



# Documentation

**Name:**Active Ecommerce Delivery App

**Author :** Active IT zone

**Software Framework :** Flutter

**Addon For:** Active eCommerce CMS

**Provided by :** codecanyon



## **Documentation**

1. What are the prerequisites?
2. How to run Flutter Application in Android Studio?
3. How to configure the App according to your setup?
4. How to change the package name?
5. How to build the App for testing (build and apk) ?
6. How to generate play store uploadable files for release?
7. How to generate app store uploadable files?
8. How to Update for Android?
9. How to configure the default language for mobile apps?
10. How to configure google maps?

## 1. What are the prerequisites?

This Flutter app can be hosted into Google Play Store + Apple Appstore as your branded eCommerce CMS app. The app will communicate with your hosted eCommerce CMS web application through APIs. That means the prerequisite to publish the eCommerce Mobile application is to have the eCommerce CMS Web application in the latest version always.

- a. Flutter version must be : **Flutter 3.10.04 • channel stable**
- b. **Android Studio Flamingo | 2022.2.1 Patch 1**  
<https://developer.android.com/studio/archive>
- c. **Java version OpenJDK Runtime Environment (build 17.0.6+0-b2043.56-9586694)**
- d. Make sure your flutter and dart versions are correct. Follow the flutter documentation from <https://flutter.dev/docs/get-started/install> to install the given version of flutter in your pc/mac.

## 2. How to run Android Application in Android Studio?

- Install Android Studio from <https://developer.android.com/studio>
- Extract the source\_code.zip. You will find this inside the flutter.zip.
- Open the folder in your android studio.
- Even if you are building an app for ios, use android studio for the build.
- Then in your android studio terminal run:  
`flutter pub get` \*\* You need this to get all 3rd party packages from pub.dev

### 3. How to configure the App according to your setup?

#### 1. App Config:

This helps you connect your app to your server.

Open lib/app\_config.dart

You can change the `copyright_text`, `app_name`, `HTTPS`, `DOMAIN_PATH` variable.

Do not change the other variables.

If your site does not have https or your are using a local machine as server (localhost)

the make `HTTPS = false`;

Your `DOMAIN_PATH` is your site url without any protocol. (see screenshot below)

