

# Admob Tutorial

Integrating the Google Mobile Ads SDK into an app is the first step toward displaying ads and earning revenue. Once you've integrated the SDK, you can choose an ad format and follow the steps to implement it. This guide is for publishers who want to monetize an Android app with AdMob and aren't using Firebase. For simplicity, I am not using Firebase.

## Prerequisites

- Use Android Studio 1.0 or higher
- Target Android API level 14 or higher
- Recommended: Create a Google AdMob account and register an app.

## Import the Mobile Ads SDK

Apps can import the Google Mobile Ads SDK with a Gradle dependency that points to Google's Maven repository. In order to use that repository, you need to reference it in the app's project-level build.gradle file. Open yours and look for an allprojects section:

Example project-level build.gradle (excerpt)

```
allprojects {
    repositories {
        google()
        jcenter()
    }
}
```

Add the google() directive above if it's not already present.

Next, open the app-level build.gradle file for your app, and look for a "dependencies" section.

Example app-level build.gradle (excerpt)

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.google.android.gms:play-services-ads:17.2.1'
}
```

Add the line in bold above, which instructs Gradle to pull in the latest version of the Mobile Ads SDK. Once that's done, save the file and perform a Gradle sync.

## Update your AndroidManifest.xml

Add your AdMob App ID to your app's AndroidManifest.xml file by adding the <meta-data> tag shown below. You can find your App ID in the AdMob UI. For android:value insert your own AdMob App ID in quotes, as shown below.

```

<manifest>
  <application>
    <!-- Sample AdMob App ID: ca-app-pub-3940256099942544~3347511713 -
    -->
    <meta-data
      android:name="com.google.android.gms.ads.APPLICATION_ID"
      android:value="[ADMOB_APP_ID]"/>
    </application>
  </manifest>

```

## Initialize MobileAds

Before loading ads, have your app initialize the Mobile Ads SDK by calling `MobileAds.initialize()` with your AdMob App ID. This needs to be done only once, ideally at app launch.

Here's an example of how to call the `initialize()` method in an Activity:

Example MainActivity (excerpt)

```

package ...
import ...
import com.google.android.gms.ads.MobileAds;

public class MainActivity extends AppCompatActivity {
  ...
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Sample AdMob app ID: ca-app-pub-3940256099942544~3347511713
    MobileAds.initialize(this, "YOUR_ADMOB_APP_ID");
  }
  ...
}

```

## Select an ad format

The Mobile Ads SDK is now imported and you're ready to implement an ad. AdMob offers a number of different ad formats, so you can choose the one that best fits your app's user experience.

**Banner:** Banner ads are rectangular image or text ads that occupy a spot within an app's layout. They stay on screen while users are interacting with the app, and can refresh automatically after a certain period of time.

## IMPLEMENT A BANNER

## Add AdView to the layout

The first step toward displaying a banner is to place AdView in the layout for the Activity in which you'd like to display it. The easiest way to do this is to add one to the corresponding XML layout file. Here's an example that shows AdView at the bottom of an Activity:

main\_activity.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

    <TextView android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <com.google.android.gms.ads.AdView
        xmlns:ads="http://schemas.android.com/apk/res-auto"
        android:id="@+id/adView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_alignParentBottom="true"
        ads:adSize="BANNER"
        ads:adUnitId="ca-app-pub-3940256099942544/6300978111">
    </com.google.android.gms.ads.AdView>

</RelativeLayout>
```

## Always test with test ads

When building and testing your apps, make sure you use test ads rather than live, production ads. Failure to do so can lead to suspension of your account.

The easiest way to load test ads is to use our dedicated test ad unit ID for Android banners:

ca-app-pub-3940256099942544/6300978111

It's been specially configured to return test ads for every request, and you're free to use it in your own apps while coding, testing, and debugging. Just make sure you replace it with your own ad unit ID before publishing your app.

## Load an ad

Once the AdView is in place, the next step is to load an ad. That's done with the `loadAd()` method in the AdView class. It takes an AdRequest parameter, which holds runtime information (such as targeting info) about a single ad request.

Here's an example that shows how to load an ad in the `onCreate()` method of an Activity:

MainActivity (excerpt)

```
package ...

import ...
import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.AdView;

public class MainActivity extends AppCompatActivity {
    private AdView mAdView;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MobileAds.initialize(this,
            "ca-app-pub-3940256099942544~3347511713");

        mAdView = findViewById(R.id.adView);
        AdRequest adRequest = new AdRequest.Builder().build();
        mAdView.loadAd(adRequest);
    }
}
```

That's it! Your app is now ready to display banner ads.

**Interstitial:** Interstitials are full-screen ads that cover the interface of an app until closed by the user. They're best used at natural pauses in the flow of an app's execution, such as between levels of a game or just after a task is completed.

