

Android SDK

Installation

Here's how to install the SDK as a library in your app.

Step 1. Get the code

First, get the code from below:

Step 2. Import our SDK as a library

If you just want to use our SDK, import the sdk directory into your workspace as a library using the Android IDE.

Step 3. Edit App Permissions

Edit your Android manifest to include the permissions needed by this app:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.djaxdemoapp"

    android:versionCode="1" android:versionName="1.0">
    <application android:icon="@drawable/ic_launcher"

        android:label="@string/app_name"
        android:allowBackup="true"

        <activity android:label="@string/app_name"
            android:name="com.example.djaxdemoapp.MainActivity"
            android:theme="@style/Theme.Sherlock">
            <intent-filter>

                <action android:name="android.intent.action.MAIN"/> <category
                    android:name="android.intent.category.LAUNCHER"/>

            </intent-
            filter> </activity>
        </application>
```

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_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.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
</manifest>
```

- **INTERNET** (required) - Grants the SDK permission to access the internet.
- **ACCESS_NETWORK_STATE** (required) - Grants the SDK permission to check for a live internet connection.
- **ACCESS_WIFI_STATE** (required) - Grants the SDK permission to access information about Wi-Fi networks.
- **WRITE_EXTERNAL_STORAGE** (recommended) - Grants the SDK permission to read from external storage.
- **ACCESS_FINE_LOCATION** (recommended) - Grants the SDK permission to access a more accurate location based on GPS.
- **ACCESS_COARSE_LOCATION** (recommended) - Grants the SDK permission to access approximate location based on cell tower.
- **READ_PHONE_STATE** (recommended) - Grants the SDK permission to read only access to phone state.
- **READ_CONTACTS** (recommended) - Grants the SDK permission to read the user's contacts data.
- **GET_ACCOUNTS** (recommended) - Grants the SDK permission to the list of accounts in the Accounts Service.

Location permissions can help monetization

Although not technically required, the *LOCATION permissions make it possible for the SDK to send location-based data to advertisers. Sending better location data generally leads to better monetization. Please note that the SDK will never wake up the phone to request the location to be updated; this would take time and battery. Instead, it will use these permissions to access the last known location of the device.

Show Ads

This section describes some of the code you'll write in order to show ads.

This document refers to something called a "zone ID". A zone ID is just a numeric ID used by MSDK to identify a context within an app where advertisements can be shown. You'll need to obtain a zone ID from your MSDK representative or your ad network. Without it, you won't be able to fetch and display ads.

Banners

You can configure your banner ad view using Java, XML, or a mixture of the two. The table below lists the XML and Java equivalents.

XML	Java Equivalent	Description	Example
msdk:zone_id	ad.setZoneid()	The zone ID associated with your app's inventory. You must include a zone ID or an error will be thrown.	"1234"
msdk:auto_refresh_time	ad.set_Auto_refresh_time()	The interval, in milliseconds, at which the AdView will request	"30000"

		new ads, if autorefresh is enabled. The minimum period is 15 seconds. The default period is 30 seconds. Set this to 0 to disable autorefresh.	
msdk:ad_width	ad.setAd_width()	The width of the advertisement to request from the server. If this is not set, the SDK requests an ad of at least android:layout_width	"320dp"
msdk:ad_height	ad.setAd_height()	The height of the view.	"50dp"
msdk:layer_style	ad.setLayer_style()	layerstyle=simple or geocities or cursor or floater	"geocities"
msdk:padding	ad.setPadding()	padding,ex:padding=2px	"2px"
msdk:align	ad.setAlign()	align=left or align=right or align=center	"left"

If you're using XML, you'll need to add the [xmlns:msdk](#) namespace attribute describing your application to your layout tag; for example this might be a RelativeLayout, LinearLayout, or FrameLayout and ScrollView.

[xmlns:msdk="http://schemas.android.com/apk/res-auto"](#)

Get the Code

First, you'll need to set up a AdView. The only required method is setZoneId, but it's usually a good idea to set the ad size. Note that the height and width you specify here must match the size of the ad zone. The code below shows a banner ad (assuming that your zone ID points at a 300x50 zone).