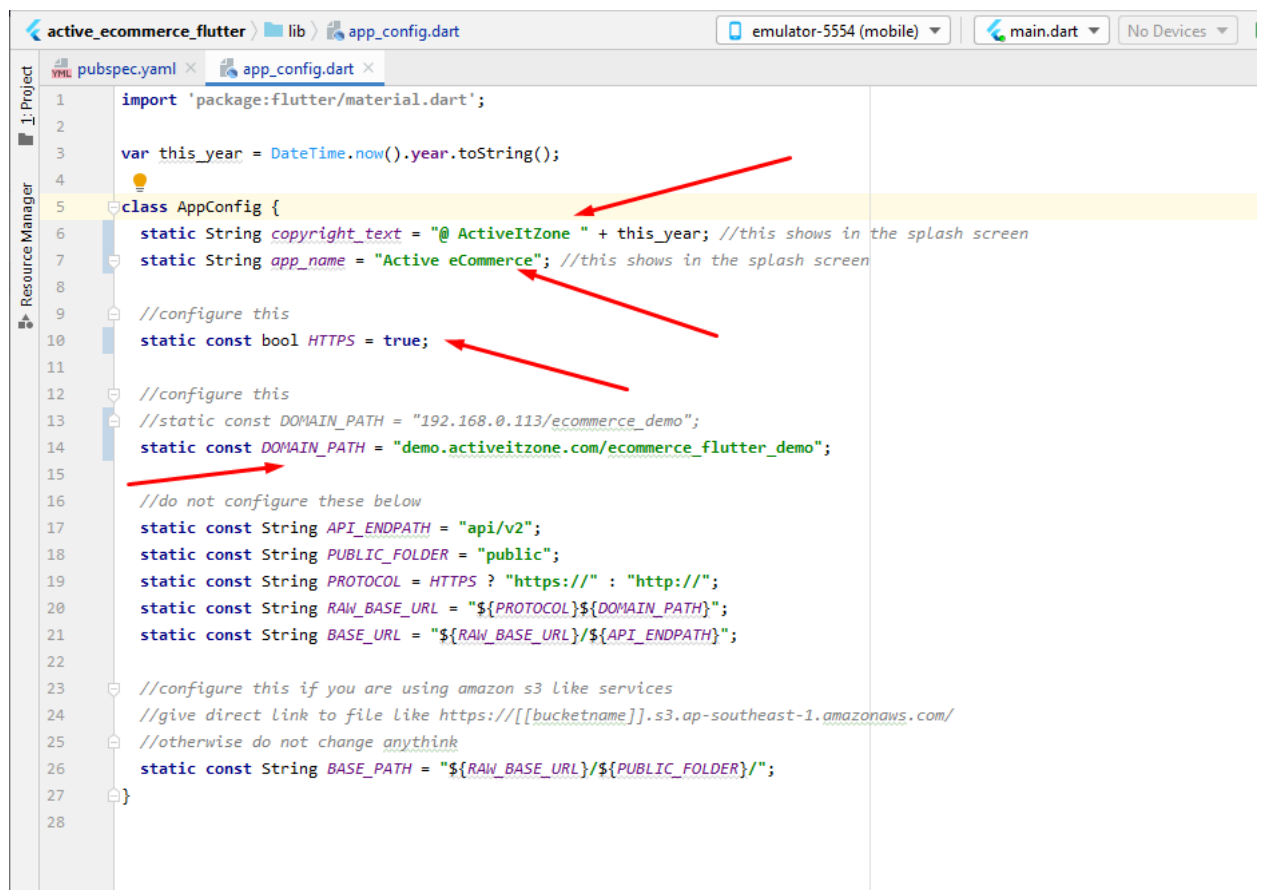
If you are using localhost , `DOMAIN_PATH` should be `"your_ip_address/your_project"`;

**\*\* "localhost/your\_project" will not not work \*\***

Normally you do not have to change the `BASE_PATH`. Keep it as given.

But if you are using s3 for image uploading your `BASE_PATH` should be :

`BASE_PATH = "https://[[bucketname]].s3. [[regeion]].amazonaws.com/"`;



```
1 import 'package:flutter/material.dart';
2
3 var this_year = DateTime.now().year.toString();
4
5 class AppConfig {
6   static String copyright_text = "@ ActiveItZone " + this_year; //this shows in the splash screen
7   static String app_name = "Active eCommerce"; //this shows in the splash screen
8
9   //configure this
10  static const bool HTTPS = true;
11
12  //configure this
13  static const DOMAIN_PATH = "192.168.0.113/eCommerce_demo";
14  static const DOMAIN_PATH = "demo.activeitzone.com/eCommerce_flutter_demo";
15
16  //do not configure these below
17  static const String API_ENDPOINT = "api/v2";
18  static const String PUBLIC_FOLDER = "public";
19  static const String PROTOCOL = HTTPS ? "https://" : "http://";
20  static const String RAW_BASE_URL = "${PROTOCOL}${DOMAIN_PATH}";
21  static const String BASE_URL = "${RAW_BASE_URL}/${API_ENDPOINT}";
22
23  //configure this if you are using amazon s3 like services
24  //give direct link to file like https://[[bucketname]].s3.ap-southeast-1.amazonaws.com/
25  //otherwise do not change anything
26  static const String BASE_PATH = "${RAW_BASE_URL}/${PUBLIC_FOLDER}/";
27 }
28
```

The screenshot shows the `app_config.dart` file in an IDE. Red arrows point to the following lines: line 6 (comment), line 7 (comment), line 10 (`HTTPS = true`), line 13 (`DOMAIN_PATH`), and line 14 (`DOMAIN_PATH`).

