Avazu Native Ad SDK Setup for Android

Date	Version	Description	Developer
2015.07.17	v1.3.1	Add Function	Steven.Cao
2015.08.27	v1.3.3	Add new parameter for C2S API	Steven.Cao
2015.11.28	v2.0.0	Add App Market and News Feed	Steven.Cao
2015.12.21	v2.0.2	Fix issue	Steven.Cao
2015.2.2	v2.1.0	Add Facebook Ad in News and fix issue	Steven.Cao

1.Introduction

1.1 What the SDK can provide

Avazu ADSDK can provide following functions in your mobile Android App:

- Show Single Banner (Recommended size: 320 * 100)
- Show Transparent Banner (Recommended size: 410* 100)
- Show Banner App Wall
- Show Single Line Button APP Wall
- Show Multiple Line Button APP Wall
- Get Ad raw data by C2S Interface(Including Google Play market Ad, DDL Ad and full screen Ad)
- Random jump to a ad's Google Play Landing page through DirectToMarket API
- An App Market Page with all function through one simple API
- A News Feed Page with all function through one simple API

For the ad types, the appearance and size of the ad are fully customizable with our function provided in the SDK.

1.2 Compatibility

The minimum runtime OS requirement is Android 2.3, or higher

2. Configuration

2.1 Traffic Source ID

Make sure you are have registered Avazu APX account and have a valid Traffic Source ID for display.

2.2 Choosing a SDK, and import third-party Ad SDK

File List:

- adsdk_2.0_with_fb.jar
- adsdk_2.0.jartn

Facebook:

If your project did not include Facebook advertising library yet, please choose adsdk_2.1.0_with_fb.jar

If your project already included Facebook advertising library, you can choose adsdk_2.1.0.jar, please make sure that the AudienceNetwork.jar you use is the latest version

Admob:

If your project did not include Admob library yet, please compile it into your project as following instruction:

Gradel project:

1. Please upgrade the Google Repository of your Android SDK to the latest version

2. Adding this line in Dependencies of your build.grade:

compile 'com.google.android.gms:play-services-ads:8.3.0'

Note: Do not add this line if you already compiled Google Play Service

Eclipse project :

1. Please upgrade the Google Repository of your Android SDK to the latest version

2. Create a new project from existing code, the path is:

<android-sdk>/extras/google/google_play_services/libproject/google-play-services_lib/

3. Open Property -> android menu of your project, adding dependency for the project you just created.

You can find the detail instruction in:
https://developers.google.com/mobile-ads-sdk/docs/admob/android/quick-start

2.3 Importing SDK files

- Copy adsdk_2.1.0_with_fb.jar or adsdk_2.1.0.jar to libs directory of your project
- Copy all files in assets directory we provided to assets directory of your project
- Adding Dependencies for android-support-v4, like:

compile 'com.android.support:support-v4:22.0.+'

2.4 Code obfuscation

Please add the following code in your pro guard file

-keep class com.facebook.**{*;}
-keep class com.google.ads.**{*;}
-keep class nativesdk.ad.adsdk.**{*;}

2.5 Configure AndroidManifest.xml

Please add the following permissions to "AndroidManifest.xml":

```
<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"/>
```

Please also add the following Activity

```
<activity
    android:name="natives dk.ad.adsdk.modules.activityad.MarketActivity"
    android:configChanges="orientation | keyboardHidden | screenSize"
    android:screenOrientation="portrait"/>

<activity
    android:name="natives dk.ad.adsdk.modules.activityad.AvLoadingActivity"
    android:configChanges="orientation | keyboardHidden | screenSize"
    android:screenOrientation="portrait"/>

<activity
    android:name="natives dk.ad.adsdk.modules.activityad.rss.NewsDetailActivity"
    android:configChanges="orientation | keyboardHidden | screenSize"
    android:screenOrientation="portrait"/>

<activity
    android:name="natives dk.ad.adsdk.modules.activityad.rss.NewsActivity"
    android:configChanges="orientation | keyboardHidden | screenSize"
    android:screenOrientation="portrait"/>

<activity
    android:name="com.facebook.ads.InterstitialAdActivity"
    android:configChanges="orientation | keyboardHidden | screenSize"
    android:exported="true"
    android:excludeFromRecents="true"
    android:noHistory="true"
    />
```

If you are willing to use createMarketShortcut API, you also need to add following code:

```
<uses-permission
android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />

<uses-permission
android:name="com.android.launcher.permission.READ_SETTINGS" />

<activity
    android:name="natives dk.ad.adsdk.modules.activityad.AvLoadingActivity"
    android:configChanges="orientation"
    android:screenOrientation="portrait">
    <intent-filter>
    <action android:name="natives dk.ad.adsdk.modules.activityad.
     AvLoadingActivity.SHORT_CUT" />
    <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

3. Display Webview Ad

AvazuADSDK provide 5 different types of ads, which include single banner, banner app wall, transparent banner, single line button App wall and multiple-line button App wall.

You can display webview Ad by following the instruction:

3.1 SDK Initialization

Call the initialization function at the startup of your application:

```
Adsdk.initialize(Context context);
```

3.2 Create AdView

1. create AvazuAdView

You can use below function:

```
AvazuAdView adView =new AvazuAdView(Context context, String adType,
String sourceId, int width, int height)
```

Parameter Description:

- context: context
- sourceId: Traffic Source Id for this Ad
- width: Ad width in dip
- height: Ad height in dip
- show_type: One of the following ad display types:

Display Format	Variable Name
Single banner	AvazuAdView.SINGLE_BANNER
Banner App Wall	AvazuAdView.MULTIPLE_LINE_BANNER
Transparent Banner	AvazuAdView.SINGLE_TRANSPARENT_BANNER
Single Line Button App Wall	AvazuAdView.SINGLE_LINE_RECTANGLE
Multiple Line Button App Wall	AvazuAdView.MULTIPLE_LINE_RECTANGLE

2. Setting Customizable Parameter

Please refer to section 3.3

3. Loading AvazuAdView

Using the function when you need to load ad:

```
adView.loadWebviewAd(Context context);
```

4. Setting Listener for Loading Result

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

Description of Callback function:

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 void onLoadAdError(AdView view, String error): This method will be called when any error occurs on loading. The string "error" contains the error message.

3.3 Customized AdView

AvazuADSDK support fully customization for different AD types, to apply the customized settings, you just to need one line code like:

```
adView.setAdIconVisibility(bolean isVisible)
```

1.Setting customized Ad Elements

Function Name	Variable Type	Usage
setAdIconVisibility	boolean	Show app's icon or not in adview
setAdTitleVisibility	boolean	Show app's title or not in adview
setAdCatagoryVisibility	boolean	Show app's category or not in adview
setAdSizeVisibility	boolean	Show app's size or not in adview
setAdRatingVisibility	boolean	Show app's rating or not in adview
setAdInstallButtonVisibility	boolean	Show install button or not in adview
setAdReviewNumberVisibility	boolean	Show app's review number or not in adview
setAdInstallNumberVisibility	boolean	Show app's install number or not in adview
setAdLoadingIndicatorVisibility	boolean	Show loading indicator or not when loading

2. Setting customized Ad Color

Function Name	Variable Type	Usage
setAdBlockBackgroundColor	String	set the background color of ad block
setAppTitleColor	String	set the font color for the app's title
setButtonBackColor	String	set the background color of install button
setButtonTextColor	String	set the font color of install button
setMainBackColor	String	set the background color of appeal

Note: When setting AD color, All color value must be a String started with "#" followed by 6 hexadecimal RGB digitals. For example, #FFFFFF represents for white color while #000000 represents for color black.

3. Setting App count

Function Name	Variable Type	Usage
setAppCount	int	Set the number of apps in ad

Note: appCount is required for these ad types:
Banner App Wall, Single Line Button App Wall and Multiple Line Button App Wall.

4. Setting Transparency Ratio For Transparent Banner

Function Name	Variable Type	Usage
setTransparentBannerAlpha	int	Set the transparent ratio for ad

Note: transparentBannerAlpha is required only for Transparent Banner, and the value should be set between 0 (completely transparent) and 100 (completely opaque).

Below you can find how to customize a banner ad:

