Avazu Native Ad SDK Setup for Android

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Developer |
| 2017.04.26 | V2.2.5 | Improve Appwall | 卢红伍 |
|  |  |  |  |
|  |  |  |  |

目录

[1.简介 3](#_Toc478401738)

[1.1 SDK功能介绍 3](#_Toc478401739)

[1.2 兼容性 3](#_Toc478401740)

[2. 接入配置 3](#_Toc478401741)

[2.1 Traffic Source ID 3](#_Toc478401742)

[2.2 导入SDK文件 3](#_Toc478401743)

[2.3 代码混淆配置 4](#_Toc478401744)

[3. 展示App Market 4](#_Toc478401745)

[3.1 初始化SDK 5](#_Toc478401746)

[3.2 配置 5](#_Toc478401747)

[3.3 预加载应用墙数据 7](#_Toc478401748)

[3.4 启动market页面 8](#_Toc478401749)



1.简介

1.1 SDK功能介绍

Avazu ADSDK 能够提供以下功能

* 通过showAppMarket接口，提供App Market页面

1.2 兼容性

支持Android 2.2（API level 8）及以上系统

2. 接入配置

2.1 Traffic Source ID

确保您已注册Avazu APX账号且已有可用的Traffic Source ID for Display

2.2 导入SDK文件

文件清单：

* avazu\_common-<version>.aar
* avazu\_appwall-<version>.aar

1. 将aar文件保存到本地工程中

* 将avazu\_common-<version>.aa 和avazu\_appwall-<version>.aar拷贝到您项目的libs文件夹中，并作为您项目的依赖库:
* 在项目的build.gradle文件中的dependencies添加：

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

compile(name: avazu\_common-<version>, ext: 'aar')

compile(name: avazu\_appwall-<version>, ext: 'aar')

compile 'com.my.target:mytarget-sdk:4.5.10' //VK广告，不需要不加

compile 'com.google.android.gms:play-services-ads:8.1.0' // Admob广告， 不需要不加

注：

1.若您已使用play-services，请不要重复添加。

2.若需要Facebook广告，请从demo源码libs/AudienceNetwork.jar 拷贝

2.3 代码混淆配置

在您的proguard文件中添加：

# AVAZU 应用墙

-keep class nativesdk.ad.adsdk.\*\* {\*;}

-dontwarn nativesdk.ad.adsdk.\*\*

-keep class nativesdk.ad.adsdkcore.\*\* {\*;}

-dontwarn nativesdk.ad.adsdkcore.\*\*

# 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.\*\*

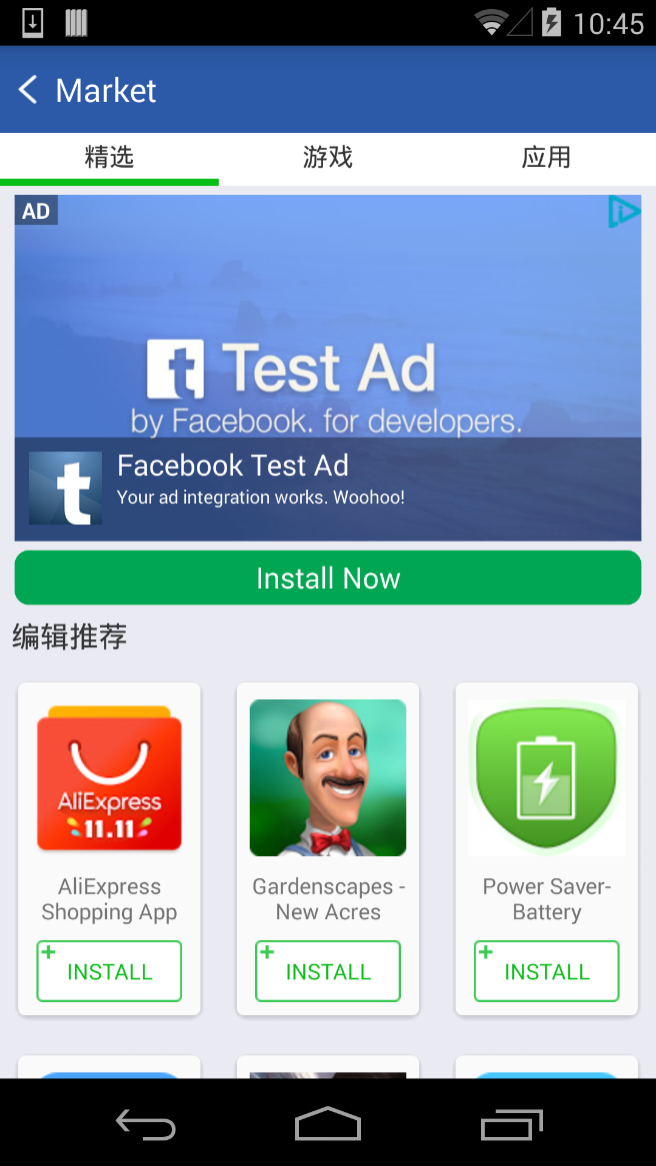
# VK 若没加，可不添加

-keep class com.mopub.\*\* {\*;}

-dontwarn com.mopub.\*

3. 展示App Market

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：



3.1 初始化SDK

在Application的onCreate里面调用该初始化方法:

|  |
| --- |
| Adsdk.initialize(Context context, String marketSourceId); |

参数说明：

* context: Application 上下文
* marketSourceId: 市场页面的Traffic Source Id

3.2 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”