## IMPLEMENT AN INTERSTITIAL

### Create an interstitial ad object

Interstitial ads are requested and shown by InterstitialAd objects. The first step is instantiating InterstitialAd and setting its ad unit ID. This is done in the `onCreate()` method of an Activity:

```

package ...

import com.google.android.gms.ads.InterstitialAd;

public class MainActivity extends Activity {

    private InterstitialAd mInterstitialAd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MobileAds.initialize(this,
            "ca-app-pub-3940256099942544~3347511713");

        mInterstitialAd = new InterstitialAd(this);
        mInterstitialAd.setAdUnitId("ca-app-pub-
3940256099942544/1033173712");
    }
}

```

### Always test with test ads

When building and testing your apps, make sure you use test ads rather than live, production ads. Failure to do so can lead to suspension of your account.

The easiest way to load test ads is to use our dedicated test ad unit ID for Android interstitials:

ca-app-pub-3940256099942544/1033173712

It's been specially configured to return test ads for every request, and you're free to use it in your own apps while coding, testing, and debugging. Just make sure you replace it with your own ad unit ID before publishing your app.

### Load an ad

To load an interstitial ad, call the `InterstitialAd` object's `loadAd()` method. This method accepts an `AdRequest` object as its single parameter:

```

package ...

import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.InterstitialAd;

```

```

public class MainActivity extends Activity {

    private InterstitialAd mInterstitialAd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MobileAds.initialize(this,
            "ca-app-pub-3940256099942544~3347511713");

        mInterstitialAd = new InterstitialAd(this);
        mInterstitialAd.setAdUnitId("ca-app-pub-
3940256099942544/1033173712");
        mInterstitialAd.loadAd(new AdRequest.Builder().build());
    }
}

```

### Show the ad

Interstitial ads should be displayed during natural pauses in the flow of an app. Between levels of a game is a good example, or after the user completes a task. To show an interstitial, use the `isLoaded()` method to verify that it's done loading, then call `show()`. The interstitial ad from the previous code example could be shown in a button's `OnClickListener` like this:

```

mMyButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mInterstitialAd.isLoaded()) {
            mInterstitialAd.show();
        } else {
            Log.d("TAG", "The interstitial wasn't loaded yet.");
        }
    }
});

```

### Create a Google AdMob account

I am not going to guide you through the whole Admob Signup process, but it's reasonably straightforward process. You need to put in bank details, company details etc. To sign up for AdMob click this link:

[https://support.google.com/admob/answer/7356219?visit\\_id=636806323355398905-4236639728&rd=1](https://support.google.com/admob/answer/7356219?visit_id=636806323355398905-4236639728&rd=1)

## **Set up an app in AdMob**

The first step to making money in your app with AdMob is setting it up. Published and unpublished apps can be added. If you set up an unpublished app, you'll need to come back to AdMob and link the app to its app store entry later when it's published.

### **Complete the following steps to set up an app.**

#### **Set up a published app**

1. Sign in to your AdMob account at <https://apps.admob.com>.
2. Click Apps in the sidebar.
3. Click Add app. A dialog box appears.
4. Click Yes.
5. Enter the name of the app and click Search.
6. Click Add beside the app you want to add.

#### **Set up an unpublished app**

1. Sign in to your AdMob account at <https://apps.admob.com>.
2. Click Apps in the sidebar.
3. Click Add app. A dialog box appears.
4. Click No.
5. Enter the app name.
6. Select a platform.
7. Click Add.
8. Click Add ad unit to create an ad unit for this app or click I'll do it later to exit the page.

When it's published, you can return to AdMob and link your app to its app store entry.

## **Your ad units**

Ad units are containers you place in your apps to show ads to users. Ad units send ad requests to AdMob, then display the ads they receive to fill the request. When you create an ad unit, you assign it an ad format and ad type(s).

### **Create a banner ad unit**

The following steps will help you create a new ad unit in your AdMob account, then implement it in your app's code. You must complete all of these steps to start showing ads in this ad unit.

#### **Instructions**

1. Sign in to your AdMob account at <https://apps.admob.com>.
2. Click Apps in the sidebar.

3. Select the name of the app you're creating this ad unit for. If you don't see it in the list of recent apps, you can click Add app to add a new app or click View all apps to search a list of all of the apps you've added to AdMob.
4. Click Ad units in the sidebar.
5. Click Get started. If you've already created ad units for this app, click Add ad unit.
6. Click Select for the Banner ad format.
7. Enter a name for this ad unit for example "flappy\_bird\_banner".
8. Click Create ad unit.
9. Follow the earlier instructions to implement this ad unit in your app code to start showing ads. You will need your app ID and ad unit ID during implementation.

Creating an interstitial ad unit is same as creating banner ad unit.

That's it! Your app is now ready to display banner and Interstitial ads.

Reference:

1. <https://developers.google.com/admob/android/quick-start>
2. Sign up for AdMob: [https://support.google.com/admob/answer/7356219?visit\\_id=636806323355398905-4236639728&rd=1](https://support.google.com/admob/answer/7356219?visit_id=636806323355398905-4236639728&rd=1)
3. Set up an app in AdMob: <https://support.google.com/admob/answer/2773509>
4. Banner: <https://developers.google.com/admob/android/banner>
5. Interstitial: <https://developers.google.com/admob/android/interstitial>

Additional Resources:

1. YouTube: [https://www.youtube.com/watch?v=w7mulkMYE\\_A](https://www.youtube.com/watch?v=w7mulkMYE_A)