

9antra.tn

# GOOGLE'S FIREBASE & FLUTTER.

Anis BEJAoui

## Part 1

# FIREBASE CONSOLE

### What do you need?

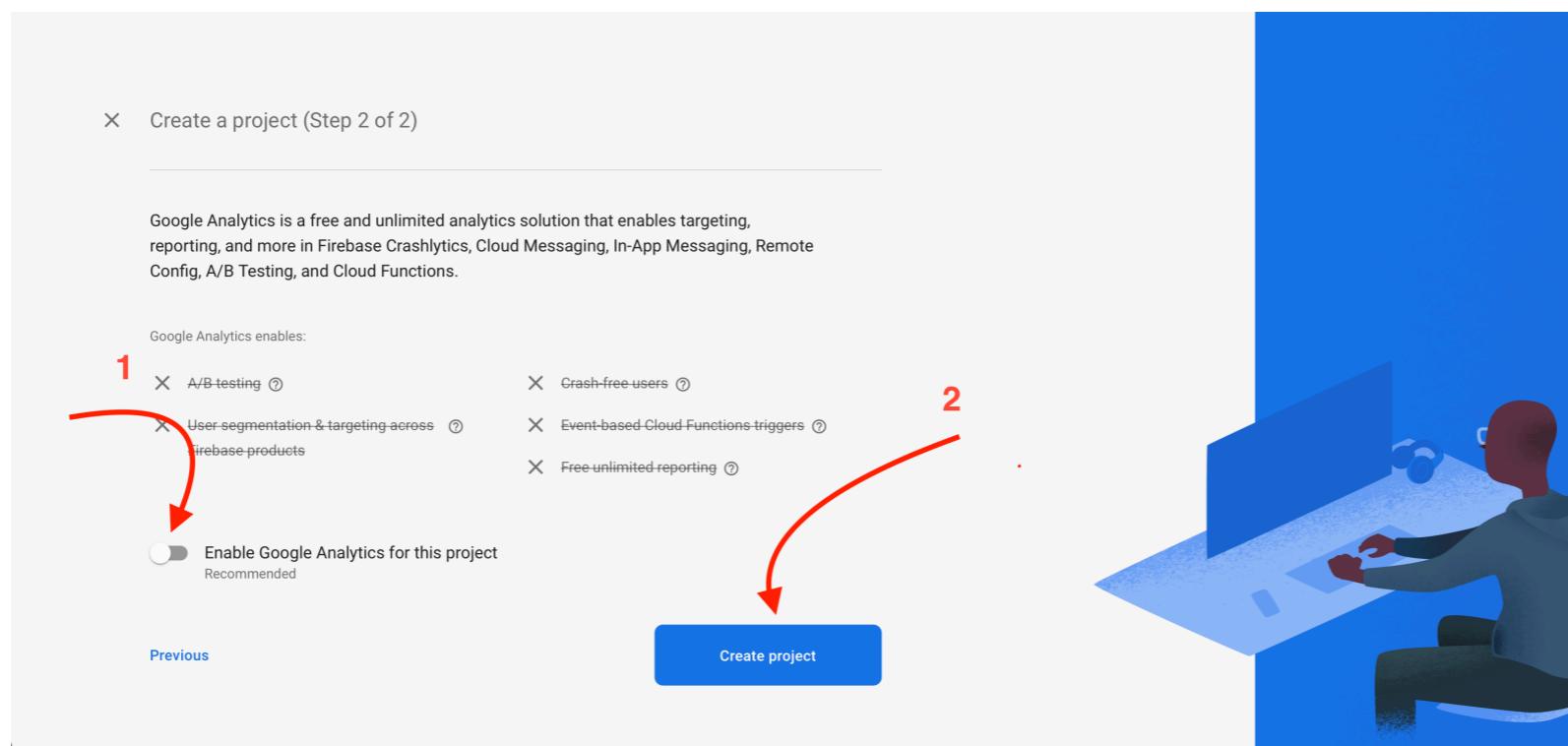
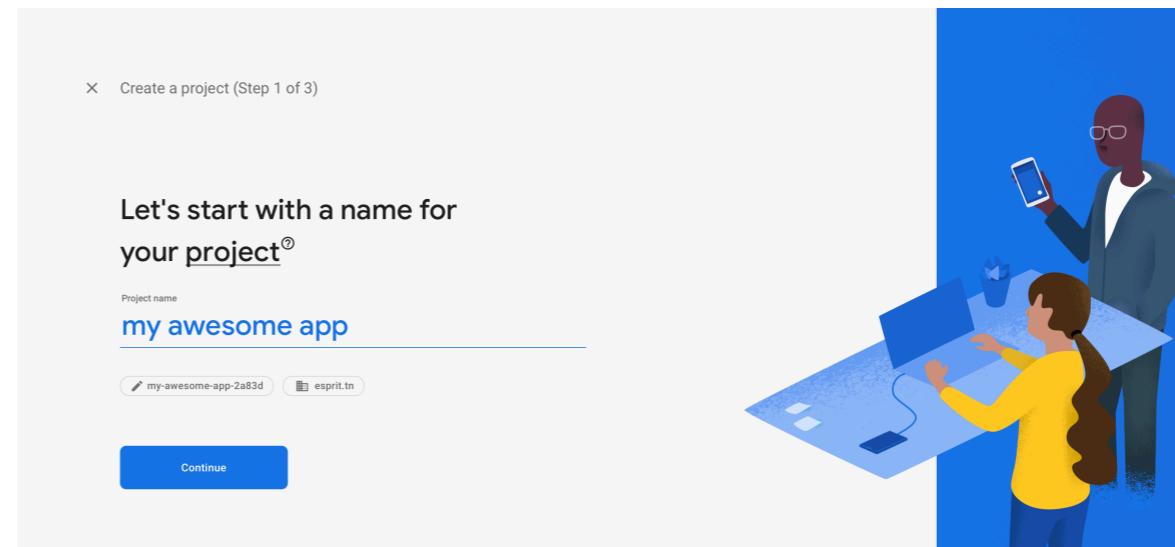
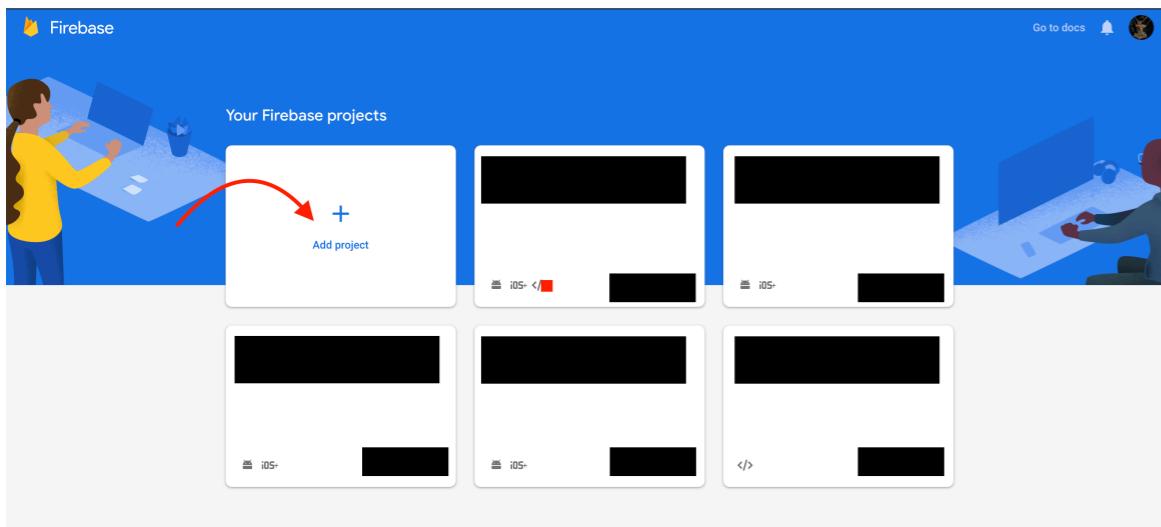
Gmail Account

