Airpush Android SDK 5.0

4 Powerful Ad Units

Seamless SDK Integration

Weekly Payments!

Table of Contents

Overview..... 1

Ad Units..... 1

- Push Notification Ads..... 1
- Icon Ads..... 1
- Smart Wall..... 2

Bugsense..... 2

Installation Instructions..... 3

- Adding the JAR..... 3
- Editing Your Manifest File..... 4
- Editing Your Main File..... 7
- Setting Test Mode..... 10

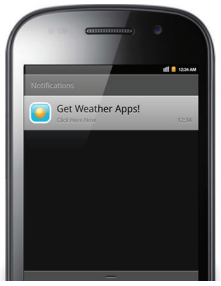
Overview

The Airpush Android SDK is a java-based library that allows publishers to monetize their app(s) with unique, innovative ad units. Developers use this SDK to monetize their entire install base, earn above industry average CPM's, and provide richer ad experiences. This document covers installation instructions, available ad units and their features, optimization, and best practices. It is written for developers with the assumption they are familiar with Android development.

Ad Units

There are 4 types of ad units available in this SDK. To enable and optimize each of these ad units, select the corresponding check box next to each ad unit in 'Step 2' of adding your app. Free weekly payments are available for each.

Push Notification Ads



A Push Notification Ad is an ad sent to the notification tray of Android devices, rather than inside of an app. This ad unit can monetize an app's entire install base- both active and inactive. CPM's range from $2-$10. You, the developer, control how often ads are sent to your users from within your account.

Best Practice: To maximize revenue set the frequency to 2 ads/day with a 0 day ad delay.

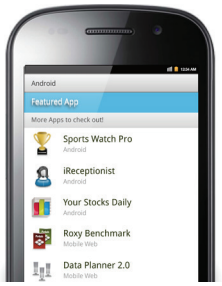
Icon Ads

This innovative ad unit is a shortcut placed on the home screen of Android devices which links to valuable content such as free apps and deals. This ad unit can also monetize an app's entire install base- both active and inactive. CPM's range from $5 - $12.

Best Practice: To maximize revenue set the frequency to the maximum setting of 5 icons per user per month.

SmartWall

SmartWall, our new full-page ad format, dynamically optimizes between multiple sub-formats including AppWall, OfferWall, Dialog Ads, Video Ads, Rich Media, and More. Benefits of this ad unit include $6+ CPM's and a rich user experience. You control when this ad unit is triggered within your app session.



Best Practice: To maximize revenue set from SmartWall, set the ad unit to launch upon app launch, app exit and in natural breaks in the app session (ie in between game levels).

Bugsense

Included in this SDK is Bugsense, the industry leading real-time crash analytics and bug tracking service. Airpush has forged an exclusive partnership with Bugsense which allows us to offer the service free to our developers. You will be notified in the dashboard of your account any time errors occur in your app. You can view your real-time analytics under the 'Crash Report' tab under Reports.

Installation Instructions

Airpush Android SDK contains the code necessary to install Airpush ads in your application. This PDF will guide you through a simple XML implementation.

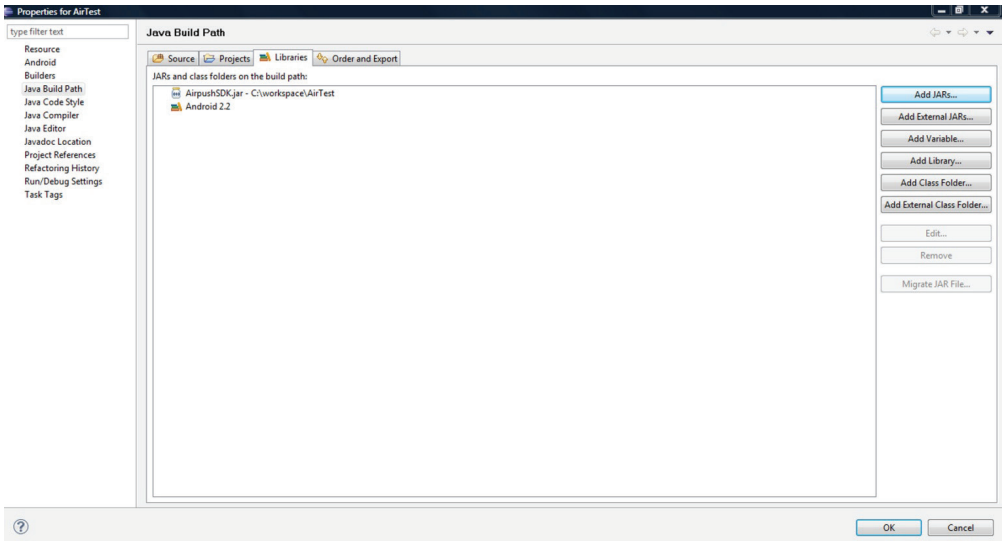
Note: You will need to replace all instances of <appid>, <package name> and <apikey> with your actual application id, package name and api key which is obtainable from Airpush portal.

