



mobiSage_Android_SDK

[使用说明]

SDK Version: mobiSage_Android_SDK_7.2.8

2016-02

目录

- 1. SDK 主要功能指南..... 3
 - 1.1 导入 SDK 3
 - 1.2 配置 AndroidManifest.xml 3
 - 1.3 申请及设置 Publish ID 4
 - 1.4 申请 slotToken 5
 - 1.5 横幅广告 5
 - 1.6 信息流广告 8
 - 1.7 插屏广告 13
- 2. 使用建议..... 15



1. SDK 主要功能指南

1.1 导入 SDK

访问 <http://mobi.adsage.cn/SDK/androidDownload> , 下载 SDK 包 , 解压后找到 mobisageSDK.jar。请按照如下步骤在工程的 Java Build Path 中引入 mobisageSDK.jar :

方法一 : Eclipse ADT 17 以下版本 , Eclipse 右键点击工程根目录 , 选择 Properties -> Java Build Path -> Libraries , 然后点击 Add External JARs... 选择指向 jar 的路径 , 点击 OK , 即导入成功

方法二 : Eclipse ADT 17 及以上版本用户 , 请在工程目录下建一个文件夹 libs , 把 jar 包直接拷贝到这个文件夹下 , 再在 Eclipse 里面刷新一下工程就好了。详情请参考

<http://tools.android.com/recent/dealingwithdependenciesinandroidprojects>

1.2 配置 AndroidManifest.xml

在工程的 AndroidManifest.xml 中 <manifest> 区域配置 Android 权限信息 , 如下所示 :

```
<manifest.....>
<application .....>
.....
<activity ...../>
<!-- 广告点击后 , 打开 Landing Site 时使用的 Activity , 必选 -->
<activity
    android:name="com.mobisage.android.MobiSageActivity"
    android:configChanges="keyboardHidden|orientation|screenSize" >
</activity>
<!-- 应用发布渠道 , 可选 -->
<meta-data android:name="Mobisage_channel" android:value="your channel"/>
</application>
<!-- sdk 所需的用户权限 , 必选 -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
</manifest>

```

如果你需要在机锋应用市场发布，请将渠道标识 “your channel” 更改为 “gfan”。

注意，错误的渠道标识可能会影响到市场的发布，请正确填写。

1.3 申请及设置 Publish ID

➤ 申请 Publish ID

访问 <http://mobi.adsage.cn>，注册用户并创建 App，将获取到每个 App 的 Publish ID，一个应用程序可以嵌入多种形式的广告，但需要使用同一个 Publish ID；一个 Publish ID 只能绑定一个应用程序；如需发布多个程序，请分别获取各自的 PublishID。

如下图所示：



➤ 设置 Publish ID 和状态标示字段

状态标识字段是为了标识不同渠道不同版本的不同审核状态而设置的。您可以通过 SDK 为不同渠道不同版本的 APP 传入唯一的状态标识字段，并且在官网平台上设置不同的状态。开发者在平台上登陆并且创建 InApp 后，可以在我的 InApp 界面点击“设置状态标识”进入编辑状态标识字段界面。建议格式

为：系统_渠道_版本号，例如"AnZhi_v1.1.3.1"。字段为选填字段，但为了最大限度保证您的应用通过商店的审核请您尽量填写，并谨慎更改相应的状态！在应用的启动 Activity 或 Application 的 onCreate 方法中设置 PublishID 和状态标示字段。

状态标示字段为选填项,务必保证设置 PublishID 操作在创建广告之前。

如下图所示：

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    MobiSageManager.getInstance().initMobiSageManager(context, "请传入申请的publishId",
        "AnZhi_v1.1.3.1");
}
```

1.4 申请 slotToken

申请 PublishID 后,需为要展示的广告类型申请 slotToken,如需展示 banner,需申请 banner 的 slotToken,该 slotToken 只能创建 banner 类型的广告位.一个 slotToken 可以创建多个相同类型广告位.

特别注意：
设置完广告位信息后，请用生成的广告位ID去配置SDK。如果您的应用没有创建广告位，将接收不到艾德思奇广告平台的广告。



名称	类型	状态	广告位ID	创建时间	操作
植入式广告	信息流广告位		bWxtvgL64HddeEs36v62UOFD	2014-06-23 13:40:27	编辑 删除
开屏	开屏广告位		hYSFVuoSCJ+1kKPFahZeuAmh	2014-06-23 13:40:08	编辑 删除
全屏广告	全屏广告位		l5aXRPgAGo2ngrHNEARMqhuw	2014-06-23 13:39:47	编辑 删除
横幅	横幅广告位		Y2JjsAz07nlTdkU55PC4Xu9F	2014-06-23 13:39:14	编辑 删除

当前显示第 1 条 到 第 4 条数据，总记录数：4 条







1.5 横幅广告

➤ 创建广告

在应用的显示广告 Activity 的 layout 文件对应位置添加如下 xml 代码：

