

# Avazu Native Ad SDK Setup for Android

## Google Play Compliant

2015-04-24	1.1.1	Release version for Google Play
2015-05-12	1.2.0	Change server

<b>1. Development Preparation.....</b>	<b>3</b>
1) Import jar package.....	3
2) Configure AndroidManifest.xml.....	3
<b>2. Main Functions.....</b>	<b>3</b>
1) SDK Initialization.....	3
a) Use a single source ID.....	3
b) Use different source IDs for different ad types.....	3
2) Retrieve AdView.....	4
a) Instantiate AdViewController.....	4
b) Generate AdView.....	5
c) AdViewStateListener.....	5
3) AdView Configuration.....	5
a) Customize App Elements.....	5
b) Customize Ad Color.....	6
c) Amount of Apps.....	6
d) Background Transparency.....	6
4) Get Raw Data.....	6
a) Get the Raw Data.....	6
b) Get the Preloaded Raw Data.....	6
c) Parameter Description.....	7

# 1. Development Preparation

## 1) Import jar package

Copy “AdSdk.1.1.1.jar”, “AdCompanion.jar” and “android-support-v4.jar” files to the “libs” directory of your project.

**Note:** If your project has already added “android-support-v4.jar”, do not add it twice.

## 2) Configure AndroidManifest.xml

Add the following permissions to “AndroidManifest.xml”.

```
<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"/>
```

Add the following codes to “AndroidManifest.xml”.

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

# 2. Main Functions

## 1) SDK Initialization

### a) Use a single source ID

Call the initialization function at the startup of your application.

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

sourceId: Traffic Source ID

### b) Use different source IDs for different ad types

Call the initialization function at the startup of your application.

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

### adTypeSourceIds)

sourceId: Default Traffic Source ID

adTypeSourceIds: Specify the Map of different traffic source IDs for different ad types

Ad Type Constants:

<b>AdType.VIEW_MULTIPLE_BANNER</b>	Banner app wall
<b>AdType.VIEW_MULTIPLE_RECT</b>	Button app wall of multiple lines
<b>AdType.VIEW_SINGLE_BANNER</b>	Single banner
<b>AdType.VIEW_SINGLE_RECT</b>	Button app wall of single line
<b>AdType.VIEW_TRANSPARENT</b>	Single transparent banner

For example, if you want to set 1234 as the traffic source ID of banner app wall

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

**Note: You can choose only one of the above initialization methods!**

## 2)Retrieve AdView

### a) Instantiate AdViewController

AdViewController(int width\_dip, int height\_dip, int show\_type, boolean should\_show\_loading)

width\_dip: Ad width in **dip**

height\_dip: Ad height in **dip**

show\_type: One of the following ad format types

AdViewController.TYPE\_RECT\_WALL: Multiple-line rectangle app wall

AdViewController.TYPE\_RECT\_SINGLE: Single-line rectangle app wall

AdViewController.TYPE\_BANNER\_WALL: Banner app wall

AdViewController.TYPE\_BANNER\_SINGLE: Single banner

AdViewController.TYPE\_BANNER\_TRANSPARENT: Single banner with transparent background

`should_show_loading`: The switch to show loading effect or not when the AdView is loading

### b) Generate AdView

Use the instantiated AdViewController object to generate AdView.

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

Place the generated AdView to the proper position in your application.

### c) AdViewStateListener

You can listen to the loading status of AdView

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

Recall methods:

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

This method will be called when the AdView starts to load

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

This method will be called when the AdView finishes loading

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

This method will be called when any error occurs on loading. The string "error" contains the error message.

## 3) AdView Configuration

AdViewController also provides configuration options.

### a) Customize App Elements

Show app icon or not

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

Show app title or not

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

Show app ranking score or not

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

Show app category or not

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

Show app package size or not

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

---

Show download button or not  
`AdViewController.setNeedBtn(boolean needBtn)`

Show app installation number or not  
`AdViewController.setNeedInstalls(boolean needInstalls)`

Show app review number or not  
`AdViewController.setNeedReviewNum(boolean needReviewNum)`

## b) Customize Ad Color

All color value must be “#” followed by 6 hexadecimal RGB **digitals**. For example, #FFFFFF is the white color.

Set the background color of app wall  
`AdViewController.setMainBackColor(String mainBackColor)`

Set the background color of the ad block  
`AdViewController.setBlockBackColor(String blockBackColor)`

Set the font color of the app title  
`AdViewController.setAppTitleColor(String appTitleColor)`

Set the font color of the download button  
`AdViewController.setButtonTextColor(String buttonTextColor)`

Set the background color of the download button  
`AdViewController.setButtonBackColor(String buttonBackColor)`

## c) Amount of Apps

For **TYPE\_RECT\_WALL**, **TYPE\_RECT\_SINGLE**, **TYPE\_BANNER\_WALL** only

Set the maximum amount of apps to show in the ad

`AdViewController.setAppCount(int appCount)`

## d) Background Transparency

For **TYPE\_BANNER\_TRANSPARENT** only

Set the transparency ratio of the background

`AdViewController.setAlpha(int alpha)`

alpha: should be between 0 (completely transparent) and 100 (completely opaque)

# 4)Get Raw Data

## a) Get the Raw Data

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

## b) Get the Preloaded Raw Data

`AvazuAdSdk.preloadRawData(Context context, final String sourceId, int maxSize, String excludePackages, boolean`

`excludeInstalled, LoadRawDataCallback callback)`

### c) Parameter Description

**String sourceId:** Traffic Source ID

**int maxSize:** The maximum ad number you want to get

**String excludePackages:** The campaign\_id of ad which you want exclude, it is useful to avoid displaying repeat ads when page changing. You can input "" when no exclude needs. If there are multiple ads needs to be excluded, you are required to divide the campaign by ",". For example: "6184, 3241".

**boolean excludeInstalled:** Exclude the apps of the current phone from the ad lists

**LoadRawDataCallback callback:** The callback method to proceed RawData

**void onLoadRawDataStart():** This will be called when the RawData starts to load. Please execute this method in the main thread.

**void onLoadRawDataFinish(List<AdRawData> datas):** This will be called when the RawData finishes loading. Please execute this method in the main thread.

---