



InMobi Android SDK

Version a300

Installation and Integration Guide for Developers

Table of Contents

1	InMobi Android SDK Integration	3
1.1	Setting Up the Environment.....	3
1.2	Step 1 – Including the InMobiAndroidSDK.jar lib	4
1.3	Step 2 – Android Manifest Changes.....	5
1.3.1	Mandatory Activity	5
1.3.2	Mandatory Permission	5
1.3.3	Optional Permissions	5
1.3.4	Sample Manifest File	6
1.4	Step 3a – Option 1 – Inflating the IMAdView Instance from the Layout XML and Setting the Configuration	7
1.4.1	Adding an IMAdView Instance to Layout XML	7
1.4.2	Setting Auto Refresh.....	8
1.4.3	Setting the Test Mode	8
1.4.4	Setting the Listener.....	8
1.4.5	Hashing of UDID (Unique Device Identifier)	8
1.5	Step 3b – Option 2 – Creating the IMAdView Instance and Setting the Configuration	9
1.6	Integrating Interstitial Ads	10
1.6.1	Interstitial State Flow.....	10
1.6.2	Setting the Interstitial Listener	10
2	Troubleshooting	11
3	More Information	11

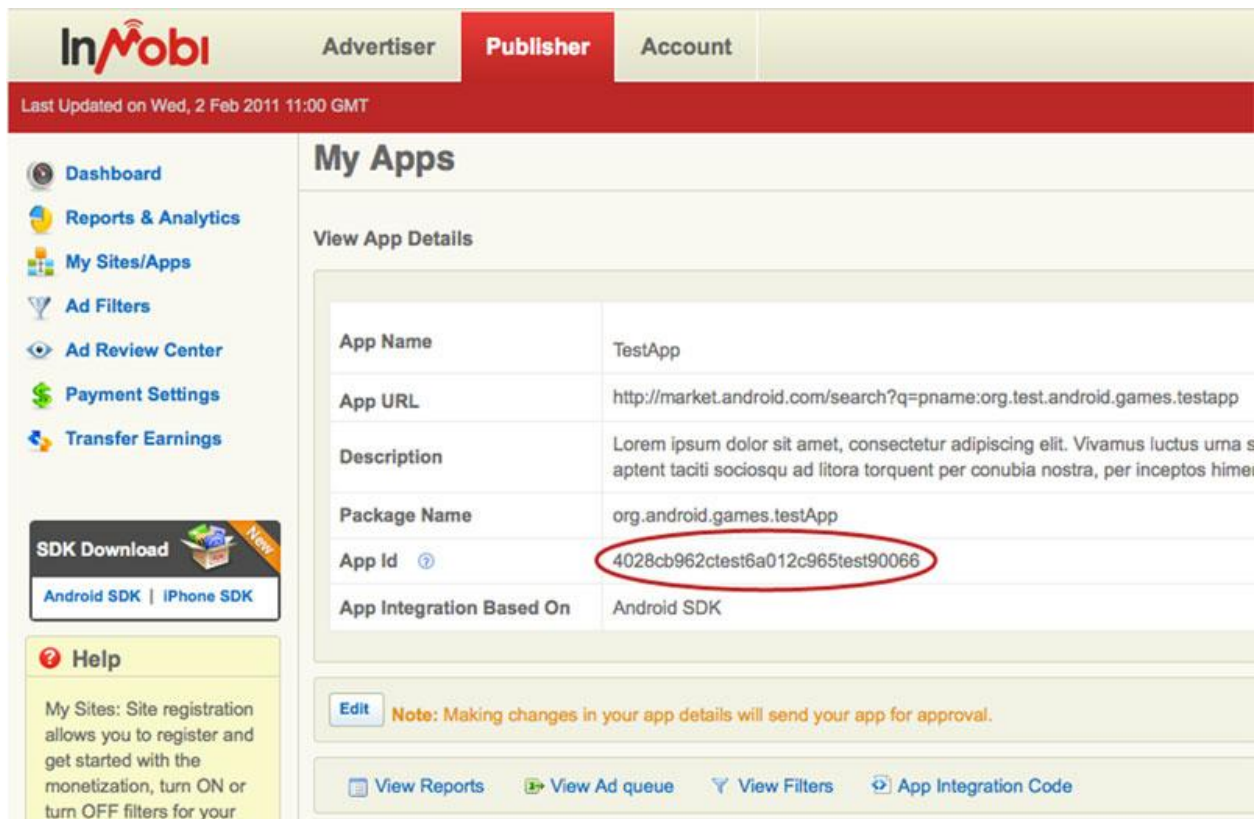
1 InMobi Android SDK Integration

The InMobi Android SDK contains the code necessary to integrate InMobi ads in your application. This SDK is designed to work across all Android platforms with a minimum OS version of 1.6.

