Nativedynamics Android Ad SDK Integration Documentation

Date	Version	Description	Developer
2017.04.26	V2.2.5	Improve Appwall	卢红伍
2017.07.04	V2.2.6	Support native AD	卢红伍

目录

1. 简介 3
1.1 SDK 功能介绍 3
1.2 兼容性 3
2. 集成准备 3
2.1 APP ID 3
2.2 Unit ID 3
3. 初始化 SDK 3
3.1 代码混淆配置： 4
3.2 初始化 5
3.3 AndroidMenifest.xml 配置 5
3.4 添加权限（可选） 5
4. 广告墙接入 6
4.1 配置 6
4.2 预加载应用墙数据 10
4.3 启动 market 页面 12
5. 原生广告接入 13
5.1 接入流程 13

1. 简介

1.1 SDK 功能介绍

Natividynamics ADSDK 能够提供以下功能：

- AppWall（广告墙）
- Native（原生广告)

1.2 兼容性

支持 Android 2.3（API level 9）及以上系统

2. 集成准备

2.1 APP ID

确保您已注册 Natividynamics 账号且已有可用的 APP ID 用来展示广告

2.2 Unit ID

在 Natividynamics 创建一个广告位后，系统会自动生成 Unit id

3. 初始化 SDK

Natividynamics SDK 可以从 natividynamics 运营获取，SDK 包含多个 aar 包

aar 包名	作用	是否必须
anative_common.aar	公共基础包	是
anative_appwall.aar	appwall 广告的功能包	否
anative_native.aar	native 广告功能包	否

将 aar 文件保存到本地工程中:

- 将所需要的 aar 包拷贝到项目的 libs 文件夹中，并作为项目的依赖库
- 在项目的 build.gradle 文件中的 dependencies 添加：

compile 'com.google.code.gson:gson:2.8.0' // 必须

compile(name: anative_common, ext: 'aar') // 必须

compile(name: anative_appwall, ext: 'aar') // 可选，用于 广告墙

compile(name: anative _native, ext: 'aar') // 可选，用于原生广告

compile 'com.google.android.gms:play-services-ads:8.4.0' // Admob 广告，可选，用于 native 广告

compile 'com.facebook.android:audience-network-sdk:4.23.0' // Fb 广告，

可选，用于 native 广告，

3.1 代码混淆配置：

在您的 proguard 文件中根据所集成的包来添加：

```
#  FB 若没加，可不添加
-keep class com.facebook.** {*;}
-dontwarn com.facebook.**

# Admob 若没加，可不添加
-keep class com.google.android.gms.** {*;}
-dontwarn com.google.android.gms.**
```

3.2 初始化

在 Application 的 onCreate 里面调用该初始化方法:

```
Adsdk.initialize(Context context, String appId);
```

参数说明：

- context: Application 上下文
- appId: 申请的应用 ID

3.3 AndroidMenifest.xml 配置

若集成了 paly-services-ads, 请在 AndroidMenifest.xml 里面添加：

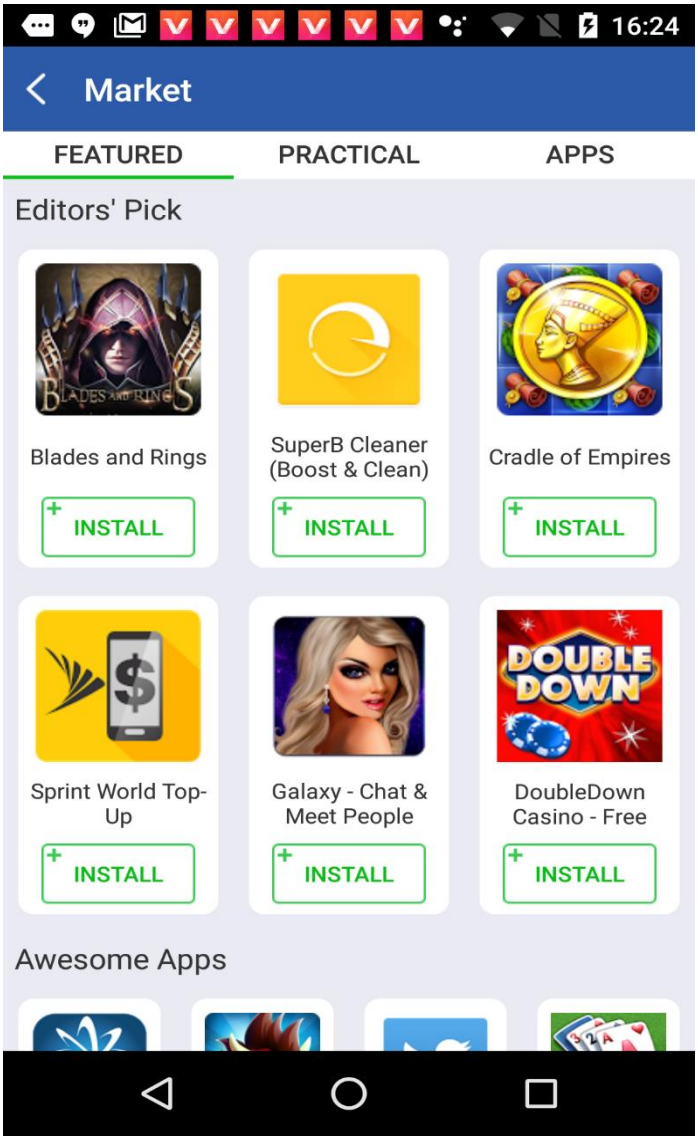
```
<activity
android:name="com.google.android.gms.ads.AdActivity"
android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|smallestScreenSize"
android:theme="@android:style/Theme.Translucent" />
<meta-data
android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
```