Firebase account : <https://firebase.google.com>

Firebase project

## Step 1

- Go to firebase dashboard, login with your Gmail account, create a new project by clicking on “Add project” follow firebase’s wizard and complete the steps



## Step 2

- You will find your project in a card titled <your-project-name>, a click will redirect you to your **Firebase project's dashboard**

The image consists of two screenshots of the Firebase console. The top screenshot shows the 'Recent projects' section, where a project named 'my awesome app' is highlighted with a red box. An arrow points from the text 'You will find your project in a card titled <your-project-name>' to this card. The bottom screenshot shows the 'Project Overview' page for 'my awesome app'. A green box highlights the project name 'my awesome app'. A red box highlights the 'Add an app to get started' button, which is located below the 'Get started by adding Firebase to your app' section. The left sidebar is highlighted with an orange box.

### Firebase project name

- This is your **Firebase project's name and NOT your flutter app but it is recommended that you have similar naming.**

### Platforms

- Here where you define your project's platforms, when working with flutter, we set up iOS and Android separately.

### Firebase project name

- **Here you will find the services, like storage, database, auth, Firestore...**

## **Part 2.a**

# **PREPARING OUR FLUTTER PROJECT**

## **TO CONNECT TO FIREBASE**

### **What do you need?**

A Flutter project

MS VS Code or Android Studio

## Preparing our Flutter app to be linked with Firebase

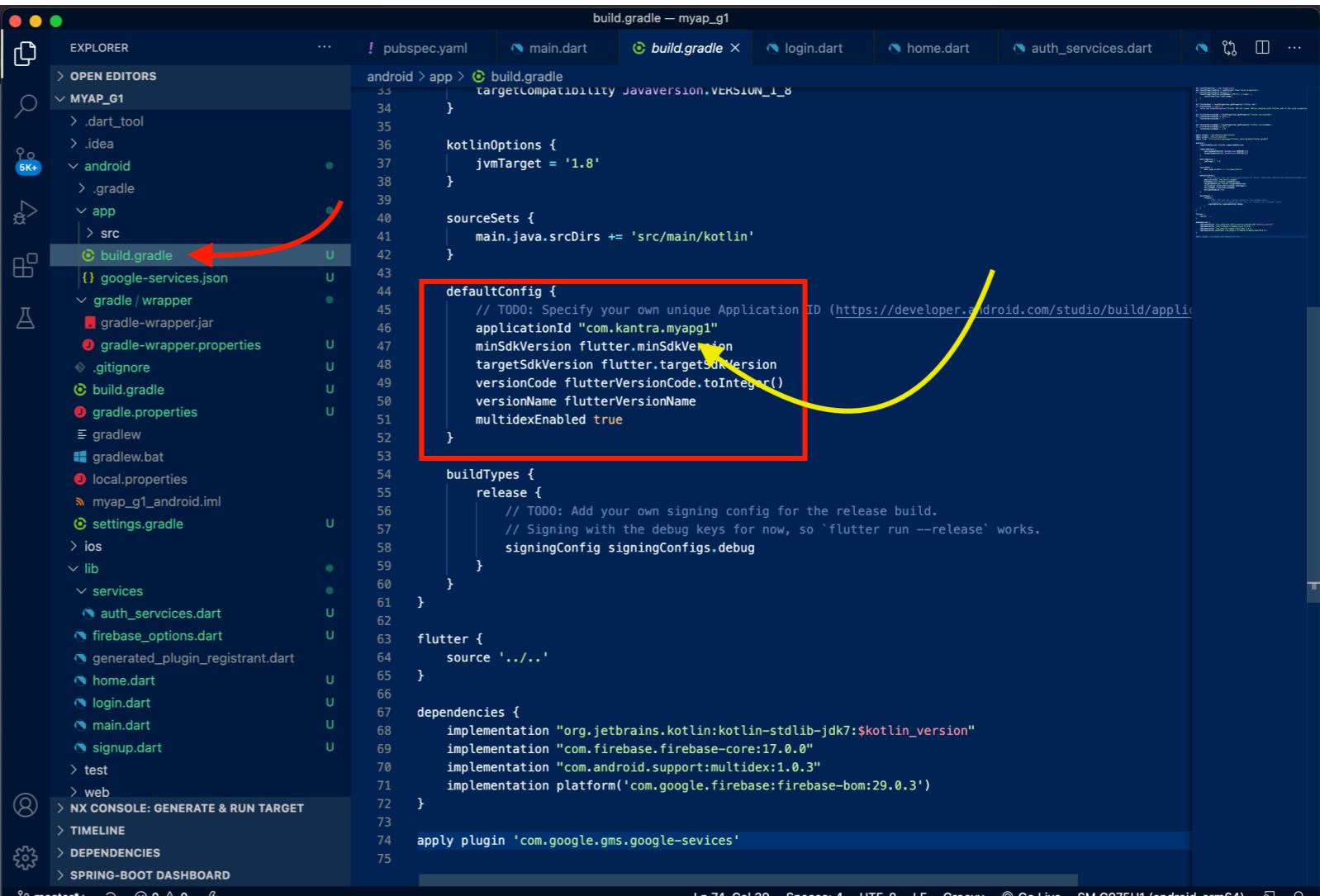
- Each app has a unique id that identifies it when going live on stores, this id generally goes under the following format:

**domain.company.name** for example: `com.facebook.facebook`, `com.google.youtube`, `com.samsung.gallery`. if you don't have a company you can simply put your name like so: `com.anis.myapp`.

This ID is known as **Apple Bundle ID** in iOS and **Android package ID** in Android, before doing anything it is best that you decided your package/bundle IDs and change them across all respective files, for Apple the process would require a Macintosh device & Xcode installed, for Android a Windows can do the trick.

Let's start by changing those IDs under Android.

