EpomSDK for Android

Epom offers themost advanced real-time optimization platform for mobile advertising campaigns, which allows developers to get maximum income from running ads on their mobile apps.

Epom SDK for Android - integration step by step

Epom offers the most advanced real time optimization platform for mobile advertising campaigns, which allows developers to get maximum income from ads for their mobile campaigns. Epom ensures that developers are as agile as possible while retaining high performance and quality of service as well.

This manual provides a simple step by step guidance for a quick and painless integration of Epom advertising into your applications.

1. Preparation

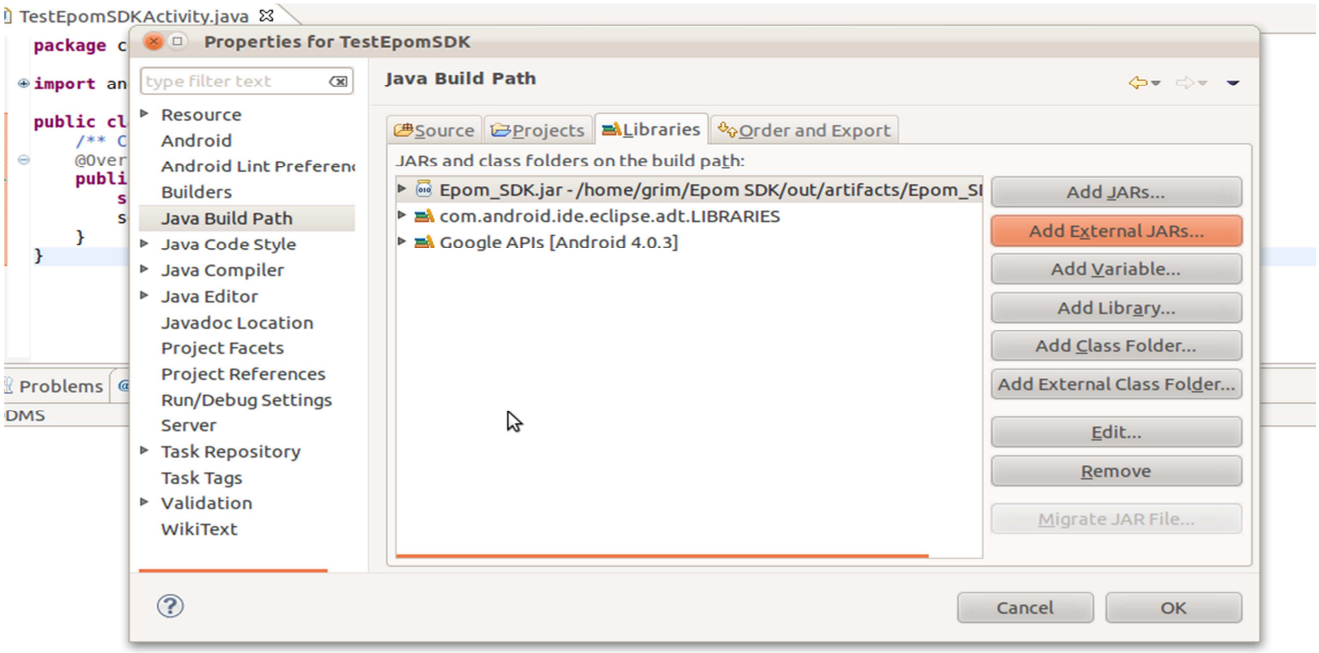
Place the Epom SDK JAR file (EpomSDK.jar) in your build path similar to any other external jar that you might already have included into your project. This is most likely to be found under the "libs" subdirectory of your project's root directory.

For Eclipse IDE projects steps are the following:

- Go to the Properties tab of your project.
- Select "Java Build Path" from the panel on the left side.
- Select "Libraries" tab in the main window.
- Click "Add External JARs" button.
- Select the JAR (EpomSDK.jar).
- To add the SDK to your Android project, Click "OK".
- Go to "Order and Export" tab from the main window
- Check the EpomSDK.jar.

