

# RedTroops SDK for Android

Requirements: Android 2.3.3+ (API 10)

## Getting Started

RedTroops SDK currently features Push Notifications, HTML5/Image Popups, and Banner Lists. In order to have the Push Notifications ready for RedTroops, you must set up Google Cloud Messaging.

## Setting Up Google Cloud Messaging (Push Notifications)

Please refer to these instructions by Google from developer.android.com. Instead of obtaining an Android key, select Server Key (No need to modify any Server Key options):

<http://developer.android.com/google/gcm/gs.html>

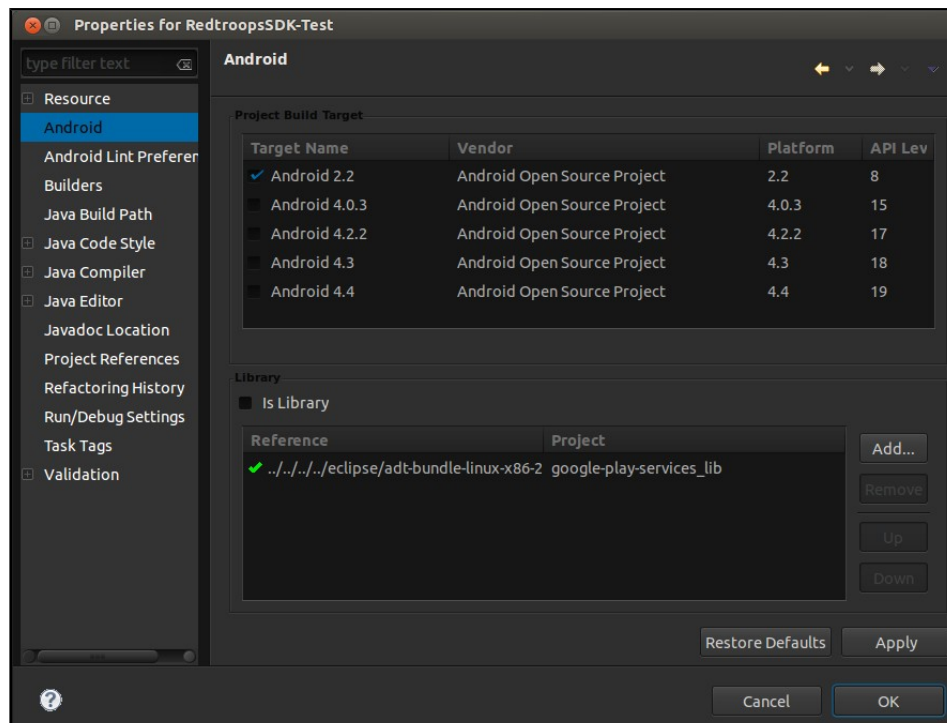
A Project Number (Sender ID) will be generated which you will use to set up the SDK, store it.

Next, you have to set up Google Play Services. Please refer to these instructions:

<http://developer.android.com/google/play-services/setup.html>

You must have now Google Play Services library in your project. You must have a project number. There should be no errors while compiling.

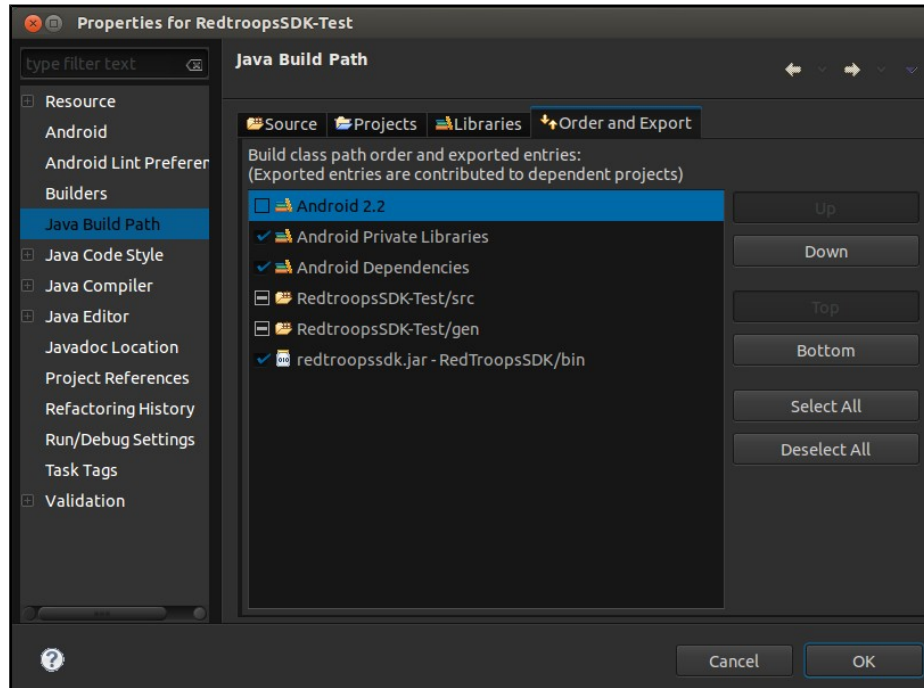
Android Support Library must be added. This can be done by right-clicking on your project → Android Tools → Add Support Library. Android Private Libraries must be checked in Order and Export.



## Setting Up RedTroops SDK In Your Project

Follow the steps below to get your RedTroops SDK running:

- 1) Download the SDK from RedTroops' website.
- 2) Right-click on your project from the Package Explorer in Eclipse → Build Path → Configure Build Path.
- 3) Click on Add External JARs, and choose RedTroopsSDK.
- 4) Go to Order and Export tab, and place a check for RedTroopsSDK.



