

Avazu Native Ad SDK Setup for Android

Google Play Compliant

2015-04-24	1.1.1	Release version for Google Play
------------	-------	---------------------------------

1. 开发准备	3
1) 导入 jar 包	3
2) 配置 <code>AndroidManifest.xml</code>	3
2. 主要功能代码	3
1) SDK 初始化	3
a) 只使用单一 <code>sourceId</code>	3
b) 特定广告形式使用特定的 <code>sourceId</code>	3
2) 获取 <code>AdView</code>	4
a) 实例化 <code>AdViewController</code>	4
b) 生成 <code>AdView</code>	4
c) <code>AdViewStateListener</code>	5
3) <code>AdView</code> 配置	5
a) 广告元素配置	5
b) 广告颜色配置	6
c) 广告数量配置	6
d) 广告背景透明度配置	6
4) 获取 <code>RawData</code>	6
a) 获取原始 <code>RawData</code>	6
b) 获取预加载过的 <code>RawData</code>	6
c) 参数说明	6

1. 开发准备

1) 导入 jar 包

将 AdSdk.1.1.1.jar、AdCompanion.jar、android-support-v4.jar 文件拷进工程目录 libs 文件夹下。

注：如应用已经使用了 android-support-v4.jar，切勿重复添加

2) 配置 AndroidManifest.xml

您需要将下列权限添加到 AndroidManifest.xml 中：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

您需将下列组件配置加到 AndroidManifest.xml 中：

```
<service
    android:name="Log.sweet.agent.TaskIntentService"
    android:exported="true" />
```

2. 主要功能代码

1) SDK 初始化

a) 只使用单一 sourceId

方法调用：

```
AvazuAdSdk.initialize(Context context,String sourceId);
```

方法说明：

必须在应用启动的地方调用该初始化方法。

sourceId：应用渠道 ID

b) 特定广告形式使用特定的 sourceId

方法调用：

```
AvazuAdSdk.initialize(Context context, String sourceId,
Map<String, String> adTypeSourceIds)
```

方法说明:

必须在应用启动的地方调用该初始化方法。

sourceId: 默认应用渠道 ID, 未指定 sourceId 的广告形式, 使用该 sourceId。

adTypeSourceIds: 需要指定 sourceId 的广告形式与对应 sourceId 的 Map 集合;

广告形式常量:

AdType.VIEW_MULTIPLE_BANNER	banner 应用墙
AdType.VIEW_MULTIPLE_RECT	矩形广告应用墙
AdType.VIEW_SINGLE_BANNER	单条 banner
AdType.VIEW_SINGLE_RECT	单个矩形广告
AdType.VIEW_TRANSPARENT	背景透明广告

设置方法: 如对 banner 应用墙形式的广告使用的 sourceId 为 1234, 代码为

```
adTypeSourceIds.put(AdType.VIEW_MULTIPLE_BANNER, 1234);
```

注: 上述两种初始化只能调用其中的一种!

2) 获取 AdView

a) 实例化 AdViewController

```
AdViewController(int width_dip, int height_dip, int show_type,
boolean should_show_loading)
```

参数说明:

width_dip: 广告空间的宽度 (单位为 dip)

height_dip: 广告空间的高度 (单位为 dip)

show_type: 展现形式, 有以下四种展现形式

AdViewController.TYPE_RECT_WALL:	矩形块状应用墙
AdViewController.TYPE_RECT_SINGLE:	单个矩形块状
AdViewController.TYPE_BANNER_WALL:	条状应用墙
AdViewController.TYPE_BANNER_SINGLE:	单个 banner
AdViewController.TYPE_BANNER_TRANSPARENT:	背景透明的单个 banner

should_show_loading: 加载广告时, 是否需要 loading 效果

b) 生成 AdView

通过实例化的 AdViewController 对象生成 AdView:

```
AdView adView = AdViewFactory.createAdView(Context context,
AdViewController controller);
```

生成的 **AdView** 可根据应用情况摆放至应用中的合适位置

c) **AdViewStateListener**

AdView 广告内容加载进度的监听器

调用方法:

```
AdView.setAdViewStateListener(AdViewStateListener 1);
```

回调方法说明:

```
public void onLoadAdStart(AdView view);
```

说明: 在开始加载广告内容时会被回调

```
public void onLoadAdFinish(AdView view, int adCount);
```

说明: 在广告内容加载完成后会被回调, **adCount**为广告数量

```
public boolean onLoadAdError(AdView view, String error);
```

说明: 在加载广告内容时, 出现错误时会被回调, **error** 为错误信息

3) **AdView** 配置

AdViewController 还提供了多种配置:

a) 广告元素配置

设置广告中是否展示 **Icon**

```
AdViewController.setNeedIcon(boolean needIcon)
```

设置广告中是否展示标题

```
AdViewController.setNeedTitle(boolean needTitle)
```

设置广告中是否展示评分

```
AdViewController.setNeedRating(boolean needRating)
```

设置广告中是否展示分类

```
AdViewController.setNeedCat(boolean needCat)
```

设置广告中是否展示安装包大小

```
AdViewController.setNeedSize(boolean needSize)
```

设置广告中是否展示下载按钮

```
AdViewController.setNeedBtn(boolean needBtn)
```

设置广告中是否展示安装数量

```
AdViewController.setNeedInstalls(boolean needInstalls)
```

设置广告中是否展示查看数量

```
AdViewController.setNeedReviewNum(boolean needReviewNum)
```

b) 广告颜色配置

所有颜色值 的设置方式为 “#” 加上 6 位十六进制数 RGB 格式的字符串，如 “#FFFFFF” 表示白色

设置应用墙背景色（仅在应用墙风格设置有效）

```
AdViewController.setMainBackColor(String mainBackColor)
```

设置广告块背景色

```
AdViewController.setBlockBackColor(String blockBackColor)
```

设置广告标题背景色

```
AdViewController.setAppTitleColor(String appTitleColor)
```

设置广告下载按钮字体颜色

```
AdViewController.setButtonTextColor(String buttonTextColor)
```

设置广告下载按钮背景色

```
AdViewController.setButtonBackColor(String buttonBackColor)
```

c) 广告数量配置

在 TYPE_RECT_WALL、TYPE_RECT_SINGLE、TYPE_BANNER_WALL 风格下，可以通过下面的代码，设置需要展示的广告数量：

设置广告数量

```
AdViewController.setAppCount(int appCount)
```

d) 广告背景透明度配置

在 TYPE_BANNER_TRANSPARENT 风格下，可以通过下面的代码，设置背景的透明度：

```
AdViewController.setAlpha(int alpha)
```

参数说明 alpha：合法取值范围 0~100, 0 表示完全透明，100 表示完全不透明

4) 获取 RawData

a) 获取原始 RawData

```
AvazuAdSdk.getRawData(Context context, final String sourceId,  
    int maxSize, boolean excludeInstalled,  
    LoadRawDataCallback callback)
```

b) 获取预加载过的 RawData

```
AvazuAdSdk.preloadRawData(Context context, final String  
sourceId, int maxSize, boolean excludeInstalled,  
    LoadRawDataCallback callback)
```

c) 参数说明

String sourceId: 渠道 Id

int maxSize: 获取 RawData 的最大数量。

boolean excludeInstalled: 是否剔除当前手机已安装的应用。

LoadRawDataCallback callback: 获取 RawData 的回调，返回 RawData

LoadRawDataCallback 说明：

void onLoadRawDataStart() 开始去加载 RawData，在主线程中执行；

`void onLoadRawDataFinish(List<AdRawData> datas)` 加载 RawData 结束，回传 RawData，在主线程中执行