## 2. Theme Config:

This helps you change your app's colors according to your theme/branding

Open lib/my\_theme.dart

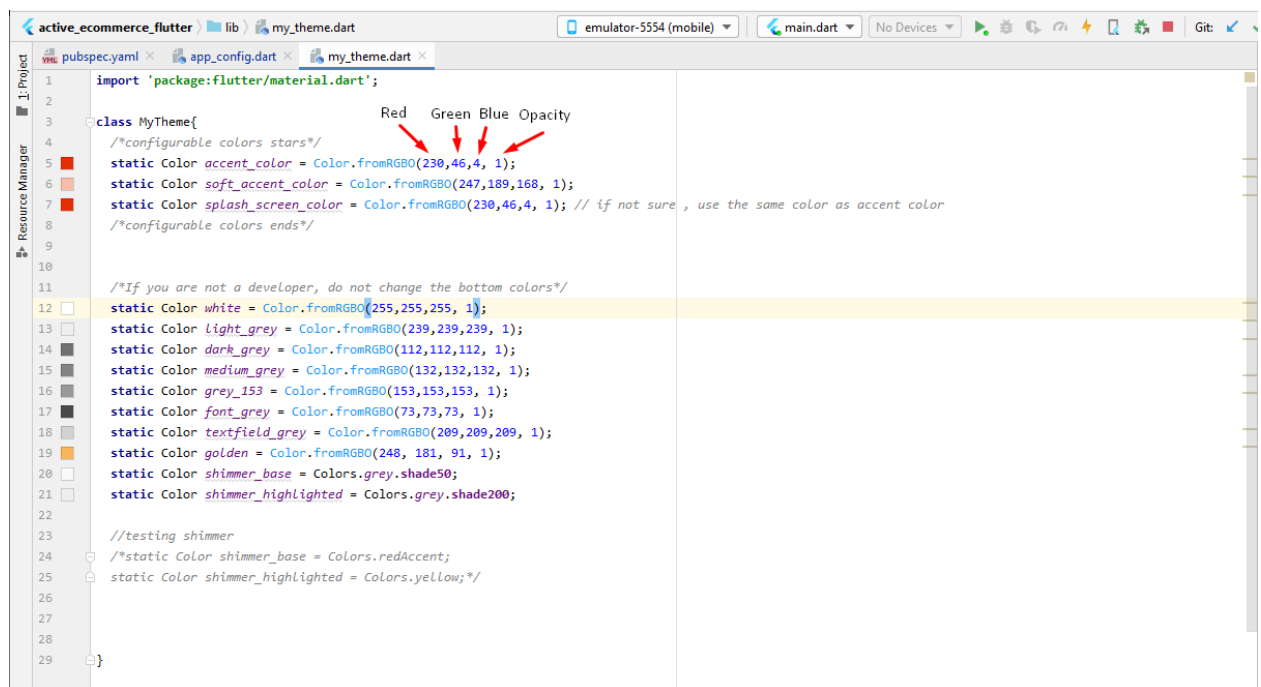
You can change the `accent_color`, `soft_accent_color`, `splash_screen_color` variable.

Flutter by default does not support hex color. Do not change the other variables.

Use <https://www.rapidtables.com/convert/color/hex-to-rgb.html> To get the RGB value if you do not already know your theme's RGB color.