3.4 添加权限（可选）

部分广告主需要 SDK 回传 device id，建议开发者添加 android.persmission.READ_PHONE_STATE 便于 SDK 获取设备 id。

4. 广告墙接入

App Market 接口能够通过简单的设定，获取一个完整的广告商店的 Activity 或者 Fragment，其界面如下：



4.1 配置

您可以根据需求配置 App Market

1. 设置商店名称：

```
public static void setAppMarketName(Context context, String name)
```

参数说明：
context: 上下文
name: 您所希望的商店名，将显示在商店的左上角，若不调用该接口，默认商店名为：“Market”

注意：必须在进程的主线程里面初始化！

2. 配置广告墙样式

```
public static void setMarketStyle
(Context context, HashMap<String, Integer> marketStyle)
```

参数说明：
context: 上下文
marketStyle : 广告墙样式配置，**参数为 resource id**, 可以配置项如下（若某一项设置为 null，表示该项用默认配置）：

TITLE_BACKGROUND_COLOR	标题栏背景色
TITLE_TEXT_COLOR	标题栏字体颜色
TITLE_TEXT_SIZE	标题栏字体大小
TITLE_BAR_HEIGHT	标题栏高度
TITLE_BACK_DRAWABLE	标题栏返回按钮图片（默认图片大小 20x36 pixel）
TABLE_BACKGROUND_COLOR	Table 栏背景色
TABLE_TEXT_COLOR	Table 栏字体颜色
TABLE_TEXT_SIZE	Table 栏字体大小
TABLE_BAR_HEIGHT	Table 栏高度
TABLE_INDICATOR_COLOR	Table 栏指示器颜色
DK_BUTTON_BACKGROUND_COLOR	大卡广告按钮背景色
DK_BUTTON_TEXT_COLOR	大卡广告按钮字体颜色

INSTALL_TEXT_BACKGROUND_DRAWABLE	安装按钮背景图片（默认：140x60 pixel）
INSTALL_TEXT_COLOR	安装按钮字体颜色
STATUS_COLOR	状态栏颜色
NAVIGATION_COLOR	导航栏颜色
WALL_BACKGROUND_COLOR	应用墙内容背景色
AD_TITLE_TEXT_COLOR	广告标题字体颜色
AD_DESCRIPTION_TEXT_COLOR	广告描述字体颜色
CATEGORY_TEXT_COLOR	广告分类字体颜色
CATEGORY_TEXT_SIZE	广告分类字体大小
CATEGORY_OF_RECOMMEND	推荐分类名称
CATEGORY_OF_POPULAR	流行分类名称
CATEGORY_OF_LIKE	喜欢分类名称
AD_CLICK_COVER_LAYER_TRANS_BACKGROUND	广告点击时界面蒙层颜色

使用示例：

```
Protected void onCreate(Bundle savedInstanceState) {
setMarketStyle();
}

private void setMarketStyle() {

        AdSdk.setMarketStyle(this, marketStyle);
}
```

4.2 预加载应用墙数据

为提高 market 页面的加载速度，以及优化广告的填充和转化，强烈建议开发者预加载应用墙的数据：

```
public static void preloadMarketData (Context context)
```