1. myapp > Android > app > build.gradle



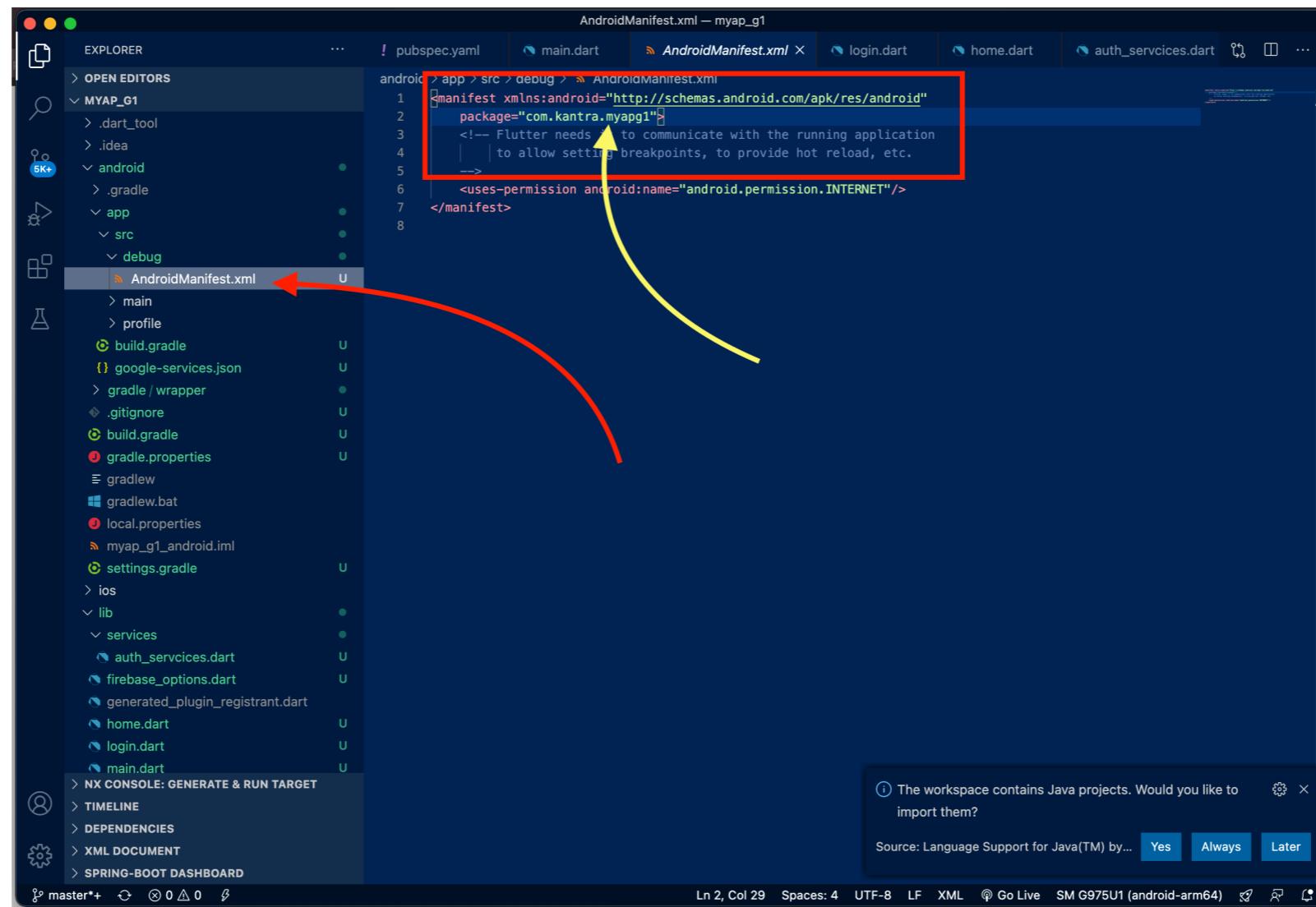
```
build.gradle — myap_g1
pubspec.yaml main.dart build.gradle login.dart home.dart auth_services.dart

EXPLORER
MYAP_G1
  .dart_tool
  .idea
  android
    .gradle
    app
      src
        build.gradle
        google-services.json
        gradle/wrapper
          gradle-wrapper.jar
          gradle-wrapper.properties
        .gitignore
        build.gradle
        gradle.properties
        gradlew
        gradlew.bat
        local.properties
        myap_g1_android.iml
        settings.gradle
      ios
      lib
      services
        auth_services.dart
        firebase_options.dart
        generated_plugin_registrant.dart
        home.dart
        login.dart
        main.dart
        signup.dart
      test
      web
      NX CONSOLE: GENERATE & RUN TARGET
      TIMELINE
      DEPENDENCIES
      SPRING-BOOT DASHBOARD

build.gradle
33   targetCompatibility javaVersion.VERSION_1_8
34 }
35
36 kotlinOptions {
37   jvmTarget = '1.8'
38 }
39
40 sourceSets {
41   main.java.srcDirs += 'src/main/kotlin'
42 }
43
44 defaultConfig {
45   // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id)
46   applicationId "com.kantra.myapg1"
47   minSdkVersion flutter.minSdkVersion
48   targetSdkVersion flutter.targetSdkVersion
49   versionCode flutterVersionCode.toInt()
50   versionName flutterVersionName
51   multiDexEnabled true
52 }
53
54 buildTypes {
55   release {
56     // TODO: Add your own signing config for the release build.
57     // Signing with the debug keys for now, so 'flutter run --release' works.
58     signingConfig signingConfigs.debug
59   }
60 }
61
62 flutter {
63   source '...'
64 }
65
66 dependencies {
67   implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
68   implementation "com.firebaseio.firebaseio-core:17.0.0"
69   implementation "com.android.support:multidex:1.0.3"
70   implementation platform('com.google.firebaseio:firebase-bom:29.0.3')
71 }
72
73 apply plugin 'com.google.gms.google-services'
```

## Preparing our Flutter app to be linked with Firebase

2. myapp > Android > app > src > debug > *AndroidManifest.xml*



## Preparing our Flutter app to be linked with Firebase

3. myapp > Android > app > src > main > Kotlin / .. / .. / .. > MainActivity.kt

The screenshot shows a dark-themed IDE interface with the following details:

- EXPLORER View:** Shows the project structure under "MYAP\_G1". Key items include ".dart\_tool", ".idea", "android", ".gradle", "app", "src", "debug", "main", "kotlin", "com", "example", "myap\_g1", "MainActivity.kt", "res", "AndroidManifest.xml", "build.gradle", "google-services.json", ".gitignore", "build.gradle", "gradle.properties", "gradlew", "gradlew.bat", "local.properties", "myap\_g1\_android.iml", "settings.gradle", "ios", "lib", "services", "auth\_servcices.dart", and "firebase\_options.dart".  
A red arrow points from the "MainActivity.kt" entry in the Explorer view to the same file in the main editor area.
- Main Editor Area:** The file "MainActivity.kt" is open. The code is as follows:

```
package com.kantra.myapg1

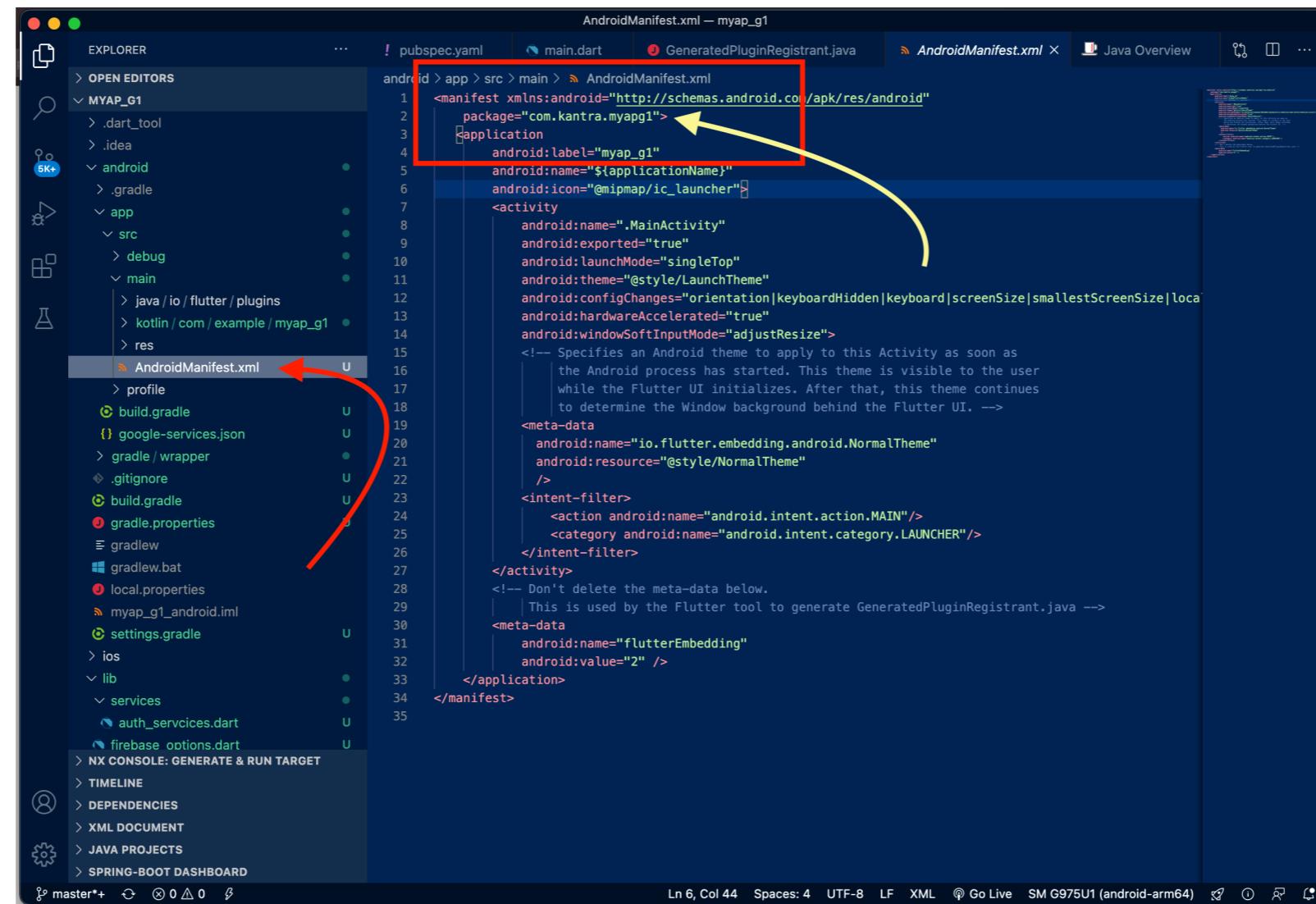
import io.flutter.embedding.android.FlutterActivity

class MainActivity: FlutterActivity() {
```

A yellow arrow points from the "MainActivity" declaration in the code back to the "MainActivity.kt" entry in the Explorer view.
- Bottom Status Bar:** Displays "master\*+", "0 △ 0", "Kotlin: Indexing", "Ln 1, Col 24", "Spaces: 4", "UTF-8", "LF", "Kotlin", "Go Live", "SM G975U1 (android-arm64)", and various icons.