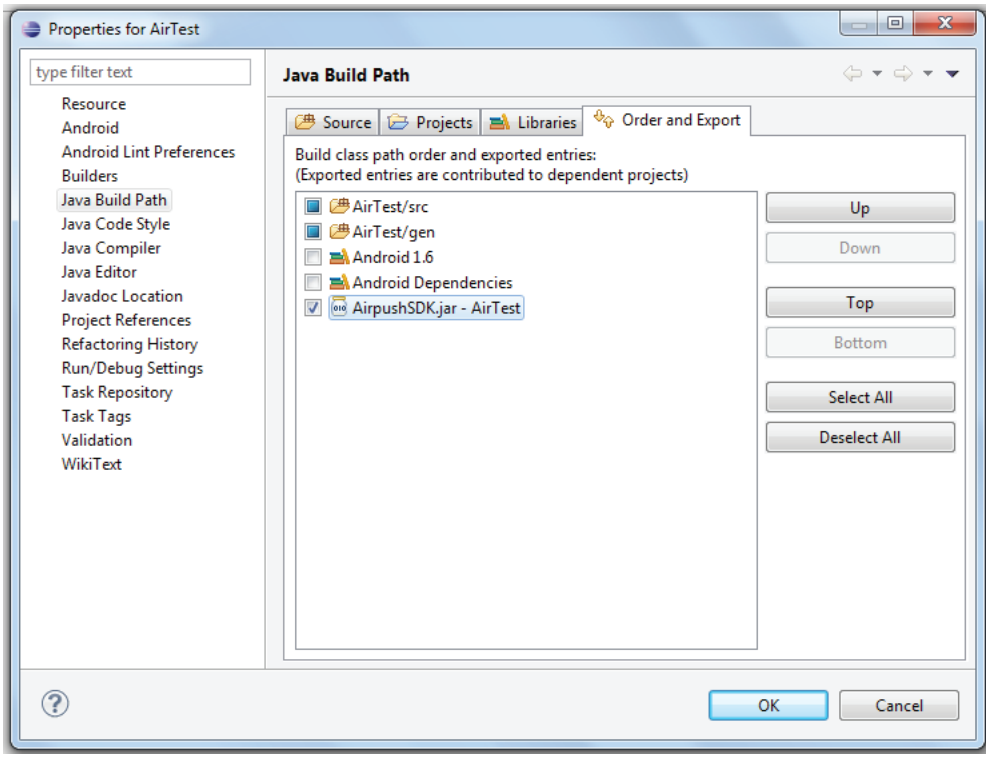
Step 1 - Adding the JAR

Copy the Airpush JAR file (AirpushSDK.jar) in your project's root directory.