2. 创建market桌面快捷方式

|  |
| --- |
| public static void createMarketShortcut  (Context context, Bitmap marketIcon, String marketName) |

参数说明：

* context: 上下文
* marketName: Bitmap格式的桌面图标
* marketName: 桌面快捷方式的名字

3. 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，可以配置项如下（若某一项不设置或设置为0，表示该项用默认配置）：
* TITLE\_BACKGROUND\_COLOR 标题背景颜色
* TITLE\_TEXT\_COLOR 标题字体颜色
* TITLE\_BAR\_HEIGHT 标题栏高度
* TABLE\_BACKGROUND\_COLOR TAB标签背景颜色
* TABLE\_TEXT\_COLOR TAB标签字体颜色
* TABLE\_BAR\_HEIGHT TAB栏 高度
* TABLE\_INDICATOR\_COLOR TAB指示器颜色
* DK\_BUTTON\_BACKGROUND\_COLOR 大卡按钮背景颜色
* DK\_BUTTON\_TEXT\_COLOR 大卡按钮字体颜色
* INSTALL\_TEXT\_BACKGROUND\_DRAWABLE 安装按钮自定义图片
* INSTALL\_TEXT\_COLOR 安装按钮字体颜色
* WALL\_STATUS\_COLOR 状态栏颜色
* WALL\_NAVIGATION\_COLOR 导航栏颜色
* BACK\_BUTTON\_DRAWABLE 返回按钮自定义图片
* APPWALL\_BACKGROUND\_COLOR 应用墙内容背景颜色
* SUBTITLE\_TEXT\_COLOR 广告分类标题字体颜色
* AD\_TITLE\_TEXT\_COLOR 广告标题字体颜色
* AD\_DESCRIPTION\_TEXT\_COLOR 广告描述字体颜色

使用示例：

|  |
| --- |
| Protected void onCreate(Bundle savedInstanceState) {  Adsdk.initialize(this);  setMarketStyle();  }  private void setMarketStyle() {  marketStyle.put(Constants.MarketStyle.TITLE\_BACKGROUND\_COLOR, R.color.white);  marketStyle.put(Constants.MarketStyle.TITLE\_TEXT\_COLOR, R.color.gray);  marketStyle.put(Constants.MarketStyle.TABLE\_BACKGROUND\_COLOR, R.color.white);  marketStyle.put(Constants.MarketStyle.TABLE\_TEXT\_COLOR, R.color.gray);  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.apx\_appwall\_adress\_button\_type1);  marketStyle.put(Constants.MarketStyle.INSTALL\_TEXT\_COLOR, R.color.gray);  marketStyle.put(Constants.MarketStyle.BACK\_BUTTON\_DRAWABLE, R.drawable.apx\_appwall\_adrss\_ic\_back);  marketStyle.put(Constants.MarketStyle.WALL\_STATUS\_COLOR, 0);  marketStyle.put(Constants.MarketStyle.WALL\_NAVIGATION\_COLOR, 0);  marketStyle.put(Constants.MarketStyle.APPWALL\_BACKGROUND\_COLOR, R.color.white);  marketStyle.put(Constants.MarketStyle.TITLE\_BAR\_HEIGHT, 0);  marketStyle.put(Constants.MarketStyle.TABLE\_BAR\_HEIGHT, 0);  marketStyle.put(Constants.MarketStyle.TABLE\_INDICATOR\_COLOR, R.color.light\_gray);  marketStyle.put(Constants.MarketStyle.SUBTITLE\_TEXT\_COLOR, R.color.gray);  marketStyle.put(Constants.MarketStyle.AD\_TITLE\_TEXT\_COLOR, R.color.gray);  marketStyle.put(Constants.MarketStyle.AD\_DESCRIPTION\_TEXT\_COLOR, R.color.light\_gray);  AdSdk.setMarketStyle(this, marketStyle);  } |

3.3 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() {  AdSdk.enableFacebookAdInMarket(MainActivity.this, "your-fb-id"); // 若无，不添加  AdSdk.enableAdmobInMarket(MainActivity.this, "your-ad-id"); // 若无，不添加  AdSdk.preloadMarketData(this.getApplicationContext());  } |

3.4 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 = new FeatureFragment(); |