```
<LinearLayout android:id="@+id/ad_container"
    android:layout_centerHorizontal="true"
    android:layout_width="fill_parent"
    android:orientation="vertical"
    android:layout_height="wrap_content"/>
```

在 Activity 中获得 ad_container，然后添加 MobiSageAdBanner 到 ad_container 中，示例代码：

```
//获取广告容器
LinearLayout ad_container = (LinearLayout) findViewById(R.id.ad_container);
//实例化广告（广告尺寸自适应不同分辨率设备），传入相应类型 slottoken，true 为大尺寸，false 为小尺寸
MobiSageAdBanner adv = new MobiSageAdBanner(context, "请传入申请的 slottoken", true);
//设置轮播时间
adv.setAdRefreshInterval(MobiSageAdBanner.AD_REFRESH_30);
//设置轮播动画
adv.setAnimeType(MobiSageAdBanner.ANIME_CUBET2B);
//设置监听
adv.setMobiSageAdBannerListener(mListener);
//将广告加入容器
ad_container.addView(adv);
```

➤ 销毁广告

广告不再使用时，通过下面的方法将其销毁：

```
if(adv != null){
    //销毁广告
    adv.destroyAdView ();
    adv = null;
}
```

➤ 广告通知

横幅广告提供了五个广告相关的通知，如下所示：

```
private MobiSageAdBannerListener mListener = new MobiSageAdBannerListener() {
```

```
//横幅广告请求成功时，此接口被回调，横幅广告可以展示
@Override
public void onMobiSageBannerSuccess(MobiSageAdBanner adView) {

}

//横幅广告展示或轮播成功时，此接口被回调，多用于聚合中统计广告展示数
@Override
public void onMobiSageBannerShow (MobiSageAdBanner adView) {

}

//横幅广告被点击时，此接口被回调，多用于聚合中统计广告点击数
@Override
public void onMobiSageBannerClick (MobiSageAdBanner adView) {

}

//横幅广告展示或轮播失败时，此接口被回调，多用于聚合中切换广告平台,msg 为错误信息
@Override
public void onMobiSageBannerError (MobiSageAdBanner adView, String msg) {

}
};
```

1.6 信息流广告

信息流广告特点是可以跟应用进行紧密结合，自定义广告 UI 展现和广告元素组合，适用于信息流、图片流等等更广泛应用进行集成。信息流广告分为两种模式：

模板模式：SDK 提供广告 view，开发者需要把广告 view 添加到需要展示广告的地方；

自定义模式：SDK 提供广告元素，开发者获取到广告图片、标题等各种元素进行组合，拼装成自定义的 view 进行展示。

- 创建广告
- 通过调用下面的方法来实例化信息流广告：

```
// 实例化广告,传入上下文,广告位 slottoken,期望宽和期望高
MobiSageAdNativeFactory mNativeGroup = new MobiSageAdNativeFactory(context,"请传入申请的
slottoken", 300, 170)
//设置展示模式,展示有四种模式:CLASSIC,IMG,SIMPLE1,SIMPLE2;开发者自行选择
mNativeGroup.SetModuleMode(MobiSageAdNativeFactory.CLASSIC);
//设置模板模式,true是模板模式,false是自定义模式
mMobiSageAdNativeFactory.setIsModule(true);
//设置广告工厂监听
mNativeGroup.setMobiSageAdNativeGroupListener(new MobiSageAdNativeFactoryListener() {
    //广告请求成功返回时通知
    @Override
    public void onMobiSageNativeGroupSuccess(MobiSageAdNativeFactory
mobiSageAdNativeFactory){
        //自定义模式在此获取广告元素信息,详见下面的“获取广告信息”章节
        getADContent(mobiSageAdNativeFactory);
    }
    //广告请求失败时通知,message为失败信息
    @Override
    public void onMobiSageNativeGroupError(MobiSageAdNativeFactory mobiSageAdNativeFactory,
String message) {
    }
    //广告加载成功时调用
    @Override
    public void onMobiSageNativeGroupLoadSuccess(MobiSageAdNativeFactory
mobiSageAdNativeFactory){
        //模板模式在此获取广告对象,把广告添加到view中
        ArrayList<MobiSageAdNative> mAdNatives = mobiSageAdNativeFactory.getAdNatives();
        for (int i = 0; i < mAdNatives.size(); i++) {
            MobiSageAdNative adNative= mAdNatives.get(i);
            //在此展示广告
        }
    }
});
```