For Eclipse projects:

- Right-click on your project from the Package Explorer tab and select "Properties"
- Select "Java Build Path" from the left panel
- Select "Libraries" tab from the main window
- Click on "Add JARs..."
- Select the JAR that's been copied to the project's root directory
- Click "OK" to add the SDK to your Android project
- Select "Order and Export" tab from the main window and check the SDK





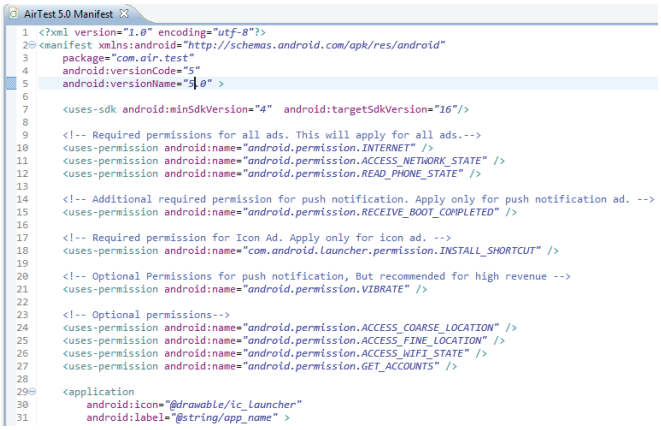
Step 2 – Editing Your Manifest File

First you will need to note your Airpush <appid> that was given to you when registering your Android application on www.airpush.com. It is a numeric code and can be found by locating your app in the apps dashboard.

Just before the closing </application> tag of your AndroidManifest.xml file, you will need to add these things:

1: Copy and paste the following XML just before the closing </application> tag:

```
<!-- Airpush Manifest declaration start -->

 <!-- Start required declaration for all ads-->

<!-- Setup your APPID and APIKEY here. Please append your APIKEY after “airpush<APIKEY>”. If you do not have APIKEY you can pass “airpush” as APIKEY. Suppose your APIKEY is “1338970568460022873” so you need to pass “airpush1338970568460022873”. -->

<meta-data android:name=”com.airpush.android.APPID” android:value=”<Your appId>” />
<meta-data android:name=”com.airpush.android.APIKEY”
 android:value=”airpush<Your ApiKey>”/>

<!-- This activity is required for all ads. -->
<activity android:exported=”false” android:name=”com.airpush.android.OptinActivity”
                android:configChanges=”orientation|keyboardHidden”
android:theme=”@android:style/Theme.Translucent” />
<!—- End of required declarations for all ads-->

<!-- Start declaration for push notification ads. -->
<service android:name=”com.airpush.android.PushService” android:exported=”false” />
<receiver android:name=”com.airpush.android.BootReceiver” android:exported=”false” >
        <intent-filter>
            <action android:name=”android.intent.action.BOOT_COMPLETED” />
             <category android:name=”android.intent.category.HOME” />
        </intent-filter>
</receiver>
<!-- End declaration for push notification ads. -->
```



2: Add The Following Permissions:

```
<!-- Required permissions for all ads. This will apply for all ads. -->
    <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” />

<!-- Additional required permission for push notification. Apply only for push
notification ad. -->
 <uses-permission android:name=”android.permission.RECEIVE_BOOT_COMPLETED” />

  <!-- Required permission for Icon Ad. Apply only for icon ad. -->
  <uses-permission android:name=”com.android.launcher.permission.INSTALL_SHORTCUT” />


<!—- Optional permission for push notification ad but highly recommended enhancing your rev-
enue stream -->
    <uses-permission android:name=”android.permission.VIBRATE” />

<!-- Optional permissions -->
    <uses-permission android:name=”android.permission.ACCESS_COARSE_LOCATION” />
    <uses-permission android:name=”android.permission.ACCESS_FINE_LOCATION” />
    <uses-permission android:name=”android.permission.GET_ACCOUNTS” />
    <uses-permission android:name=”android.permission.ACCESS_WIFI_STATE” />
```

Step 3 - Edit Your Main File

Inside onCreate() method, please add :

```
Airpush airpush=new Airpush(getApplicationContext());
```

1. To start push notification, call the following method:

```
airpush.startPushNotification(false);
```

For banner push, the developer needs to copy the airpush_notify.xml file that is included along with SDK download to project layout folder. Developer must not make any changes to that file. An image with push_icon.png name is required. It will be displayed with push notification on status bar.