You should keep the Opacity value 1 (Opacity can be 0, 0.1, 0.2, ..... ,0.9 ,1)

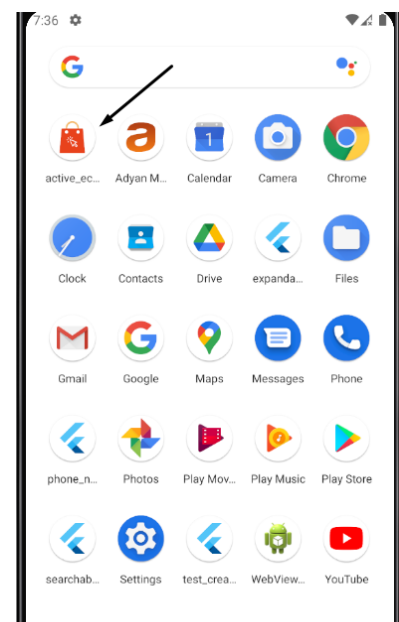
See the screenshot below.



## 3. Configure the launcher icon:

This helps you change your app's launcher icon.

Change the `app_logo.png` in `assets` folder with your own logo. Your file name should also be `app_logo.png` and it should be a 512x512 png image and the image format should be the same.



After replacing the file , **uninstall** your app from your emulator. Otherwise the logo will not be changed.

Then in your android studio terminal run:

```
flutter pub get
```

Then run :

```
flutter pub run flutter_launcher_icons:main
```

Then run your app (shift +10). The app will be installed again with your given launcher icon.

\*\*The logo used in various screens inside the app is **delivery\_app\_long.png** you can change it with your own logo or keep as it is. Remember the file name, dimension and image format should be the same

#### 4. How to change the package name ?

This is very important. Your app cannot have the same package name as other app. If it does, the playstore will not accept it as an unique application. So rename your app according to your business/brand name. Try to write an unique package name.

Naming convention : <https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html>

For example

Let's say your package is : **com.onatcipli.networkUpp**

And your app name is **"Network Upp"**

Then ,

Run this command inside your flutter project root.

Run the command in android studio terminal :

```
flutter pub run rename --bundleId com.onatcipli.networkUpp
```

```
flutter pub run rename --appname "Network Upp"
```

Try uninstalling the app from the emulator , then run the commands and then restart the app.  
If it does not work, first uninstall, then restart the app then run the commands.

**\*\*In case the above do not work:**

### **In Android**

for **package name** just change in build build.gradle only (android/app/build.gradle)

```
defaultConfig {  
    applicationId "bundleId com.onatcipli.networkUpp"  
    .....  
}
```

### **For iOS**

Change the bundle identifier from your Info.plist file inside your ios/Runner directory.

```
<key>CFBundleIdentifier</key>  
<string>bundleId com.onatcipli.networkUpp</string>
```

If you face issues consult a flutter developer.

## **5. How to Build the app for testing (build an apk) ?**

<https://flutter.dev/docs/deployment/android> see the doc for reference

In terminal run : **flutter build apk**

It will build an apk and show the folder. You can then install it in your phone to test, or share to multiple users for testing .

## **6. How to generate play store uploadable files for release?**

<https://flutter.dev/docs/deployment/android> see the doc for reference

Signing the app:

To publish on the Play Store, you need to give your app a digital signature. Use the following instructions to sign your app.

Go through the screenshots below carefully to understand how to generate key and use it for the released signed app:

**Note:**

- The `keytool` command might not be in your path—it's part of Java, which is installed as part of Android Studio. For the concrete path, run `flutter doctor -v` and locate the path printed after 'Java binary at:'. Then use that fully qualified path replacing `java` (at the end) with `keytool`. If your path includes space-separated names, such as `Program Files`, use platform-appropriate notation for the names. For example, on Mac/Linux use `Program\ Files`, and on Windows use `"Program Files"`.
- The `-storetype JKS` tag is only required for Java 9 or newer. As of the Java 9 release, the keystore type defaults to `PKS12`.



