深圳市蓝莓互娱科技有限公司

蓝莓聚合SDK集成文档(iOS)

(V4.0-20180717)

在您阅读此文档时，我们假定您已经具备了基础的 iOS 应用开发经验，并能够理解相关基础概念。

本版本SDK中获取了广告标识IDFA,为避免App提交审核时出现不必要麻烦，请选择：是。选项建议勾选2，4选项即可。

如果项目之前已经接过我们 SDK，只是从v1.0 /v2.0升级到 v4.0的，

1、添加路径 LmMobSDK/frameworks/ 里面的 GoogleMobileAds.framework

2、更换 libLmMobSDK.a

3、更换 LmMobSDK.h

4、TARGETS >Build Phases >Link Binary with Librairies > + >添加 libresolv.9.tbd

5、在项目的info.plist中添加: Privacy - Location Always and When In Use Usage Description，类型为字符类型。

即可，调用不用作任何修改。

如果项目是首次接入我们SDK，请详细阅读下文：

一、联系我们获取最新版本的 LmMobSDK-Demo

解压 Demo 后把目录 LmMobSDK 整个文件夹拷贝到你们项目，包括:

1、frameworks

2、Assets

3、libLmMobSDK.a

4、LmMobSDK.h

二、项目环境配置

SDK环境配置请按以下步骤进行：

第一步：

将下面的库和框架添加到

TARGETS >Build Phases >Link Binary with Librairies > + >搜索

1、libc++.tbd

2、libsqlite3.tbd

3、libxml2.tbd

4、libz.tbd
5、Accelerate.framework
6、AdSupport.framework
7、AudioToolbox.framework
8、AVFoundation.framework
9、CFNetwork.framework
10、CoreGraphics.framework
11、CoreLocation.framework
12、CoreTelephony.framework
13、CoreMedia.framework
14、CoreMotion.framework
15、EventKit.framework
16、Foundation.framework
17、GLKit.framework
18、iAd.framework
19、ImageIO.framework
20、MediaPlayer.framework
21、MessageUI.framework
22、MobileCoreServices.framework
23、QuartzCore.framework
24、Security.framework
25、Social.framework
26、StoreKit.framework
27、SystemConfiguration.framework
28、WatchConnectivity.framework
29、WebKit.framework
30、JavaScriptCore.framework
31、UIKit.framework
32、libresolv.9.tbd
(原项目如已添加，则略过)

第二步:
TARGETS >Capabilities >Keychain Sharing > 开关改为 "ON"

第三步:
TARGETS > Build Settings > (Linking >) Other Linker Flags > + > 添加参数：
1、-ObjC (或者-all_load)
2、~~-fobjc-arc~~

第四步：

~~TARGETS > Build Settings > Enable Bitcode 设置为 "NO"~~

第五步：

申请相关权限

广告商 Adcolony 提供的广告 SDK 依赖 EventKit.framework ，根据苹果官方文档的说明，自 iOS 10.0 之后需要申请相关的权限否则会导致程序的崩溃。所以需要在 Info.plist 文件中加入如下配置项

```
<key>NSCalendarsUsageDescription</key>
<string>Some ad content may create a calendar event</string>
<key>NSCameraUsageDescription</key>
<string>Some ad content may access camera to take picture.</string>
<key>NSPhotoLibraryUsageDescription</key>
<string>Some ad content may require access to the photo library.</string>
<key>NSMotionUsageDescription</key>
<string>Some ad content may require access to accelerometer for interactive ad experience.</string>
```

（V4.0新增）

因SDK本身获取了地理位置信息,请在项目的info.plist中添加

Privacy - Location Always and When In Use Usage Description

```
<key>NSLocationWhenInUseUsageDescription</key>
<string></string>
```

三、SDK初始化

在 AppDelegate.m 文件//或者游戏入口函数中调用初始化代码

引用 #import "LmMobSDK.h" 头文件

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


    //    开发者向SDK方申请获取的APP 的 key
        NSString * appkey = @"eac6500c-33a6-4d8d-afa4-def79935f9c1";
//请填写实际获取的key
        [LmMobSDK initialize: appkey callback:^(BOOL success, NSDictionary * _Nullable result) {