1.1 Setting Up the Environment

Retrieve the App ID from the InMobi website by performing the following steps:

1. Login to your InMobi account at <http://www.inmobi.com>.
2. Click **My Sites/Apps** from the navigation bar.
3. Click the app name to retrieve your App ID.
4. Copy the App ID value and use it in either `IMAdView` or `IMAdInterstitial`.



The screenshot shows the InMobi Publisher Account dashboard. The top navigation bar includes 'Advertiser', 'Publisher' (selected), and 'Account'. A red banner at the top indicates 'Last Updated on Wed, 2 Feb 2011 11:00 GMT'. The left sidebar contains a navigation menu with items: Dashboard, Reports & Analytics, My Sites/Apps, Ad Filters, Ad Review Center, Payment Settings, and Transfer Earnings. Below the menu is a 'SDK Download' section with links for 'Android SDK' and 'iPhone SDK'. A 'Help' section is also visible. The main content area is titled 'My Apps' and shows 'View App Details' for an app named 'TestApp'. The details table includes fields for App Name, App URL, Description, Package Name, App ID, and App Integration Based On. The App ID field is circled in red and contains the value '4028cb962ctest6a012c965test90066'. Below the table is an 'Edit' button and a note: 'Note: Making changes in your app details will send your app for approval.' At the bottom, there are links for 'View Reports', 'View Ad queue', 'View Filters', and 'App Integration Code'.

Field	Value
App Name	TestApp
App URL	http://market.android.com/search?q=pname:org.test.android.games.testapp
Description	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus luctus urna s aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himen
Package Name	org.test.android.games.testApp
App ID	4028cb962ctest6a012c965test90066
App Integration Based On	Android SDK

Note: Developers are requested to enter their Android Market application URL in their InMobi Publisher Account.

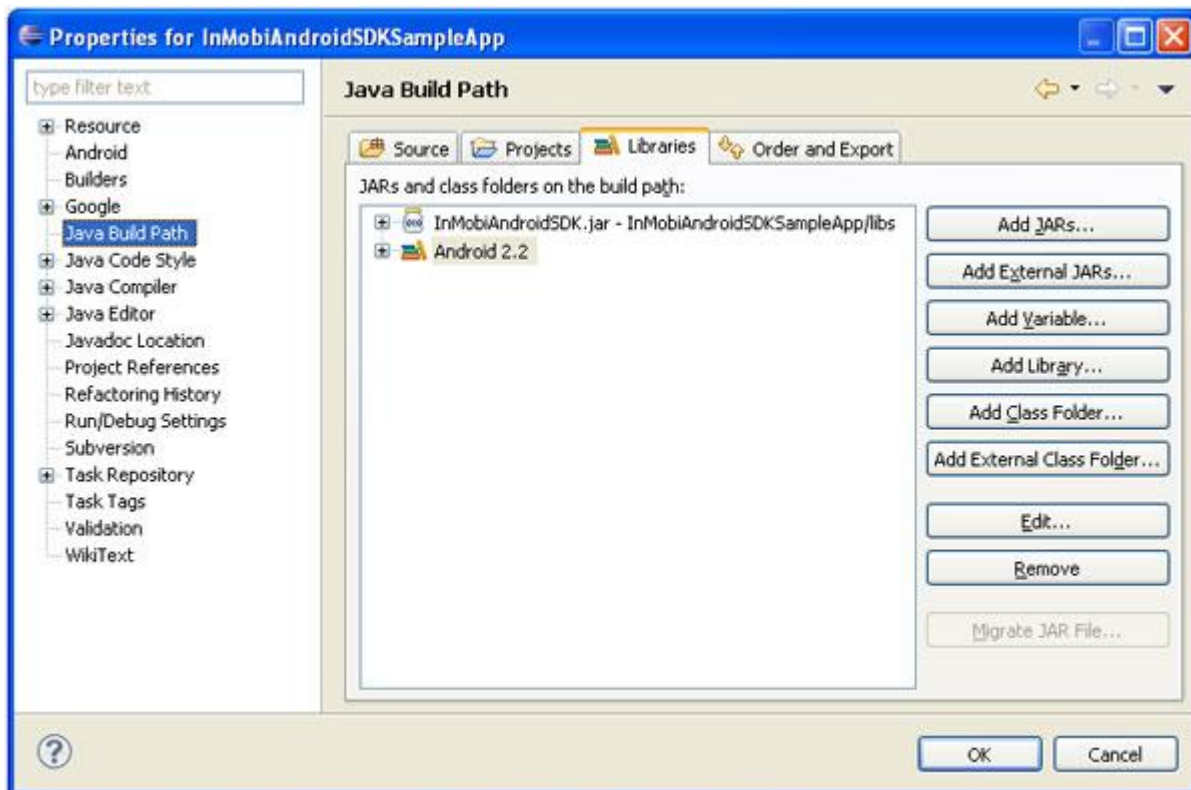
1.2 Step 1 – Including the InMobiAndroidSDK.jar lib