```
AvazuAdView adView = new AvazuAdView(this,
AvazuAdView.SINGLE_LINE_RECTANGLE, "15887", showWidthDip, showHeightDip);
adView.setAdCatagoryVisibility(false);
adView.setAdSizeVisibility(false);
adView.setAdNumer(6);
adView.loadWebviewAd(this);
```

4. C2S Interface

You can get access to C2S interface by:

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

Parameter Description:

- Source ID: Traffic Source ID
- 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".
- limitNumber: The maximum ad number you want to get
- market: you can choose the Ad type of the raw data as below, the default value is "google"

Input String	Ad Type
google	The click url can direct to Google Play Market
ddl	The click url can direct to download APK
optin	The click url can direct to a full screen Ad

- creatives: If specified (i.e., creatives=1), images of various sizes will be returned. Default is "0", meaning no image will be returned.

Below you can find the available image size:

Size	Parameter Name
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

- listener: Get the callback of RawData, return the data of Ads, it has three override function:

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

public void onLoadRawDataSuccess(List<FetchAdResult.Ad> data):
This will be called when the RawData finishes loading. Please execute this method in the main thread.

public void onLoadRawDataFail(Error mError):This will be called when fetching Rawdata failed. Please execute this method in the main thread.

Below is the data structure of FetchAdResult.Ad.

Key	Description
campaignid	campaignid The campaign ID of the ad
payout	The payout of one app installation for this ad in USD
pkgname	The package name of the app
title	The title of the app
description	The description of the app
icon	The URL of the app icon (a png image with the size of 100x100)
appcategory	The category of the app
apprating	The rating of the app (0-5 stars)
appreviewnum	The review number of the app
appinstalls	The installation number of the app
appsize	The package file size of the app
creatives	The sizes and URLs of ad images
clkurl	The click URL of the ad

Note：You need to check if it is null before attempt to get the ad picture in your desired size

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

5. DirectToMarket

DirectToMarket is a simple API call that takes the user directly to a popular app on Google Play.

You can get access to the API by:

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

Parameter Description:

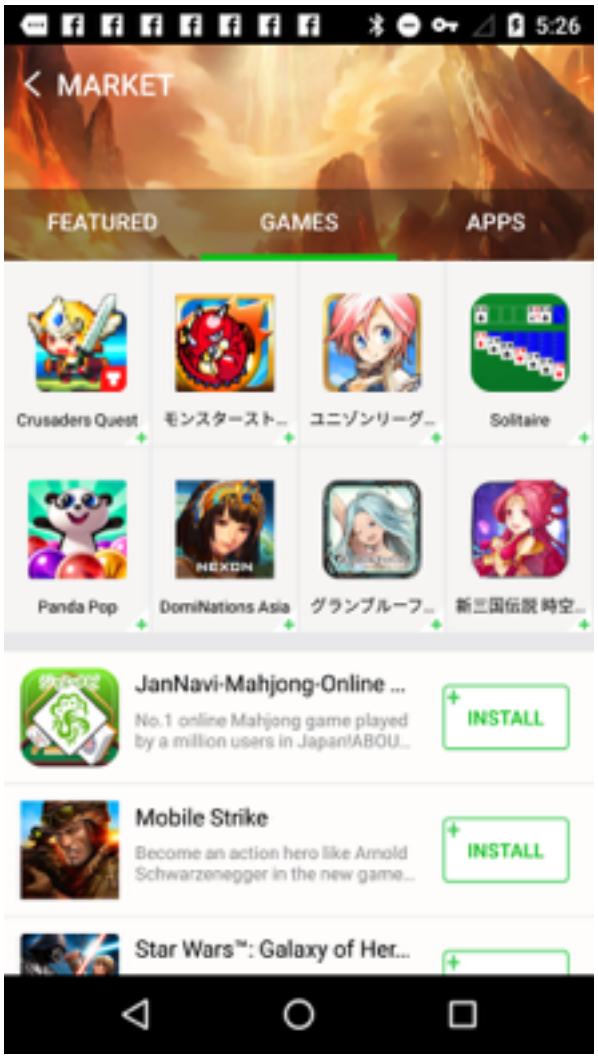
- Source ID: Traffic Source ID
- listener: Listen to the result of Google Play landing page jumping

public void DirectToMarketSuccess(): Jump to Google Play landing page Successfully

public void DirectMarketToFail(): Failed to jump to Google Play landing page

6. App Market

You can get a App Market Page through a very simple API, please check the UI in below picture



6.1 Configuration

You can configure the App Market as you like:

1.Setting App Market Name

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

Parameter Description:

- context: context
- name:The name of the App Market, it will be demonstrated on left-top of the market

If you do not call the API, it will show the default name: “Market”

2. Enable the Facebook Ad

```
public static void enableFacebookAdInMarket(Context context,
 String placementId)
```

Parameter Description:

- context: context
- placementId: facebook placement Id

Note: Before Calling this Api, please make sure that you already included Latest Facebook SDK or using adsdk_2.0_with_fb.jar. You can find the detail information in part 2.2

3. Enable the Admob Ad

```
public static void enableAdmobInMarket(Context context, String unitId)
```

Parameter Description:

- context: 上下文
- placementId: Admob unitId

Note: Before Calling this Api, please make sure that you already included Latest Google Play Service Library. You can find the detail information in part 2.2

4. Creating a shortcut for Market

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

Parameter Description:

- context: context
- marketName: Market icon in bitmap
- marketName: Market name

Customizing Pictures:

We have provided nine background pictures, you can choose the one you like to be the header background picture of each page, please notice that you need to name them exactly the same as the original picture in assets directory we provided(including picture format) . You can also design the ui yourself, but please remember to keep the same size and name with the original one.

6.2 Show App Market

There are two ways to show App Market

1. Show Activity
You can call the API whenever you need to show the App Market