Epom SDK has the following dependencies:

- Android Support v4 library



2. Setting permissions in Manifest.xml file

Please add the following lines to the application's manifest.xml file, before the closing *</manifest>*tag:

<uses-permissionandroid:name="android.permission.INTERNET"/>
<uses-permissionandroid:name="android.permission.READ_PHONE_STATE"/>

INTERNET	Allows application to use network connection sockets. Make sure the INTERNET permission is enabled to allow the SDK to access the network and exchange ads with ad servers.
READ_PHONE_STATE	Allows application to access the phone state in read only mode.This permission is enabled for getting the device ID, which allows app to benefit from highly targeted campaigns.

You can enable location targeting in your app, which means configuring your app to receive location-based ads. Enable the optional *ACCESS_COARSE_LOCATION* and *ACCESS_FINE_LOCATION* permissions to allow the SDK to access the user location details.

ACCESS_COARSE_LOCATION	Allows an application to access coarse (Cell-ID, Wi-Fi) location.

ACCESS_FINE_LOCATION	Allows an application to access fine (GPS) location.

ACCESS_NETWORK_STATE permission allows to detect network connectivity and in this way enable ad rotation in your app.

ACCESS_NETWORK_STATE	Allows an application to view the state of all networks.
WRITE_EXTERNAL_STORAGE	Allowsanapplicationtowritetoexternalstorage

With all the above your final AndroidManifest.xml may look something like this:

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

Also, you need to specify Epom synchronization service in your application's manifest file, somewhere between *<application>*and *</application>* tags:

```
<serviceandroid:name="com.epom.android.synchronization.EpomSynchronizer"/>
```

The below lines need to be added, too, because they are required by AdMob and InMobi mobile advertising networks:

```
<activity android:name="com.inmobi.androidsdk.IMBrowserActivity"
        android:configChanges="keyboardHidden|orientation|keyboard"/>

<activity android:name="com.google.ads.AdActivity"
      android:configChanges="keyboard|keyboardHidden|orientation|screenLayout
      |uiMode|screenSize|smallestScreenSize"/>
```

When using **Millenial Media** the following needs to be added:

```
<activity android:name="com.millennialmedia.android.MMAdViewOverlayActivity"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"
        android:configChanges="keyboardHidden|orientation|keyboard" />
<activityandroid:name="com.millennialmedia.android.VideoPlayer"
        android:configChanges="keyboardHidden|orientation|keyboard" />
```

When using **Inneractive** as ad source add following to Your Android Manifest .xml file:

```
<activityandroid:name="com.inneractive.api.ads.InneractiveFullScreenView" />
```

When using **TapIt** as ad provider add following into Android Manifest .xml before </application>:

```
<activity android:name="com.tapit.adview.AdActivity"
        android:config Changes="keyboard|keyboardHidden|orientation"/>
```

In case of using Tapjoy as ad content provider next lines needed to be added:

```
<activity android:name="com.tapjoy.TJCOffersWebView"
android:configChanges="keyboardHidden|orientation" />
<activity android:name="com.tapjoy.TapjoyFeaturedAppWebView"
android:configChanges="keyboardHidden|orientation" />
<activity android:name="com.tapjoy.TapjoyVideoView"
android:configChanges="keyboardHidden|orientation" />
```