- 5) Edit your manifest as follows:

Add the following permissions (Mandatory), change <PACKAGE-NAME> into your app's package name:

```
<!-- Permissions for RedTroops SDK-->
<permission
    android:name="<PACKAGE-NAME>.permission.C2D_MESSAGE"
    android:protectionLevel="signature" />
<uses-permission android:name="<PACKAGE-NAME>.permission.C2D_MESSAGE" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

<!-- GET_ACCOUNTS permission is only needed if the minSdkVersion is lower than
14 (4.0.4), you may remove it otherwise -->
<uses-permission android:name="android.permission.GET_ACCOUNTS" />

<uses-permission android:name="android.permission.INTERNET"/>
<!-- End of Permissions for RedTroops SDK-->
```

In application tag, add the following activities, receiver, service, and meta-data (Mandatory):

```
<!-- RedTroops SDK (MANDATORY) -->
<activity android:name="com.RedTroops.RedTroopsSDK.RedTroopsNagActivity"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.RedTroops.RedTroopsSDK.RedTroopsMoreAppActivity"
    android:screenOrientation="portrait"/>

<!-- GCM Receiver for RedTroops SDK -->
<receiver
    android:name="com.RedTroops.RedTroopsSDK.gcm.GcmBroadcastReceiver"
    android:permission="com.google.android.c2dm.permission.SEND" >
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />

        <category android:name="<PACKAGE-NAME>" />
    </intent-filter>
</receiver>

<service android:name="com.RedTroops.RedTroopsSDK.gcm.GcmIntentService" />

<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />

<!-- End of RedTroops SDK (MANDATORY) -->
```

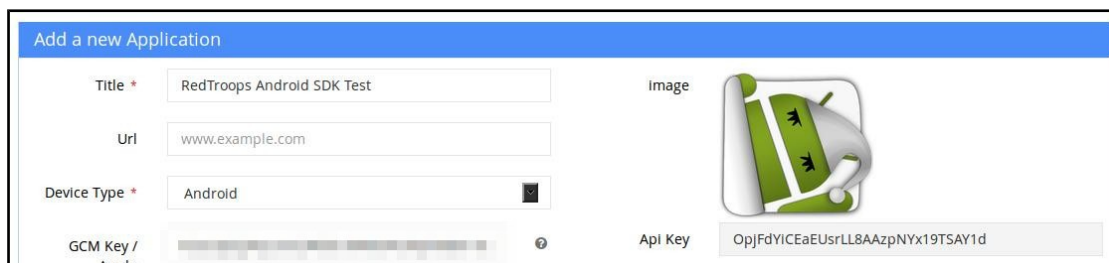
6) Go to your assets folder, and create a new file. Name it “redtroops.properties”. Add the following keys and their corresponding values. For example:

```
gcm_sender_id=330435028307
api_key=R0mLGV9uDFq875yT26L4omko27fYZ8nG
app_id=4
```

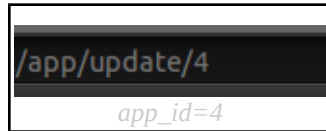
*gcm\_sender\_id* is the Project Number that you obtained from Google Cloud Console when you created the project.



*api\_key* is the API key that you obtained from RedTroops developer website when you added the application.



`app_id` is the application id. You can obtain it from the address bar when you are editing the App.



7) In your main activity's `OnCreate`, call:

```
RedTroopsSDK.getInstance(this).init(initFinishedListener);
```

Where `initFinishedListener` is a listener to declare as follows:

```
private initFinishedListener initFinishedListener = new initFinishedListener() {  
  
    @Override  
    public void onSuccess() {  
        // TODO Do on init success. Most probably showHTML5ImagePopup();  
    }  
  
    @Override  
    public void onFail() {  
        // TODO Do on init failure  
    }  
};
```

8) Whenever you want to show an HTML5/Image popup, call:

```
RedTroopsSDK.getInstance(this).showHTML5ImagePopup();
```

It is preferred to call this in `initFinishedListener`'s `onSuccess()` so that it is made sure that the initialization has finished.

9) Whenever you want to show the banner list, call:

```
RedTroopsSDK.getInstance(this).showBannerList();
```

10) To end your session, add the following to your last activity's `onDestroy`:

```
RedTroopsSDK.getInstance(this).endSession();
```

This should only be called once after each app run when the user is no longer using the app.

11) **Optional:** By default the icon for Push Notification is “ic\_launcher”, you may change it by calling:

```
RedTroopsSDK.getInstance(this).setPushNotificationIcon("ic_launcher");
```

The icon must be in any drawable folder.

## **Important Notes**

1. When creating a Push Notification, a custom sound can be specified. The custom sound has to be in res/raw folder. The custom sound name should not contain the file type. For example, the proper custom sound name to set is “c\_4” for the RedTroopsSDK-Test app. If the custom sound name was invalid, or if the custom sound was not found, no sound will be played.
2. The default icon for the Push Notification is ic\_launcher. If a custom icon was not set using setPushNotificationIcon and ic\_launcher was not found, the Push Notification will not appear.
3. The user will not be notified if Google Play Services app was not installed on their device.
4. Make sure that you clean and build your project each time you add a library or change settings of Order and Export tab.
5. The test app will show errors if RedTroops SDK was not added. This can be done by changing the Java Build Path as documented in Setting Up RedTroops SDK In Your Project section.

**Refer to the test app RedtroopsSDK-Test that is available alongside the SDK.**