```
//设置请求广告个数，一次最多请求10个  
mNativeGroup.LoadAds(5);
```

注意：根据宽高请求广告(宽高比在 0.5-2 之间为有效 size)，我们会根据有效 size 自适应填充广告内容。创建信息流广告传入的宽高与系统原生控件一样，即实际像素。如果创建一个宽为 400 的广告视图，那么在任何机型上都是 400 像素宽，所以开发者创建广告的时候为了适配不同屏幕，需根据屏幕密度动态计算广告尺寸。

➤ 设置单个信息流广告的展示、点击监听

通过工厂获得单个信息流广告 (MobiSageAdNative 对象) 后，可对其设置展示和点击的监听，如下所示：

```
private MobiSageAdNativeListener mListener = new MobiSageAdNativeListener () {  
    //广告点击时调用  
    void onMobiSageNativeClick(MobiSageAdNative adView){  
    }  
    //广告展示成功时调用  
    void onMobiSageNativeShow(MobiSageAdNative adView){  
    }  
};  
adNative.setMobiSageAdNativeListener(mListener);
```

➤ 获取广告信息

信息流广告与普通横幅广告的差别除了可自定义尺寸外，主要的差别在于信息流广告会返回广告本身的元素信息。目前包含的信息有：广告宽度，广告高度(注：不是请求的宽高而是返回广告的宽高，服务器会根据开发者请求的宽高返回接近的尺寸，此处获得的即为实际尺寸)，广告标题，广告 logo 网址(为一张图片)，image，广告描述信息，广告星级，广告包大小，有多少人在玩(使用)人数，评论数。

开发者可以将广告 view 与广告信息拼装成自己喜欢的形式展示出来(使用方法参考 Sample)：

```
Public void getADContent(MobiSageAdNativeFactory mobiSageAdNativeFactory){  
    ArrayList<MobiSageAdNative> mAdNatives = mobiSageAdNativeFactory.getAdNatives();  
    for(int i=0,i< mAdNatives.size(),i++){
```

```

MobiSageAdNative adNative = mAdNatives.get(i); //获取单个广告数据

int width = adNative.getAdWidth(); //获取广告实际宽度

int height = adNative.getAdHeight(); //获取广告实际高度

String title = adNative.getTitle(); //广告标题, String 类型

String image = adNative.getImage(); //广告展示图片链接地址, String 类型

String Logo = adNative.getLogo(); //广告对应应用logo图片链接地址, String 类型

String desc = adNative.getDesc(); //广告描述, String 类型

double star = adNative.getStar(); //广告星级, double 类型

int people = adNative.getPeople(); //在线人数 int 类型。

int status = adNative.getState(); //广告状态 , int 类型。

int ratingnum = adNative.getRatingnum(); //广告评论数, int 类型。

int id = adNative.getAdId(); //广告唯一标识 , int 类型。

}
}
    
```

广告状态指当前广告下载状态，开发者在自定义 UI 时，可进行广告状态引导，状态信息请参考以下表格：

状态码	含义	建议开发者按钮显示文案
0	下载失败	点击重试
1	未下载	下载
2	下载中	下载中
3	下载完成	点击安装
4	安装完成	点击启动

➤ 展示广告

模板模式：

直接将广告视图放入父 view 中即可

```

FrameLayout frameLayout = new FrameLayout();
    
```

```
frameLayout.addView(adNative);
```

自定义模式:

界面绘制好后,需要调用 `attachToView` 方法,否则广告不会展示,也不可点击

```
adNative.attachToView(view);//view 为开发者通过广告图片、标题等元素组合拼装的自定义 view
```

注意:如果是开发者自己根据元素绘制的界面一定记得调用 `attachToView` 方法,否则不展示也不可点击.

➤ 销毁广告

广告不再使用时,通过下面的方法将其销毁():

```
//模板模式
if(mNativeGroup!= null){
    //销毁广告
    mNativeGroup.destroyAdView();
    mNativeGroup = null;
}
//自定义模式
if(adNative!= null){
    //销毁广告
    adNative.detachFromView();
    adNative = null;
}
```

➤ 广告分类标签 (非必须设置)

在广告请求前输入期望请求的广告分类,地理位置,及 app 信息

可在通过设置,来精确投放广告.

第一个标签、请求广告类型的标签。

```
//请求广告类型标签
String[] tag = {"ts", "ss", "aa"};
//标志便签是正选还是反选, 0是正选, 1是反选。
int mInver = 0;
mNativeGroup.setTags(mInver, strs);
```

第二个标签、类型为 HashMap,应用和用户的信息

结构如下:

```
"customtag":{
  "appinfo":{
    "content":
      ["aa","bb"],
    "channel":
      [
        {"s":"手机",
         "f":"新闻",
         "t":"android"},