参数说明：

- context: 上下文，建议使用 ApplicationContext

使用示例：

```
    @Override
    protected void onResume() {
        super.onResume();
        preloadMarketWall();
    }

    private void preloadMarketWall() {
            marketStyle.put(Constants.MarketStyle.TITLE_BACKGROUND_COLOR,
R.color.white);
            marketStyle.put(Constants.MarketStyle.TITLE_TEXT_COLOR, R.color.gray);
            marketStyle.put(Constants.MarketStyle.TITLE_BAR_HEIGHT,
R.dimen.title_bar_height);
            marketStyle.put(Constants.MarketStyle.TITLE_TEXT_SIZE,
R.dimen.anative_appwall_title_text_size);
            marketStyle.put(Constants.MarketStyle.TITLE_BACK_DRAWABLE,
R.drawable.appwall_adrss_ic_back);
           marketStyle.put(Constants.MarketStyle.TABLE_BACKGROUND_COLOR,
R.color.white);
            marketStyle.put(Constants.MarketStyle.TABLE_TEXT_COLOR, R.color.gray);
```

```
        marketStyle.put(Constants.MarketStyle.TABLE_TEXT_SIZE,
R.dimen.table_text_size);
        marketStyle.put(Constants.MarketStyle.TABLE_INDICATOR_COLOR,
R.color.light_gray);
        marketStyle.put(Constants.MarketStyle.TABLE_BAR_HEIGHT,
R.dimen.table_bar_height);
 marketStyle.put(Constants.MarketStyle.DK_BUTTON_BACKGROUND_COLOR,
R.color.black);
        marketStyle.put(Constants.MarketStyle.DK_BUTTON_TEXT_COLOR,
R.color.white);

marketStyle.put(Constants.MarketStyle.INSTALL_TEXT_BACKGROUND_DRAWABLE,
R.drawable.anative_appwall_adress_button_type1);
        marketStyle.put(Constants.MarketStyle.INSTALL_TEXT_COLOR, R.color.gray);
        marketStyle.put(Constants.MarketStyle.STATUS_COLOR, null);
        marketStyle.put(Constants.MarketStyle.NAVIGATION_COLOR, null);
        marketStyle.put(Constants.MarketStyle.WALL_BACKGROUND_COLOR, null);
        marketStyle.put(Constants.MarketStyle.AD_TITLE_TEXT_COLOR, R.color.gray);
        marketStyle.put(Constants.MarketStyle.AD_DESCRIPTION_TEXT_COLOR,
R.color.light_gray);

marketStyle.put(Constants.MarketStyle.AD_CLICK_COVER_LAYER_TRANS_BACKGRO
UND, R.color.cover_layer_background);

        marketStyle.put(Constants.MarketStyle.CATEGORY_TEXT_COLOR,
R.color.gray);
        marketStyle.put(Constants.MarketStyle.CATEGORY_TEXT_SIZE,
R.dimen.category_text_size);
        marketStyle.put(Constants.MarketStyle.CATEGORY_OF_RECOMMEND,
R.string.recommend);
        marketStyle.put(Constants.MarketStyle.CATEGORY_OF_POPULAR,
```

```
R.string.popular);
            marketStyle.put(Constants.MarketStyle.CATEGORY_OF_LIKE, R.string.like);

        AdSdk.preloadMarketData(this.getApplicationContext());
    }
```

4.3 启动 market 页面

有两种方式启动 market 页面：

1. 用 Activity 启动：
   在需要打开 App market 处调用该接口

```
public static void showAppMarket(Context context)
```

2. 用 Fragment 启动：
   在打开 market fragment 前需要调用

```
public static void setMarketFragmentMode(Context context, boolean
isFragmentMode)
```

参数说明：
context: 上下文
isFragmentMode: 设为 true 则为 fragment 接入模式

在需要获取 Fragment 对象的地方用如下方法获取

```
Fragment fr = AdSdk.getFeatureFragment(Context context);
```

5. 原生广告接入

集成后的可能效果

5.1 接入流程

请将 anative_common.aar, anative_native.aar 复制到相应位置

1. 确保 SDK 已经在 Application 中初始化
2. 若集成第三方广告源， 请在 build.gradle 文件中增加相应依赖：

```
compile 'com.google.android.gms:play-services-ads:8.4.0'        // AdMob
compile 'com.facebook.android:audience-network-sdk:4.23.0'    // Fb
```

3. 初始化和预加载

构建一个 NativeAd 对象，并传入该广告位 unit ID，设置广告加载监听；展示广告之前，提前将广告拉到本地，提升用户体验，提高广告收入

```
INativeAd  nativeAd = new NativeAd(Context activityContext, String unitId);
nativeAd.setNativeAdListener(new NativeAdListener() {
        @Override
        public void onAdLoaded() {  // 广告加载成功回调
        }
        @Override
        public void onError(String error) {
```

```
        }
    })
mNativeAd.load(); // 加载广告
```

4. 显示广告

当广告预加载成功后，调用接口 show 展示广告；adContainer 是用来展示原生广告的容器。

```
if (nativeAd.isLoaded()) {
    nativeAd.show(ViewGroup adContainer); // 显示广告
}
```

广告容器配置参考：

```
<FrameLayout
    android:id="@+id/native_container"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```