    }];


    //setDebugMode指定是否调试模式，调试模式下会打印log
```

```
    [LmMobSDK setDebugMode:YES];

    return YES;
}
```

appkey 是开发者key，需要联系商务获取，并填入，是区分不同开发者收益的唯一标识。

四、SDK调视频广告播放

引用 #import "LmMobSDK.h" 头文件,遵守代理 <LmMobSDKPreloadDelegate,LmMobSDKADDelegate,LmMobSDKADRewardDelegate>

1). 判断是否准备好

播放激励视频广告前，请先确认是否可以播放，即视频广告是否已经准备 好，有两种方法可以获取这个状态，

第一种方式:通过代理回调方式(被动)

第1步：设置好预加载代理，当视频准备好后，SDK 会立即通过代理函数回调:

<LmMobSDKPreloadDelegate>

第2步：设置预加载代理

```
//设置预加载代理
   [LmMobSDK setPreloadADDelegate:self];
```

第3步：代理回调方法

```
@protocol LmMobSDKPreloadDelegate <NSObject>
@optional
//一、初始化回调
/**
 * 1、当SDK有广告主的广告视频预加载成功时，会调用此代理。
 * @param adver   adver isEqualToString: @"LmMobSDK" 才是本SDK预加载成功，且并不代表有视频可以播放了
 */
- (void) lmMobSDK:(NSString * _Nullable)adver
  PreloadSuccess:(NSString * _Nullable)result;

/**
 * 2、当SDK预加载发生错误时，会调用此代理。
 * @param adver   adver isEqualToString: @"LmMobSDK" 才是本SDK预加载失败
```

```
 *  @param result 回调SDK初始化失败的原因
 */
- (void)lmMobSDK:(NSString * _Nullable)adver PreloadFailed:(NSString *_Nullable)result WithError:(NSError* _Nullable) error

//3、视频广告就绪回调
- (void)lmMobSDKVideoLoaded:(NSString * _Nullable) result


@end
```

第二种方式:调用 isReady 方法判断(主动)

```
//获取当前是否有可播放的视频广告
// [LmMobSDK isReady];
```

2). 播放视频

第一步，在广告已准备好的情况下，在要播放激励视频的地方调用播放
方法。

```
/**
 *  播放激励视频
 *  @param  scene 场景-->预留参数,现在传空字符串@""
 *  @param  view  在哪个控制器上播放
 *  @param delegate  播放代理
 */

[LmMobSDK showAD:@"" WithViewController:self delegate:self];
```

第二步，设置播放接口回调代理
<LmMobSDKADDelegate>

```
@protocol LmMobSDKADDelegate <NSObject>
@optional
//二、播放回调
 //  5、播放成功回调
```

```
- (void)lmMobSDKShowSuccess:(NSString* _Nonnull)result;

// 6、播放失败回调
- (void)lmMobSDKShowFailed:(NSString* _Nonnull)result
WithError:(NSError* _Nonnull)error;

// 7、完成播放回调
- (void)lmMobSDKADComplete:(NSString* _Nonnull)result;

// 8、点击跳转（App Store）回调
- (void)lmMobSDKADClick:(NSString* _Nonnull)result;
-
// 9、关闭广告回调
- (void)lmMobSDKADClose:(NSString* _Nonnull)result;
```

第三步，设置奖励接口回调代理

```
<LmMobSDKPreloadDelegate>

//设置奖励回调代理
[LmMobSDK setADRewardDelegate:self];


@protocol LmMobSDKPreloadDelegate <NSObject>
@optional
//三、奖励回调
 // 10、奖励广告条件达成回调
- (void)lmMobSDKADAwardSuccess:(NSString* _Nonnull)result;

// 11、奖励广告条件未达成回调
- (void)lmMobSDKADAwardFailed:(NSString* _Nonnull)result
WithError:(NSError* _Nonnull)error;
```