```
public static void showAppMarket(Context context, String marketSourceId)
```

Parameter Description:

- context: context
- marketSourceId: Traffic source Id of market

Note: Please make sure you already have called Adsdk.initialize(context) in your Applicaton

2. Show Fragment

You need to call functions below before show Market Fragment

```
public static void setMarketSourceId(Context context, String marketSourceId)
```

Parameter Description:

- context: context
- marketSourceId: Traffic Source Id of market

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

Parameter Description:

- context: context
- isFragmentMode: set to true if you use fragment mode

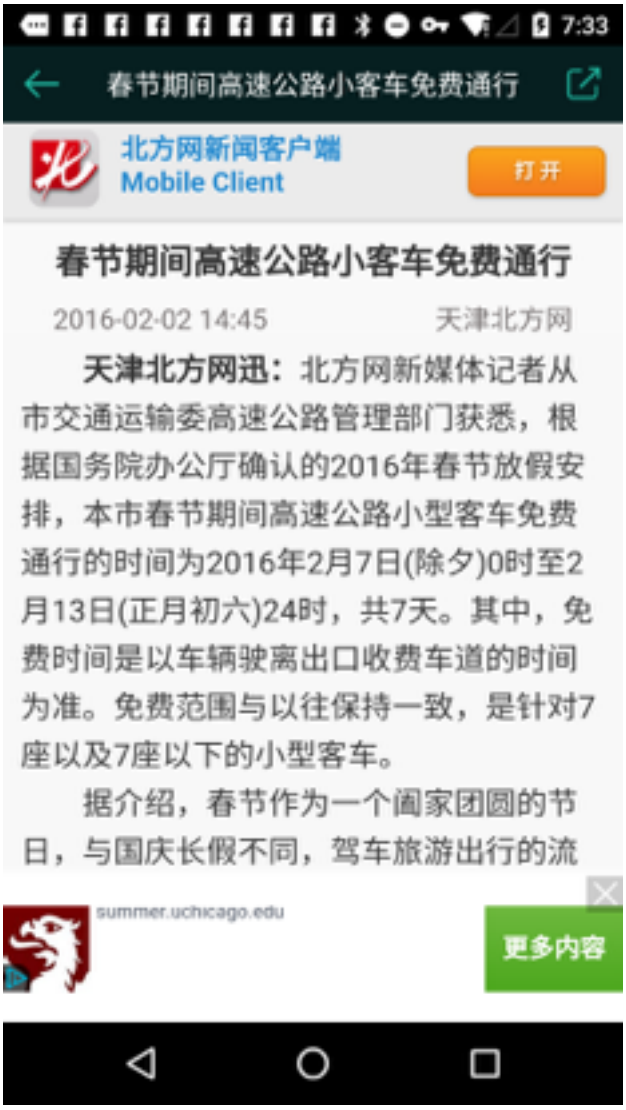
Get Fragment object by:

```
Fragment fr = new FeatureFragment()
```

7. News Feed

You can get the News Feed Page through a very simple API, please check the UI in below picture

New List Page



New Detail Page

7.1 Configuration

1. Enable the Facebook Native Ad in News Feed

```
public static void enableFacebookAdInNewsFeed(Context context,
String placementId)
```

Parameter Description:

- context: context
- placementId: facebook placement Id

2. Enable the Facebook Banner Ad in News Feed

```
public static void enableFacebookBannerInNewsFeed(Context context,
String placementId)
```

Parameter Description:

- context: context
- placementId: facebook placement Id

3. Enable the Facebook Interstitial Ad in News Feed

```
public static void enableFacebookInterstitialInNewsFeed(Context context,
String placementId)
```

Parameter Description:

- context: context
- placementId: facebook placement Id

Note: Before Calling this Api, please make sure that you already included Latest Facebook SDK or using adsdk_2.0_with_fb.jar. You can find the detail information in part 2.2

4. Enable the APX Native Ad in News Feed

```
public static void enableApxNativeAdInNewsFeed(Context context,
String newsSourceId)
```

Parameter Description:

- context: context
- newsSourceId: Traffic Source Id of news

7.2 Show News Feed

There are two ways to show App Market

1. Show Activity
You can call the API whenever you need to show the News Feed Page

```
public static void showNewsFeed(Context context, boolean newTask)
```

Parameter Description:

- context: context
- newTask: set to true if you start the news activity not from an Activity

Note: Please make sure you already have called Adsdk.initialize(context) in your Application

2. Show Fragment

You need to call functions below before show News Fragment

```
public static void setNewsFeedFragmentMode(Context context,
boolean fragmentMode)
```

Parameter Description:

- context: context
- fragmentMode: set to true if you use fragment mode

Get Fragment object by:

```
Fragment fr = new NewsTabFragment()
```