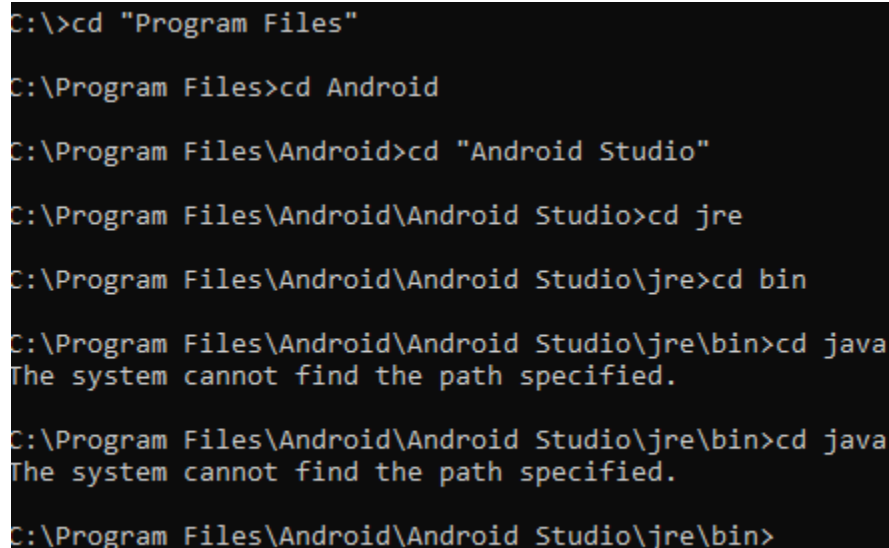
```
C:\flutter_projects\active_ecommerce_flutter>flutter doctor -v
[v] Flutter (Channel stable, 1.22.4, on Microsoft Windows [Version 10.0.19041.867], locale en-US)
    • Flutter version 1.22.4 at C:\flutter
    • Framework revision 1aafb3a8b9 (5 months ago), 2020-11-13 09:59:28 -0800
    • Engine revision 2c956a31c0
    • Dart version 2.10.4

[v] Android toolchain - develop for Android devices (Android SDK version 30.0.1)
    • Android SDK at C:\Users\User\AppData\Local\Android\sdk
    • Platform android-30, build-tools 30.0.1
    • Java binary at: C:\Program Files\Android\Android Studio\jre\bin\java
    • Java version OpenJDK Runtime Environment (build 1.8.0_242-release-1644-b01)
    • All Android licenses accepted.

[v] Android Studio (version 4.0)
    • Android Studio at C:\Program Files\Android\Android Studio
    • Flutter plugin installed
    • Dart plugin version 193.7547
    • Java version OpenJDK Runtime Environment (build 1.8.0_242-release-1644-b01)

[v] VS Code (version 1.53.2)
```

Find binary path



```
C:\>cd "Program Files"
C:\Program Files>cd Android
C:\Program Files\Android>cd "Android Studio"
C:\Program Files\Android\Android Studio>cd jre
C:\Program Files\Android\Android Studio\jre>cd bin
C:\Program Files\Android\Android Studio\jre\bin>cd java
The system cannot find the path specified.
C:\Program Files\Android\Android Studio\jre\bin>cd java
The system cannot find the path specified.
C:\Program Files\Android\Android Studio\jre\bin>
```

Then generate and store the key (image on next page)

```

Microsoft Windows [version 10.0.19041.807]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\User>cd ..
C:\Users>cd ..
C:\>cd "Program Files"
C:\Program Files>cd Android
C:\Program Files\Android>cd "Android Studio"
C:\Program Files\Android\Android Studio>cd jre
C:\Program Files\Android\Android Studio\jre>cd bin
C:\Program Files\Android\Android Studio\jre\bin>cd java
The system cannot find the path specified.
C:\Program Files\Android\Android Studio\jre\bin>cd java
The system cannot find the path specified.
C:\Program Files\Android\Android Studio\jre\bin>keytool.exe -genkey -v -keystore c:\flutter_projects\active_ecommerce_flutter\key.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias key
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: Test
What is the name of your organizational unit?
[Unknown]: Test
What is the name of your organization?
[Unknown]: Test
What is the name of your City or Locality?
[Unknown]: Test
What is the name of your State or Province?
[Unknown]: Test
What is the two-letter country code for this unit?
[Unknown]: us
Is CN=Test, OU=Test, O=Test, L=Test, ST=Test, C=us correct?
[no]: yes
Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=Test, OU=Test, O=Test, L=Test, ST=Test, C=us
Enter key password for <key>
(RETURN if same as keystore password):
Re-enter new password:
[Storing c:\flutter_projects\active_ecommerce_flutter\key.jks]