# Preparing our Flutter app to be linked with Firebase

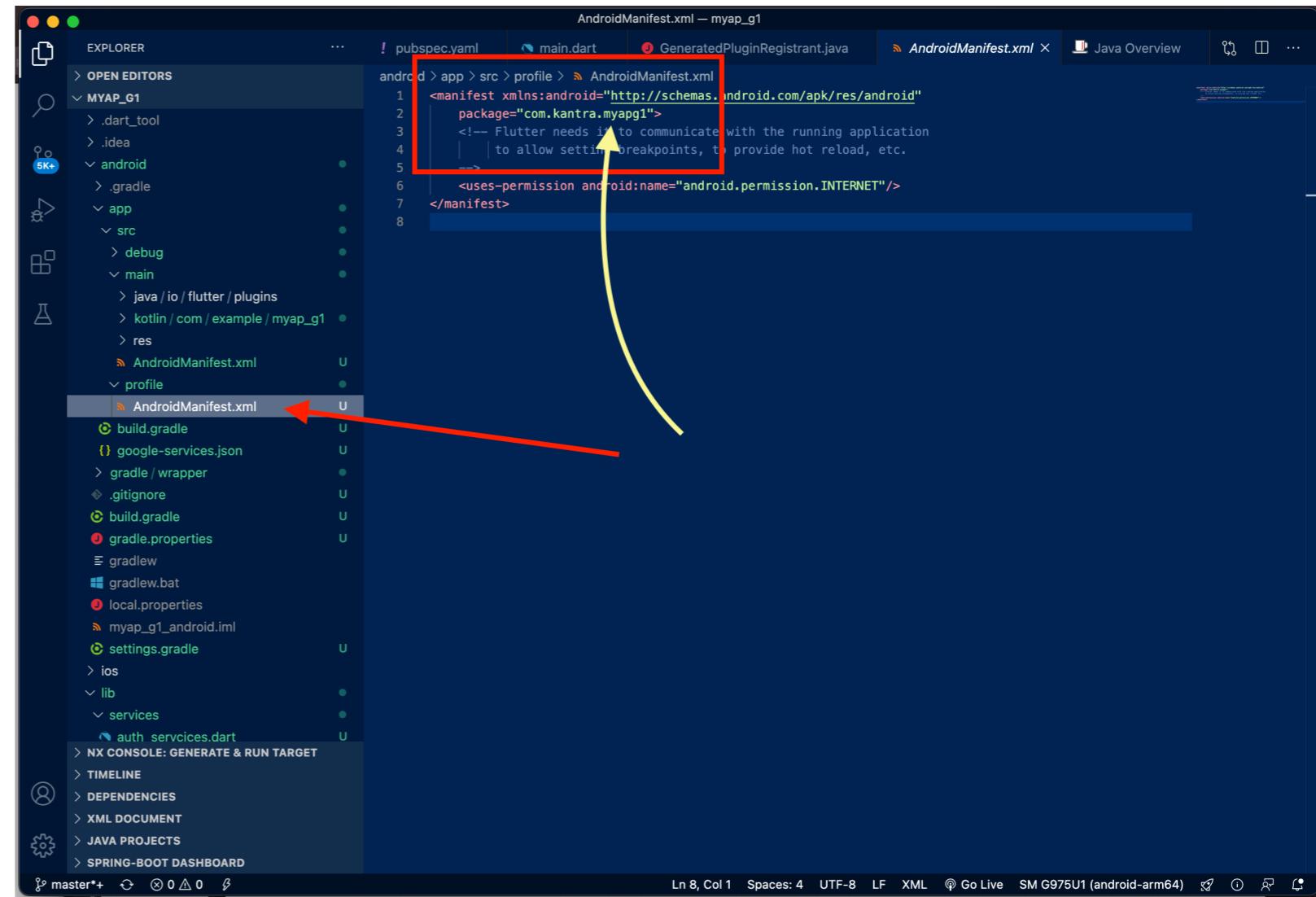
4. myapp > Android > app > src > main > AndroidManifest.xml



```
AndroidManifest.xml — myap_g1
pubspec.yaml main.dart GeneratedPluginRegistrant.java AndroidManifest.xml Java Overview
android > app > src > main > AndroidManifest.xml
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="com.kantra.myapg1">
3     application
4       android:label="myap_g1"
5       android:name="${applicationName}"
6       android:icon="@ipmap/ic_launcher">
7         activity
8           android:name=".MainActivity"
9             android:exported="true"
10            android:launchMode="singleTop"
11            android:theme="@style/LaunchTheme"
12            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale"
13            android:hardwareAccelerated="true"
14            android:windowSoftInputMode="adjustResize">
15              <!-- Specifies an Android theme to apply to this Activity as soon as
16                  the Android process has started. This theme is visible to the user
17                  while the Flutter UI initializes. After that, this theme continues
18                  to determine the Window background behind the Flutter UI. -->
19              <meta-data
20                android:name="io.flutter.embedding.android.NormalTheme"
21                android:resource="@style/NormalTheme"
22              />
23              <intent-filter>
24                <action android:name="android.intent.action.MAIN"/>
25                <category android:name="android.intent.category.LAUNCHER"/>
26              </intent-filter>
27            </activity>
28            <!-- Don't delete the meta-data below.
29                | This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
30            <meta-data
31              android:name="flutterEmbedding"
32              android:value="2" />
33          </application>
34      </manifest>
```

## Preparing our Flutter app to be linked with Firebase

5. myapp > Android > app > src > profile > AndroidManifest.xml



## Preparing our Flutter app to be linked with Firebase

Now your Android Package ID should be set, let's move on.