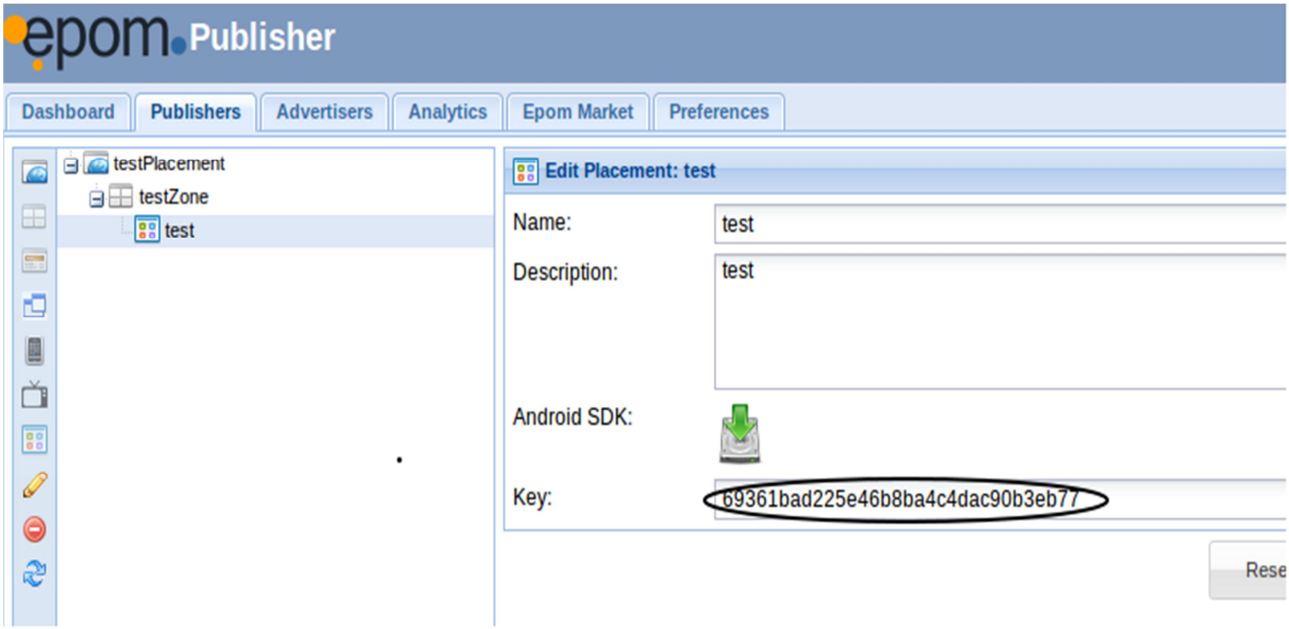
Given all the above, your sample application's Manifest.xml file is likely to look like this:

```
<?xmlversion="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.epom.test"
    android:versionCode="3"
    android:versionName="4.5.0">
    <application android:label="@string/app_name" android:debuggable="true">
    <activity android:name=".TestEpomSDK"
        android:label="@string/app_name"
        android:configChanges="keyboardHidden|orientation|keyboard">
        <intent-filter>
            <actionandroid:name="android.intent.action.MAIN"/>
            <categoryandroid:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity android:name="com.inmobi.androidsdk.IMBrowserActivity"
        android:configChanges="keyboardHidden|orientation|keyboard" />
    <activity android:name="com.google.ads.AdActivity"
        android:configChanges="keyboard|keyboardHidden|orientation|screenLayout
        |uiMode|screenSize|smallestScreenSize"/>
    <service android:name="com.epom.android.synchronization.EpomSynchronizer" />
    </application>
    <uses-sdkandroid:minSdkVersion="3"/>
    <uses-permissionandroid:name="android.permission.INTERNET"/>
    <uses-permissionandroid:name="android.permission.READ_PHONE_STATE"/>
    <uses-permissionandroid:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permissionandroid:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permissionandroid:name="android.permission.ACCESS_NETWORK_STATE"/>
</manifest>
```

3. Obtaining your Epom key

In order to ensure best experience you should disable refreshing optionin any external ad providers that you may use.

Make sure you have your key on hand for the rest of the implementation, taken from our website or provided by your personal account manager. Epom key is essential entity that identifies you within the system. It is the only mandate prerequisite for getting started as EpomSDK user.



If you have some trouble locating your Epom key or any questions about it, please contact us at *support@epom.com*.

4. Integration

To make the integration process more flexible and as easy as possible, we are offering two ways of adding the Epom View to your application:

1. As a tag in the layout XML file.
2. As API for forming and displaying ad viewing element from your code at runtime.

Way #1:Declarative description in layout file.

