**Connect Flutter app to Firebase**

Before using Flutter Firebase on Android, you must first connect to your Firebase project with your Android application.

**Generating a Firebase project configuration file**

On the Firebase Console: (https://console.firebase.google.com/project/\_/overview),add a new Android app or select an existing Android app for your Firebase project.`

The "Android package name" must match your local project's package name. The current package name can be found in ‘android/app/src/main/AndroidManifest.xml’ usually package name: ‘com.yourcompany.yourproject’.

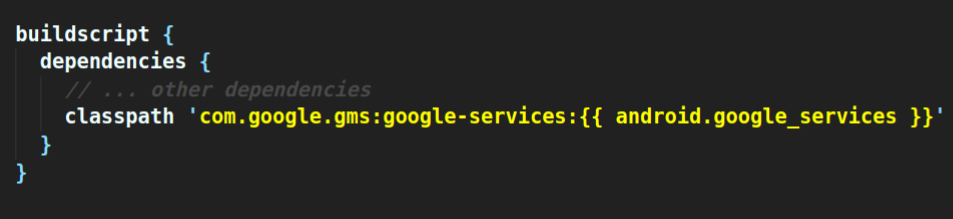
*When creating a new Android app "debug signing certificate SHA-1" is optional, however, it is required for Dynamic Links & Phone Authentication & Google Sign in. To generate a certificate run (cd android && ./gradlew signingReport) and copy the SHA1 from the ‘debug’ key. This generates two variant keys. You can copy the 'SHA1' that belongs to the ‘debugAndroidTest’ variant key option.*

Once your Android app has been registered, download the configuration file from the Firebase Console: ‘google-services.json’. Add this file into the ‘android/app’ directory within your Flutter project.

**Installing your Firebase configuration file**

First, add the 'google-services' plugin as a dependency inside of the ‘android/build.gradle’ file:

1. ‘android/build.gradle’



Lastly, execute the plugin by adding the following underneath the line ‘apply plugin: 'com.android.application’, within

the ‘/android/app/build.gradle’ file:

1. "android/app/build.gradle"

apply plugin*:* 'com.google.gms.google-services'

**Building for Android**

Due to the large number of classes in some of the Firebase SDKs , it may bump you over the 64k method limit on the Android

build system and you may get an error stating ‘Error while merging dex archives: The number of method references in a .dex file cannot exceed 64K’.

If you do get this error, we suggest enabling Multidex for Android.

**Enabling Multidex**

If your app only targets Android 21 or higher ‘minSdkVersion’ then multidex is already enabled by default and you do not need the multidex support library.

However, if your ‘minSdkVersion’ is set to 20 or lower, then you must use the multidex support library and make the following modifications to your app project:

Open the ‘/android/app/build.gradle’ file. Under `dependencies` add the multidex module, and enable it

within the ‘defaultConfig’:



Visit the [official Android documentation](https://developer.android.com/studio/build/multidex#mdex-gradle) to learn more.