This step is related to configuring Firebase on Android OS, this is a platform specific configuration, we will add some dependencies and edit some parameters.

### 1. Adding the following dependencies

1. First of all let's go back to **myapp > Android > app > build.gradle**

2. Now add the following packages under dependencies block

```
implementation "com.google.firebaseio:firebase-core:17.0.0"
```

```
implementation "androidx.multidex:multidex:2.0.1" [updated]
```

```
implementation platform('com.google.firebaseio:firebase-bom:29.2.1') [updated]
```

3. Now out of dependencies block and add the following plugin in line 24 (along with the other plugins)

```
apply plugin: 'com.google.gms.google-services' [updated]
```

4. Lastly let's switch one more variable to true:

in the code block defaultConfig and at the end add the following line:

```
multiDexEnabled true [updated]
```

```
23  
24     apply plugin: 'com.google.gms.google-services'  
25     apply plugin: 'com.android.application'  
26     apply plugin: 'kotlin-android'  
27     apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"  
28
```

```
68     dependencies {  
69         implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
70         implementation "com.google.firebaseio:firebase-core:17.0.0"  
71         implementation "androidx.multidex:multidex:2.0.1"  
72         implementation platform('com.google.firebaseio:firebase-bom:29.2.1')  
73     }
```



Now go to **myapp > Android > build.gradle [updated]**

```
android > ⚡ build.gradle
1  buildscript {
2      ext.kotlin_version = '1.6.10'
3      repositories {
4          google()
5          mavenCentral()
6      }
7
8      dependencies {
9          classpath 'com.google.gms:google-services:4.3.10' ←
10         classpath 'com.android.tools.build:gradle:4.1.0' ←
11         classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
12     }
13 }
```

# BACK TO FIREBASE

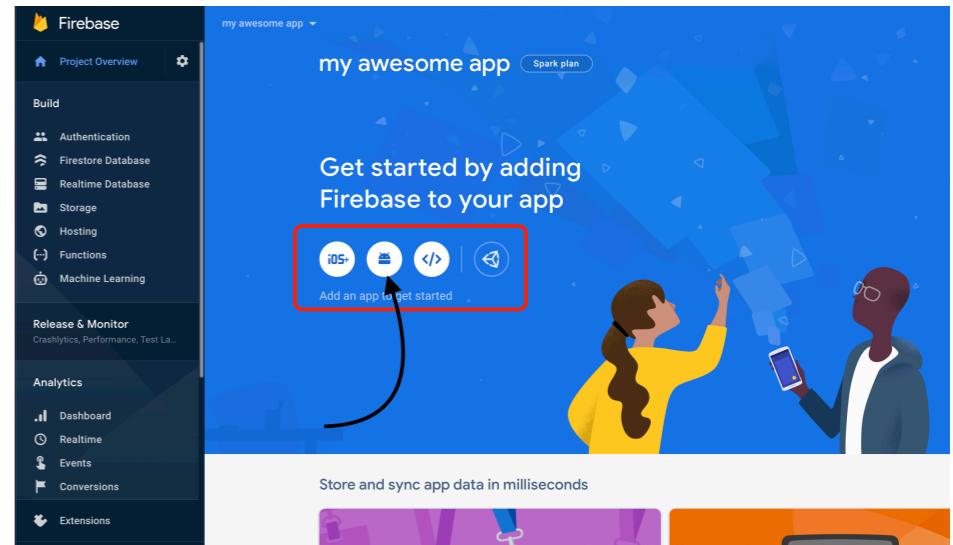
**Now let's go back to firebase and finish configuring the project**

**a. Start by choosing Android from the platform list**

**Now let's configure the Android platform:**

**1. Start by filling out the form (skip SHA fingerprint)**

**2. Click register app and got to step 2**



× Add Firebase to your Android app

1 Register app

Android package name ?  
 com.example.myapp

App nickname (optional) ?  
 app name

Debug signing certificate SHA-1 (optional) ?  
 skip this  
Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