        {"s":"平板",
         "f":"科技",
         "t":"aPad"}]},
  "usertag":{
    "lon":100,
    "bcircle":
      ["abc","efg"],
    "address":"xx 路 xx 号",
    "province":"北京",
    "addtag":
      ["大学",
       "教育"],
    "addname":"北京大学",
    "lat":100,
    "country":"中国",
    "city":"北京"}}
```

代码如下:

```
HashMap<String, Object> customtag = new HashMap<String, Object>();
HashMap<String, Object> usertag = new HashMap<String, Object>();
HashMap<String, Object> appinfo = new HashMap<String, Object>();
usertag.put("country", "中国");
usertag.put("province", "北京");
usertag.put("city", "北京");
```

```
usertag.put("address", "xx路xx号");
usertag.put("addname", "北京大学");
usertag.put("adddtag", "[\"大学\", \"教育\"]");
usertag.put("bcircle", "[\"abc\", \"efg\"]");
usertag.put("lon", 100);
usertag.put("lat", 100);
JSONArray channel = new JSONArray();
JSONObject cone = new JSONObject();
JSONObject ctwo = new JSONObject();
try {
    cone.put("f", "新闻");
    cone.put("s", "手机");
    cone.put("t", "android");
    ctwo.put("f", "科技");
    ctwo.put("s", "平板");
    ctwo.put("t", "aPad");
} catch (JSONException e) {
    e.printStackTrace();
}
channel.put(cone);
channel.put(ctwo);
appinfo.put("channel", channel);
JSONArray content = new JSONArray();
content.put("aa");
content.put("bb");
appinfo.put("content", content);
customtag.put("usertag", usertag);
        customtag.put("appinfo", appinfo);

mNativeGroup.setCustomtag(customtag);
```

详情见 sample 信息流经典类型。

注意：这些放在mNativeGroup.LoadAds(5);之前

1.7 插屏广告

➤ 创建广告

//实例化广告，传入相应类型 slottoken,广告尺寸自适应不同分辨率设备。

```
MobiSageAdPoster adv = new MobiSageAdPoster(activity, "请传入申请的 slottoken",
        MobiSageAdPoster.SIZE_300X250);
```

插屏广告支持的尺寸及枚举值见下表：

广告尺寸	枚举值
300X250 (推荐)	MobiSageAdPoster.SIZE_300X250
320X480	MobiSageAdPoster.SIZE_FULL_SCREEN

插屏的尺寸还可以在官网上动态设置,如果官网设置了,则以官网的设置为准.



➤ 设置广告监听

插屏广告提供了四个广告相关的通知,如下所示:

```
private MobiSageAdPosterListener mListener = new MobiSageAdPosterListener() {
    //插屏广告预加载成功时,此接口被回调,建议在收到此通知后,展示广告,否则影响广告展示的成功率
    @Override
    public void onMobiSagePosterPreShow () {
        adv.show();
    }

    //插屏广告被点击时,此接口被回调,多用于聚合中统计广告点击数
    @Override
    public void onMobiSagePosterClick () {
    }

    //插屏广告请求失败时,此接口被回调,多用于聚合中切换广告平台,msg为错误信息
    @Override
    public void onMobiSagePosterError (String msg) {
    }

    //插屏广告被关闭或被销毁时,此接口被回调,一般情况下不用处理
}
```

```
@Override
public void onMobiSagePosterClose (){
}
};
```

➤ 展示广告

```
//建议在 onMobiSagePosterPreShow 回调时展示
adv.show();
```

2. 使用建议

➤ **运行及开发环境：**

最低环境为 Android Api Level 9

编译环境为 Android Api Level 19

➤ **关于代码混淆：**

由于 mobiSage SDK 已经过混淆，禁止再次混淆。在对应用进行混淆时，请参考加入以下混淆代码：

如果使用 android studio, 去掉 `-libraryjars libs/mobisageSDK.jar`

```
-optimizations !code/simplification/arithmetic,!field/*,!class/merging/*,!code/allocation/variable
-dontoptimize
-dontwarn
-libraryjars libs/mobisageSDK.jar
-keep class com.mobisage.*.** {*; }
-keep interface com.mobisage.*.** {*; }
```

➤ **关于横幅广告的使用：**

横幅广告所在的 activity 当被遮挡或隐藏时，需将横幅广告释放掉，以防过多的无效广告请求和展示，影响广告点击率和收益。

➤ **关于插屏广告的使用：**

插屏广告支持广告的预加载，在广告被实例化时，会启动广告请求缓存线程，因此建议将插屏广告的实例化前置，这样在预加载成功的方法被回调时，调用广告的展示方法，可以提高广告的填充率。插屏广告在被用户关闭后，插屏广告会自动释放，无需在应用中对其再次进行释放。

联系方式：（参见 <http://mobi.adsage.cn/>）

邮箱：mobi@adsage.com

电话：010-62662626-8294

客服：QQ：1497071580