeObj nativeAdDidShow:data];
}
```