2 Download config file

3 Add Firebase SDK skip this

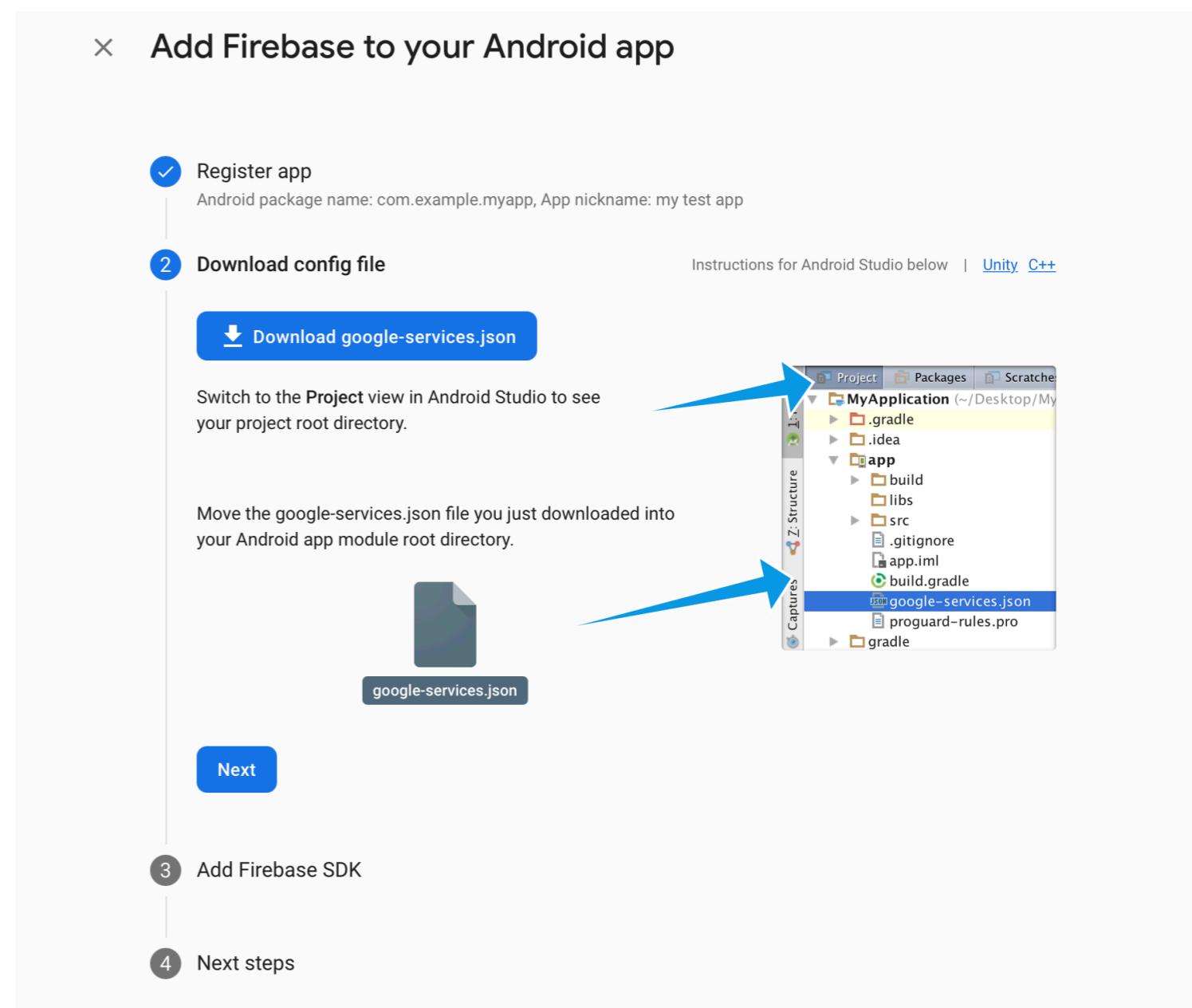
4 Next steps

## Download the .json file

**Make sure the name is correct and there is no numbers at the end it should look something like 'google-services.json', like the diagram shown in firebase wizard IT IS MANDATORY THAT YOU PLACE YOUR JSON FILE LIKE SO:**

**Myapp > Android > app > google-services.json**

**NOW SKIP STEP 3 AND CONFIRM IN STEP 4**



# CONFIGURING FLUTTER & DART

## **NOW FIREBASE CONSOLE & ANDROID PLATFORM ARE READY**

**The last part is configuring Flutter & Dart**

**Let's start by :**

**1. Opening the windows CLI (command line interface, CMD, terminal)**

**2. Navigate to 'My documents' and type the following command:**

**npm install -g firebase-tools**

**3. After the installation is complete Navigate to your app's directory and type the following:**

**dart pub global activate flutterfire\_cli**

**4. After the last command finish processing type again:**

**flutterfire configure**

**Follow the wizard hit enter and you should be set!**

**Let's write some dart code.**

**Let's start by importing Firebase dependencies into pubspec.yaml**

**We will need:**

**firebase\_auth: ^3.3.4**

**firebase\_core: ^1.10.6**

**cloud\_firestore: ^3.1.5**

**Now in your main.dart**

**Add one more import:**

**Import 'firebase\_options.dart'**

**And finally let's edit our main function:**

```
void main() async {
```

```
    WidgetsFlutterBinding.ensureInitialized();
```

```
    await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
```

```
    runApp(MyApp());
```

```
}
```

**AND THAT IS IT**