2. To start icon ad: `airpush.startIconAd();`

Using SmartWall in your application:

Airpush's smart wall is comprised of following three sub Ad formats:

Dialog Ad
AppWall Ad
LandingPage Ad

Airpush's Ad serving determines and displays the best sub-Ad-Format to maximize your revenue from Interstitial Ad Placements in your application. However, you can still choose to call a specific sub-Ad-Format from above options by using following methods in your code:

1. To start dialog Ad: `airpush.startDialogAd();`

2. To start AppWall Ad: `airpush.startAppWall();`

We would highly recommend our developers initializing Smartwall on App Exit to maximize monetization. Here is the sample code for showing Airpush's Smartwall on app exit:

For individual calls use the below code:

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
                airpush.startAppWall();
                airpush.startDialogAd();
                airpush.startLandingPageAd();
                finish();
        }
        return super.onKeyDown(keyCode, event);
           }
```

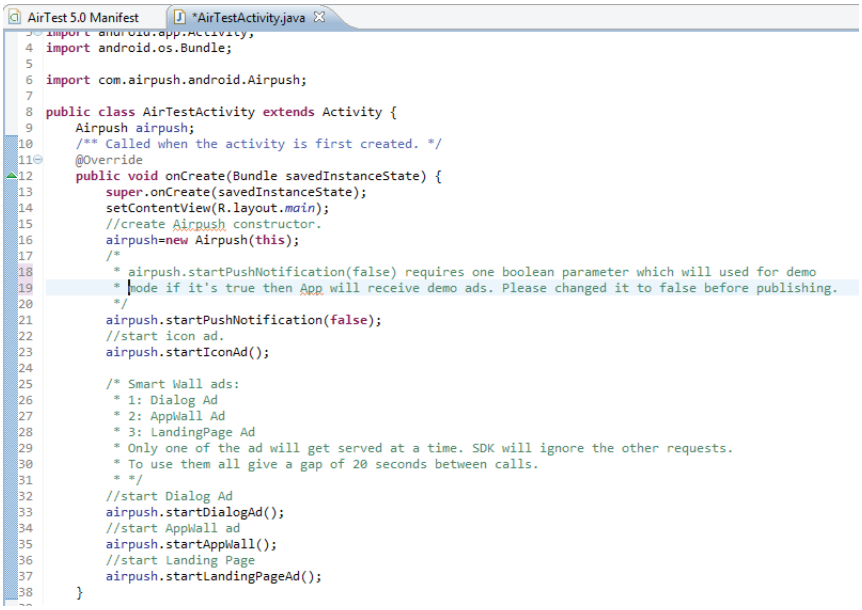
If you want to show Ad decided by SDK, then use the below code.

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
new Airpush(getApplicationContext());
                finish();
 }
        return super.onKeyDown(keyCode, event);
           }
```

Note: This code can be used in Activity file only.

If your application supports android version 1.5 or below, please initialize the SDK as given below:

```
 if(Integer.parseInt(VERSION.SDK) > 3){
   Airpush airpush=new Airpush (getApplicationContext());

airpush.startPushNotification(false);  //start push notification.
airpush.startIconAd();                 //start icon.
airpush.startDialogAd();               //start dialog ad.
airpush.startAppWall();                //start app wall.
airpush.startLandingPageAd();          //start landing page.

        }
```



Step 4 – Setting Test Mode

While integrating Airpush SDK into your application it is recommended that you use test mode. This will help you verify that your application is working fine with integrated SDK and test ads are returned every few minutes on an emulator or real device before publishing your app to Android Market. To enable test mode, please include the Boolean parameter while initializing Airpush as illustrated in the code below:

```
Airpush airpush=new Airpush(getApplicationContext());
Airpush.startPushNotification(true);
```

Once you have verified SDK installation, you can turn test mode off by removing the Boolean parameter.
You can either generate an <apikey> from permission API page or use “airpush” as API key.

Note: We have test mode for Push Notification Ads only.

For better ratings and user experiences add this disclaimer to your app’s market description:

“This free app is ad supported and may contain ads in the notification tray and/or home screen.”

Sample Application Code and Support:

Included with this SDK is an Airtest Example Project (Airtest.rar) and in case of any issues integrating SDK, please feel free to contact publishersupport@airpush.com

Note: For the privacy of your users, Airpush never stores personally identifiable information. Our SDK encrypts all IMEI numbers using md5 hashing.