In order to add EpomView to your layout:

a) Add attributes set in the XML layout for this view into res/values (attrs.xml).Paste the following configuration parameters into your attrs.xml file:

```
<resources>
    <declare-styleable name="com.epom.android.configuration">
        <attr name="key" format="string"/>
        <attr name="refreshInterval" format="integer"/>
        <attr name="adType" format="string"/>
        <attr name="viewBackground" format="string"/>
    </declare-styleable>
</resources>
```

Here are the **mandatory** parameters you **MUST** add to your layout file:

key	the Epom key provided during the registration process

Here are the **optional** parameters you can add to your requests:

refreshInterval	Time frequency of ads refreshing, in seconds. The minimum refresh interval is 15 seconds, and the maximum is 90 seconds, which is 1.5 minutes. In case this parameter is omitted or entered value is out of specified bounds, value of 15 seconds is being applied. IN ORDER TO ENSURE BEST USER EXPERIENCE WE COMPEL YOU TO DISABLE ALL EXTERNAL REFRESH MECHANISMS.
adType*	The type of the ad you want to display. Currently supported values are: **BANNER_320X50** **BANNER_300X250** **BANNER_468X60** **BANNER_728X90** **BANNER_120X600**
viewBackground	Backgroundcolorofaddisplayingviewcould be specified by means of a text equivalent for desired color. *app:viewBackground = "magenta"* Defaultstoblackwhennotspicifiedexplicitly.

* In case if some outer ad network doesn't support some format, it rollbacks to default **banner_320X50.**

b) Insert a reference into the *attrs.xml* file in your layout element:

Insert a reference to the attrs.xml file in your layout element by adding xmlns line that includes your package name specified in AndroidManifest.xml. For example, if your package name is com.epom.sample, you would include this line:

xmlns:app= "http://schemas.android.com/apk/res/com.epom.sample".

c) Add EpomView to the layout:

```
<com.epom.android.view.EpomView
    android:id="@+id/epomView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```
        app:key="49d835910115c8523f80db8405b77ae9"
        app:refreshInterval="20"
        app:adType="BANNER_320X50"/>
```

Your sample application may look something like this:

```
<?xmlversion="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res/com.epom.sample"
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
        <TextView
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="@string/intro"
                android:padding="4dip"
                android:textSize="16sp"        />
        <com.epom.android.view.EpomView
                android:id="@+id/epomView"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                app:key="49d835910115c8523f80db8405b77ae9"
                app:refreshInterval="20"
                app:adType="BANNER_320X50" />
        <Button
                android:text="@+id/Button01"
                android:id="@+id/Button01"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content">
        </Button>
</LinearLayout>
```

Way #2: Imperative creating at runtime.

In order to create EpomView element programmatically:

a) Prepare configuration parameters map:

```
Map<ParamType , String>params = newHashMap<ParamType, String>();
params.put(ParamsType.KEY,"49d835910115c8523f80db8405b77");
params.put(ParamsType.REFRESH_INTERVAL,"25");
```

Here are all the parameters that you can use:

*ParamsType.**KEY***- Epom user key.
*ParamsType.**REFRESH_INTERVAL***- time for ad refresh.
*ParamsType.**ADTYPE***- specific ad type.

b) Create *EpomView* object using one of the following constructors:

public EpomView(Context context);
public EpomView(Context context, AttributeSetattrs);
public EpomView(Context context, AttributeSetattrs, intdefStyle);

c) Assign created parameters map to EpomView object:

epomView.setConfiguration(params);

d) Place created epomView somewhere within your existing layout:

LinearLayoutmyLayout = (LinearLayout) findViewById(R.id.main);

myLayout.addView(epomView);

EpomView element starts requesting for ads right after creation. To suspend ad requests use *pause()* and *launch()* methods of *EpomView* class.

6. Questions

We will be glad to help you, anytime. So if you have any questions regarding the integration, send us an email to *support@epom.com* or visit our support section at www.epom.com/support.