

## Synopsis

### Before you begin

To prepare your app, complete the steps in the following sections.

### App prerequisites

- Make sure that your app's build file uses the following values:
  - Minimum SDK version of 21 or higher
  - Compile SDK version of 33 or higher

### Set up your app in your AdMob account

Register your app as an AdMob app by completing the following steps:

1. [Sign in to](#) or [sign up for](#) an AdMob account.
2. [Register your app with AdMob](#). This step creates an AdMob app with a unique [AdMob App ID](#) that is needed later in this guide.

### Configure your app

1. In your Gradle settings file, include the [Google's Maven repository](#) and [Maven central repository](#):

[KotlinGroovy](#)

```
pluginManagement{
```

```
    repositories {
```

```
        google()
```

```
        mavenCentral()
```

```
gradlePluginPortal()

}

}

dependencyResolutionManagement{

    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)

    repositories {

        google()

        mavenCentral()

    }

}

rootProject.name = "My Application"

include(":app")
```

2. Add the dependencies for the Google Mobile Ads SDK to your app-level build file:

[KotlinGroovy](#)

```
dependencies{  
  
    implementation("com.google.android.gms:play-services-ads:23.3.0")  
  
}
```

3. Add your AdMob app ID, as [identified in the AdMob web interface](#), to your app's AndroidManifest.xml file. To do so, add a <meta-data> tag with android:name="com.google.android.gms.ads.APPLICATION\_ID". You can find your **app ID** in the AdMob web interface. For android:value, insert your own AdMob app ID, surrounded by quotation marks.

```
<manifest>
```

```
<application>
```

```
<!-- Sample AdMob app ID: ca-app-pub-3940256099942544~3347511713 -->
```

```
<meta-data
```

```
    android:name="com.google.android.gms.ads.APPLICATION_ID"
```

```
    android:value="ca-app-pub-xxxxxxxxxxxxxxxx~yyyyyyyyyy"/>
```

```
</application>
```

```
</manifest>
```

In a real app, replace the sample app ID with your **actual AdMob app ID**. You can use the sample ID if you're just experimenting with the SDK in a Hello World app.

Also, note that failure to add the <meta-data> tag exactly as shown results in a crash with the message:

Missing application ID.

*(Optional) Declare **AD\_ID** permission for previous versions to work with Android 13.*

### **Initialize the Google Mobile Ads SDK**

Before loading ads, initialize the Google Mobile Ads SDK by calling [MobileAds.initialize\(\)](#).

This method initializes the SDK and calls a completion listener once both the Google Mobile Ads SDK and adapter initializations have completed, or after a 30-second timeout. This needs to be done only once, ideally at app launch.

**Note:** If you're using AdMob Mediation, wait until the completion handler is called before loading ads. This ensures that all mediation adapters are initialized.

**Ads may be preloaded by the Google Mobile Ads SDK or mediation partner SDKs upon initialization.** If you need to obtain consent from users in the European Economic Area (EEA), set any request-specific flags, such as [setTagForChildDirectedTreatment\(\)](#) or [setTagForUnderAgeOfConsent\(\)](#), or otherwise take action before loading ads, ensure you do so before initializing the Google Mobile Ads SDK.

Here's an example of how to call the `initialize()` method on a background thread within an Activity:

[JavaKotlin](#)

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

```
import com.google.android.gms.ads.initialization.InitializationStatus;
```

```
import com.google.android.gms.ads.initialization.OnInitializationCompleteListener;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
newThread(  
  
    ()->{  
  
        // Initialize the Google Mobile Ads SDK on a background thread.  
  
        MobileAds.initialize(this,initializationStatus->{});  
  
    })  
  
    .start();  
  
}  
  
}
```

### Select an ad format

The Google 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 ad units display rectangular ads that occupy a portion of an app's layout. They can refresh automatically after a set period of time. This means users view a new ad at regular intervals, even if they stay on the same screen in your app. They're also the simplest ad format to implement.

[Implement banner ads](#)

#### Interstitial

Interstitial ad units show full-page ads in your app. Place them at natural breaks and transitions in your app's interface, such as after level completion in a gaming app.

[Implement interstitial ads](#)

## **Native**

Native ads are ads where you can customize the way assets such as headlines and calls to action are presented in your apps. By styling the ad yourself, you can create a natural, unobtrusive ad presentations that can add to a rich user experience.

[Implement native ads](#)

## **Rewarded**

Rewarded ad units enable users to play games, take surveys, or watch videos to earn in-app rewards, such as coins, extra lives, or points. You can set different rewards for different ad units, and specify the reward values and items the user received.

[Implement rewarded ads](#)

## **Rewarded interstitial**

Rewarded interstitial is a new type of incentivized ad format that lets you offer rewards, such as coins or extra lives, for ads that appear automatically during natural app transitions.

Unlike rewarded ads, users aren't required to opt in to view a rewarded interstitial.

Instead of the opt-in prompt in rewarded ads, rewarded interstitials require an intro screen that announces the reward and gives users a chance to opt out if they want to do so.

[Implement rewarded interstitial ads](#)

## **App open**

App open is an ad format that appears when users open or switch back to your app. The ad overlays the loading screen.

[Implement app open ads](#)

#### **Additional resources**

The [Google Mobile Ads repository](#) on GitHub demonstrates how to use the different ad formats that this API offers.