Copy the `InMobiAndroidSDK.jar` file to your project by performing the following steps:

1. Create a subdirectory named `libs` in the root directory of your project. This will already be done for you if you have used Android's **activityCreator** tool.
2. Copy the `InMobiAndroidSDK.jar` file into the `libs` directory.

In Eclipse, perform the following steps:

1. Right-click your project from the **Package Explorer** tab.
2. Select **Properties**.
3. Select **Java Build Path** from the left panel.
4. Select the **Libraries** tab from the main window.
5. Click **Add JARs...**.
6. Select the `InMobiAndroidSDK.jar` file you copied earlier to the `libs` directory.
7. Click **OK** to add the InMobi SDK to your Android project.



1.3 Step 2 – Android Manifest Changes

Here, you are given instructions to make changes to your Android manifest file.

1.3.1 Mandatory Activity

Ensure that you add the `com.inmobi.androidsdk.IMBrowserActivity` activity to your `AndroidManifest.xml` within the `<application>` tag. This activity will be used to open ads in the embedded browser and to display interstitial ads.

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

1.3.2 Mandatory Permission

Ensure that you add the `INTERNET` permission to your `AndroidManifest.xml` file just before the closing `</manifest>` tag:

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

1.3.3 Optional Permissions

1.3.3.1 Showing Geo-Targeted Ads

To allow InMobi to show geo-targeted ads, you need to add the `ACCESS_COARSE_LOCATION` and `ACCESS_FINE_LOCATION` permissions.

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

1.3.3.2 Getting Network State

To get the network state for targeting purposes, you need to add the `ACCESS_NETWORK_STATE` permission.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

1.3.3.3 Enabling Calls from Ads

To enable users to make calls from the ads, you need to add the `CALL_PHONE` permission.

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

1.3.4 Sample Manifest File

Your final `AndroidManifest.xml` file may look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.inmobi.sample" android:versionCode="1"
    android:versionName="1.0">

    <application android:label="@string/app_name" android:icon="@drawable/inmobi_app_new">
        <activity android:name=".InMobiAdActivity" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!-- Mandatory activity declaration to show the Ad in the embedded browser -->
        <activity android:name="com.inmobi.androidsdk.IMBrowserActivity"
            android:configChanges="keyboardHidden|orientation|keyboard" />

    </application>

    <!-- Internet is mandatory permission -->
    <uses-permission android:name="android.permission.INTERNET" />

    <!-- optional permissions to serve location based Ads -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <!-- optional permission to enable the user to make calls from Ads -->
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <!-- optional permission to get the network state for targeting purposes -->
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

1.4 Step 3a – Option 1 – Inflating the IMAView Instance from the Layout XML and Setting the Configuration

1.4.1 Adding an IMAView Instance to Layout XML

You can add an `IMAdView` to your activity through layout XML files. Add the XML snippet in your layout XML file to start receiving ads.

For example, if you want an ad with a 320x50 size, add the XML attribute shown below, to your parent `ViewGroup`:

```
<com.inmobi.androidsdk.IMAdView android:layout_width="320dp"
android:layout_height="50dp" android:id="@+id/imAdview"
appId="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" adSlot="15" />
```

When the activity is loaded, the auto Ad Refresh starts automatically.

Note: The `AppID` and `AdSlot` attributes are mandatory, without which the `IMAdView` cannot be inflated.