This simple example doesn't take advantage of all of the capabilities provided by the SDK – for example, you can also pass in the user's age and gender, as well as whether an ad click should open the device's native browser.

Sample code :

Xml Format:

```
<com.dragonmedia.adserver.AdView
android:id="@+id/adView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="20dp"
msdk:zone_id="115" //Place Your Zone id
msdk:ad_width="468" //Place Your Ad width
msdk:ad_height="60" //Place Your Ad Height
msdk:auto_refresh_time="30"> //Interval in
Milliseconds </com.dragonmedia.adserver.AdView>
```

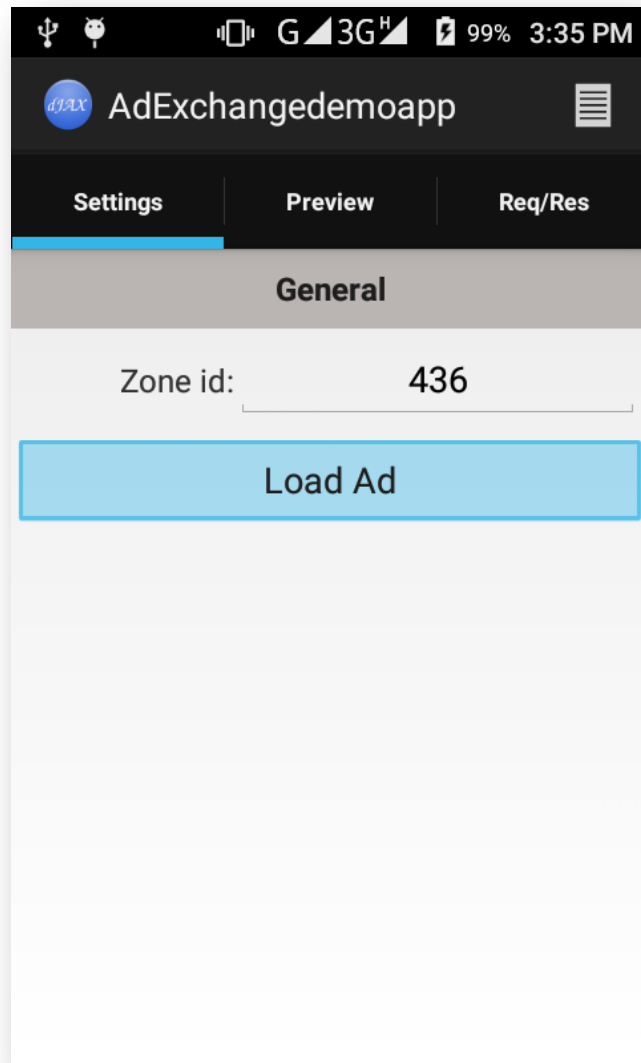
Code Format:

```
LinearLayout.LayoutParams params = new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,LinearLayout.Layout
Params.WRAP_CONTENT);

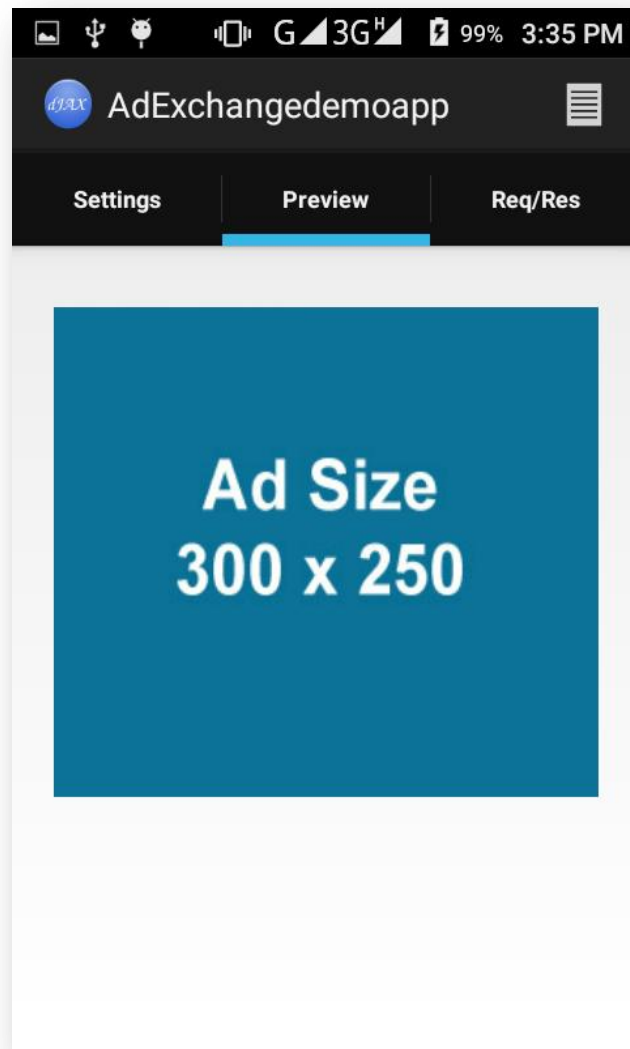
LinearLayout adFrame = (LinearLayout)
findViewById(R.id.preview);
AdView ad = new AdView(this);
ad.setZoneid("436"); //Place Your Zone id
ad.setAd_width("300"); //Place Your Ad width (Optional)
ad.setAd_height("250"); //Place Your Ad Height (Optional)
ad.setAuto_refresh_time(45000); //Interval in Milliseconds
/*Loads an Ad*/ ad.LoadAd();
ad.setLayoutParams(params);
adFrame.addView(ad);
adFrame.bringToFront();
```

