

Avazu Native Ad SDK Setup for Android

Date	Version	Description	Developer
2015.07.17	v1.3.1	Add Function	曹诗聪
2015.08.27	v1.3.3	Add new parameter for C2S API	曹诗聪

1. 简介	3
1.1 SDK功能介绍	3
1.2 兼容性	3
2. 接入配置	4
2.1 Traffic Source ID	4
2.2 导入SDK文件	4
2.3 配置AndroidManifest.xml	4
3. 展示广告	5
3.1 初始化SDK	5
3.2 创建AdView	5
3.3 自定义AdView	6
4. C2S接口	9
5. DirectToMarket	11

1.简介

1.1 SDK功能介绍

Avazu ADSDK 能够提供以下功能

- 展示单个广告banner (推荐尺寸: 320 * 100)
- 展示透明banner (推荐尺寸: 400 * 110)
- 展示广告banner墙
- 展示单行矩形广告墙
- 展示多行矩形广告墙
- 通过C2S接口获取原始广告数据(包括Google Play广告, DDL广告以及全屏订阅类广告)
- 通过DirectToMarket接口, 随机提供一个Google Play地址进行跳转

以上这些广告展现方式, 开发者可以根据SDK提供的方法实现广告外观和展示应用数量的定制

1.2 兼容性

支持Android 2.3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

请将以下文件添加到您的项目中：

- [adsdk_1.3.1.jar](#)

同时导入android support v7 和 Google Service服务

注：如不添加这两个依赖库，会导致应用无正常运行，如应用已经使用了android-support-v7和gms:play-services，请不要重复添加

2.3 配置AndroidManifest.xml

为了保证SDK能够正确运行，您需要在AndroidManifest中添加如下权限：

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

3. 展示广告

AvazuADSDK 提供了5种不同种类的Webview广告展现形式，包括单个广告banner，广告banner墙单行矩形广告墙，多行矩形广告墙，透明banner。

您可以选择通过以下步骤展示广告：

3.1 初始化SDK

在应用启动的地方调用该初始化方法：

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

sourceId: 可用的的 Traffic Source ID是，每次创建广告实例时都需要用到它，否则将无法正确展示广告

3.2 创建AdView

创建Adview步骤如下：

1.实例化AdViewSettings

您可以实例化以下方法：

```
AdViewSettings(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: 广告的展现形式，有以下五种：

广告形式	变量名
单个广告banner	AdViewSettings.TYPE_BANNER_SINGLE
banner广告墙	AdViewSettings.TYPE_BANNER_WALL
透明banner	AdViewSettings.TYPE_BANNER_TRANSPARENT
单行矩形广告墙	AdViewSettings.TYPE_RECT_SINGLE
多行矩形广告墙	AdViewSettings.TYPE_RECT_SINGLE

- should_show_loading: 加载广告时是否需要加载效果

2. 生成AvazuAdView

通过实例化的AdViewSettings对象生成AdView：

```
AvazuAdView adView =new AvazuAdView(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

3. 加载 AvazuAdView

在需要显示广告的时候用此方法加载广告:

```
adView.loadWebViewAd();
```

4. 设置广告加载结果监听

调用方法:

```
adView.setAdViewStateListener(AdViewStateListener l);
```

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, 当adCount为0时应用可根据自身需要进行特定处理

```
public void 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

3.3 自定义AdView

AvazuADSDK支持丰富的定制化广告, 您可以选择展示的广告的颜色搭配, 展示内容以及展示应用的数量, 只需要像如下的代码进行设置就可:

```
AdViewSettings.setNeedIcon(boolean needIcon);
```

1. 广告元素配置

方法名	变量类型	变量名	作用
setNeedIcon	boolean	needIcon	设置广告中是否显示应用图标
setNeedTitle	boolean	needTitle	设置广告中是否显示应用名称
setNeedCat	boolean	needCat	设置广告中是否显示应用类别
setNeedSizr	boolean	needSize	设置广告中是否显示应用尺寸
setNeedRating	boolean	needRating	设置广告中是否显示应用评分
setNeedBtn	boolean	needBtn	设置广告中是否显示安装按钮
setNeedReviewNum	boolean	needReviewNum	设置广告中是否显示应用查看次数
setNeedInstalls	boolean	needInstalls	设置广告中是否显示应用安装次数

2. 广告颜色配置

方法名	类型	变量名	作用
setBlockBackColor	String	blockBackColor	设置广告块背景色
setAppTitleColor	String	appTitleColor	设置应用名字体颜色
setButtonBackColor	String	buttonBackColor	设置下载按钮颜色
setButtonTextcolor	String	buttonTextColor	设置下载按钮上的字体颜色
setMainBackColor	String	mainBackColor	设置应用墙背景色

注意: 配置广告颜色时, 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, “#000000”表示黑色

3. 展示应用数量配置

方法名	类型	变量名	作用
setAppCount	int	appCount	设置广告中出现的应用数量

注意: 以下广告类型必须设置appCount: 广告banner墙, 单行矩形广告墙, 多行矩形广告墙

4. 设置广告透明度

变量名	类型	变量名	作用
setAlpha	int	alpha	设置透明广告的透明度

注意: 只有在透明banner的广告形式下才用设置transparentBannerAlpha, 合法取值范围0~100,0表示完全透明, 100表示完全不透明

以下是一个定制banner的完整代码:

```
AdViewSettings adSettings = new AdViewSettings(400, 110,
AdViewSettings.TYPE_BANNER_TRANSPARENT, false);
adSettings.setNeedIcon(true);
adSettings.setNeedBtn(true);
adSettings.setNeedCat(true);
adSettings.setNeedInstalls(true);
adSettings.setNeedRating(true);
adSettings.setNeedReviewNum(true);
adSettings.setNeedTitle(true);
adSettings.setAppTitleColor("#FFFFFF");
adSettings.setMainBackColor("#f5f5f5");
adSettings.setBlockBackColor("#080807");
adSettings.setAlpha(80);
AvazuAdView adView = new AvazuAdView(this, adSettings);
adView.loadWebviewAd();
```


4. C2S接口

您可以通过以下方式使用C2S接口:

```
AdSDK.getAdRawData(Context context, final String sourceId,
String excludePackages, int limitNumber, int creatives, String market,
FetchRawDataListener listener)
```

参数说明:

- Source ID: Traffic Source ID
- excludePackages: 传入不想显示的广告的 campaign_id, 用于翻页时不显示重复广告,若有多个以“,”分隔,若无则传入“”,例如:“6184, 3241”
- limitNumber: 本次想要拉取广告的个数的上线
- creatives: 是否返回多尺寸图片素材,值为 1 表示返回 图片素材,默认为 0(不返回图片素材)。可能返回的图片尺寸有:

尺寸	参数名
320x50	banner
320x480	phone_fullscreen
480x320	phone_fullscreen_landscape
1024x768	tablet_fullscreen
768x1024	tablet_fullscreen_landscape
300x250	medium
728x90	leaderboard
160x600	skyscraper
1200x627	content_stream_image

- market: 选择获取Rawdata的广告类型，具体如下，若传入其他值默认传入google

传入String	对应广告类型
google	click_url为跳转Google Play的广告
ddl	click_url为直接下载APK的广告
optin	click_url为可显示为全屏的广告

- listener：获取RawData的回调，返回RawData， 有以下可重写方法：

public void onLoadRawDataStart(): 开始加载RawData， 在主线程中执行

public void onLoadRawDataSuccess(List<FetchAdResult.Ad> data):

加载RawData结束， 回传RawData， 在主线程中执行

public void onLoadRawDataFail(Error mError): 加载RawData失败， 在主线程中执行

FetchAdResult.Ad为获取的广告数据， 其数据结构如下：

字段	描述
campaignid	广告 ID
payout	一个转化的价格(CPI),单位为美元
pkgname	App 的包名(package name)
title	App 的标题
description	App 的描述(不超过 100 个英文字母)
icon	App 的 icon 地址(100x100 的 png 图片)
appcategory	App 分类
apprating	App 的评分(0-5 星)
appreviewnum	App 的评论数
appinstalls	App 的安装量
appsize	App 的安装文件大小
creatives	App 的多尺寸图片素材地址
clkurl	广告的点击 URL

注：在获取creatives中您想要的尺寸的url数组前，需进行判空处理，范例如下：

```

if (datadata.tablet_fullscreen != null) {
    L.d(datadata.tablet_fullscreen);
}

```

5. DirectToMarket

DirectToMarket接口能够实现挑选一个随机广告直接跳转Google Play页面, 具有高度自定义性, 请在您应用中自定义的广告跳转按钮按下时调用。

您可以通过以下方式使用 DirectToMarket 接口:

```
public static void directToMarket(Context context, String souceId,
DirectToMarketManager.DirectToMarketListener listener)
```

参数说明:

- Source ID: Traffic Source ID
- listener: 用来监听跳转Google Play的结果

public void DirectToMarketSuccess(): 跳转成功

public void DirectMarketToFail(): 跳转失败