Select an `AdSlot` from the following set of values declared in `IMAdView`.

```
/**
 * This ad slot will be deprecated in the future. Use ad slot 15 instead.
 */
static final int INMOBI_AD_UNIT_320X48 = 9;
/**
 * Medium Rectangle size for an InMobi ad, 300x250 pixels, designed for all device
 * sizes.
 */
static final int INMOBI_AD_UNIT_300X250 = 10;
/**
 * Leaderboard size for an InMobi ad, 728x90 pixels, designed for tablet screen
 * size.
 */
static final int INMOBI_AD_UNIT_728X90 = 11;
/**
 * Full Banner size for an InMobi ad, 468x60 pixels, designed for tablet screen
 * size.
 */
static final int INMOBI_AD_UNIT_468X60 = 12;
/**
 * Skyscraper size for an InMobi ad, 120x600 pixels, designed for tablet screen
 * size.
 */
static final int INMOBI_AD_UNIT_120X600 = 13;
/**
 * Standard rich media banner size for an InMobi ad, 320x50 pixels.
 */
static final int INMOBI_AD_UNIT_320X50 = 15;
```



```
/**
```

If you want to set configuration, use the ID provided in the XML to get the `IMAdView` instance.

```
IMAdView imAdView = (IMAdView) findViewById(R.id.imAdview);
```

1.4.2 Setting Auto Refresh

After you add the `IMAdView` to the parent `ViewGroup`, the Ad Refresh will start automatically. The default refresh interval is 60 seconds.

To switch off the Auto Refresh feature, use the method shown here:

```
imAdView.setRefreshInterval(IMAdView.REFRESH_INTERVAL_OFF);
```

To manually load a new ad, use the method shown here:

```
imAdView.loadNewAd();
```

1.4.3 Setting the Test Mode

When developing your app, set the Test Mode to `true`.

```
IMAdRequest adRequest = new IMAdRequest();  
adRequest.setTestMode(true);  
imAdView.setIMAdRequest(adRequest);
```

Note: Ensure that you set the Test Mode to `false` when distributing the app to users. By default, Test Mode is set to `false`.

1.4.4 Setting the Listener

If you need ad status notifications, you need to implement the `IMAdListener` interface and register the instance with `IMAdView`.

```
imAdView.setIMAdListener(imAdListener);
```

1.4.5 Hashing of UDID (Unique Device Identifier)

By default, to protect user privacy, the UDID (Android ID) is encrypted by the SDK using MD5 hashing, which is a standard method of "one way" encryption. You can disable this as shown here:

```
IMAdRequest adRequest = new IMAdRequest();  
adRequest.setUDIDHashingAllowed(false);  
imAdView.setIMAdRequest(adRequest);
```


To create the `IMAdView` instance programmatically, perform the following steps:

- ```
IMAdView imAdView = new IMAdView(activity, IMAdView.INMOBI_AD_UNIT_320X50,
 "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx");
```

- `activity`
- `adSlot` (One of the `adSlot` constant given in the `IMAdView`)
- `appId` (Ensure that your app is in the **Activated** state in your InMobi account)

- ```
final float scale = getResources().getDisplayMetrics().density;

    int width = (int) (320 * scale + 0.5f);

    int height = (int) (50 * scale + 0.5f);
```

- ```
imAdView.setLayoutParams(new LinearLayout.LayoutParams(width, height));
```

- ```
LinearLayout parent = (LinearLayout)findViewById(R.id.linearLayoutParent);
```

- ```
parent.addView(imAdView);
```

Copyright © 2011 | All rights reserved | [www.inmobi.com](http://www.inmobi.com)



## 2 Troubleshooting

### I am not able to see ads. What do I do?

Check if you have provided a valid App ID. For integration purposes, you can make a request for ads in Test Mode. If you are not using Test Mode, check if the App ID is in the **Activated** status by logging into your InMobi Publisher Account.

### What attributes should I verify before submitting my app to the Android Market?

First, check if your requests have been set with Test Mode set to `false`.

Next, verify that the App ID provided is exactly the same as it appears in your Publisher section on [www.inmobi.com](http://www.inmobi.com).

### I get 'No Fill' response when I make a request for ads in the Test or Non-Test Mode. What should I do?

In both the Test and Non-Test modes, depending on various parameters such as server load, non-availability of targeted ads, and so on, the Ad Server may send a **No Fill** response.

Try reloading the ad after some time to keep receiving the ads. Although a **No Fill** is common in the advertising space, you can write to us with your queries.

### Why am I being served only test ads and not live ads?

Live ads will be served only after your site or app is approved by InMobi Administrators. You can use the time until approval, to test your integration using our test Ad Server.

**Note:** Testing must only be done through a mobile device browser. Ads will not be served to non-mobile device browsers from PCs and laptops.

To start testing your Android app before it gets certified, and even before you register it with us, use the following Test App ID:

**4028cba631d63df10131e1d3818b00cc**

## 3 More Information

You can find more information in the Java Docs and sample project within the ZIP file.

For any SDK integration queries, please send an email along with your Login ID to [helpdesk@inmobi.com](mailto:helpdesk@inmobi.com).