After Installation:

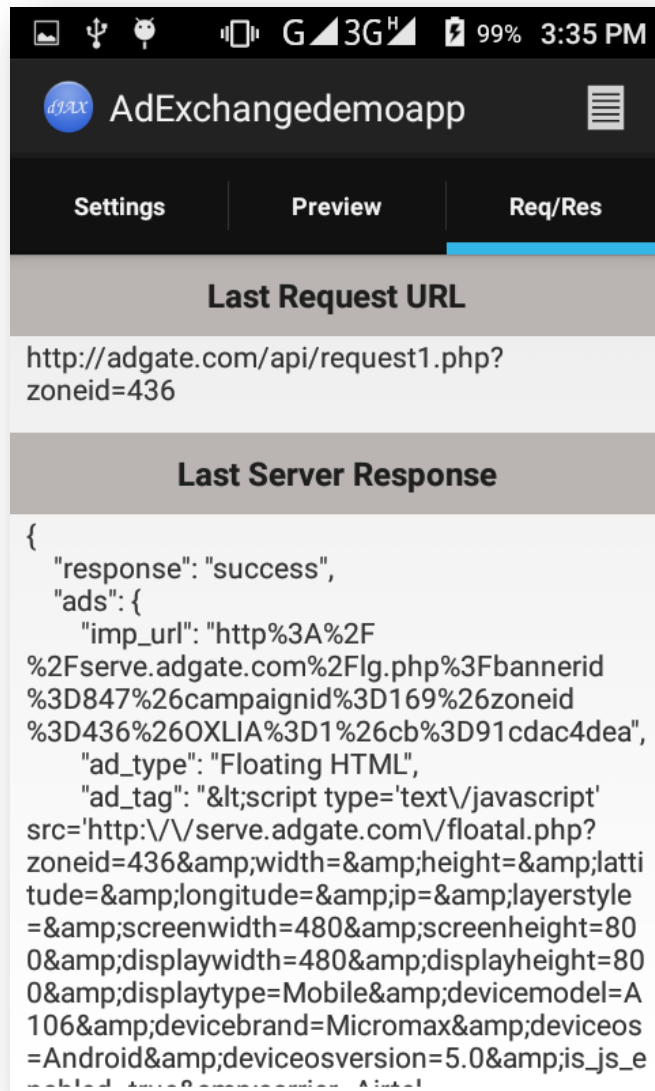
1. Enter the zone id then click load ad button.



2. The requested banner is loaded in the preview tab as shown below.



3. You can view the request and response in the request tab.



4. The Log of the app can be viewed by clicking on log icon in menu.



5. You can email the logs by clicking on the Email Logs button.