```

Binary location

File path to save the key

#### active\_ecommerce\_flutter

Name	Date modified	Type	Size
.dart_tool	4/1/2021 5:58 PM	File folder	
.git	4/1/2021 7:41 PM	File folder	
.idea	4/1/2021 9:44 PM	File folder	
android	3/28/2021 9:59 PM	File folder	
assets	4/1/2021 5:53 PM	File folder	
build	4/1/2021 9:01 PM	File folder	
dummy_assets	3/31/2021 8:41 PM	File folder	
ios	11/29/2020 7:19 PM	File folder	
lib	4/1/2021 5:29 PM	File folder	
test	11/29/2020 7:19 PM	File folder	
.flutter-plugins	4/1/2021 9:00 PM	FLUTTER-PLUGINS...	3 KB
.flutter-plugins-dependencies	4/1/2021 9:00 PM	FLUTTER-PLUGINS...	8 KB
.gitattributes	11/29/2020 9:09 PM	Text Document	1 KB
.gitignore	11/29/2020 7:19 PM	Git Ignore Source ...	1 KB
.metadata	11/29/2020 7:19 PM	METADATA File	1 KB
.packages	4/1/2021 8:59 PM	PACKAGES File	15 KB
active_ecommerce_flutter.iml	1/17/2021 6:49 PM	IML File	1 KB
FlutterEcommerceAPI.postman_collectio...	3/25/2021 8:38 PM	JSON Source File	39 KB
key.jks	4/1/2021 9:56 PM	JKS File	3 KB
pubspec.lock	4/1/2021 8:59 PM	LOCK File	26 KB
pubspec.yaml	4/1/2021 8:45 PM	Yaml Source File	3 KB
README.md	11/29/2020 7:19 PM	Markdown Source...	1 KB

flutter\_projects > active\_ecommerce\_flutter >

Name	Date modified	Type	Size
.dart_tool	4/1/2021 10:38 PM	File folder	
.git	4/1/2021 10:51 PM	File folder	
.idea	4/3/2021 3:50 PM	File folder	
android	4/1/2021 10:44 PM	File folder	
assets	4/1/2021 5:53 PM	File folder	
build	4/1/2021 10:39 PM	File folder	
dummy_assets	3/31/2021 8:41 PM	File folder	
ios	11/29/2020 7:19 PM	File folder	
lib	4/1/2021 5:29 PM	File folder	
test	11/29/2020 7:19 PM	File folder	
.flutter-plugins	4/3/2021 3:51 PM	FLUTTER-PLUGINS...	3 KB
.flutter-plugins-dependencies	4/3/2021 3:51 PM	FLUTTER-PLUGINS...	8 KB
.gitattributes	11/29/2020 9:09 PM	Text Document	1 KB
.gitignore	11/29/2020 7:19 PM	Git Ignore Source ...	1 KB
.metadata	11/29/2020 7:19 PM	METADATA File	1 KB
.packages	4/1/2021 10:38 PM	PACKAGES File	15 KB
active_ecommerce_flutter.iml	1/17/2021 6:49 PM	IML File	1 KB
FlutterEcommerceAPI.postman_collectio...	3/25/2021 8:38 PM	JSON Source File	39 KB
key.jks	4/1/2021 9:56 PM	JKS File	3 KB
pubspec.lock	4/1/2021 10:38 PM	LOCK File	26 KB
pubspec.yaml	4/1/2021 8:45 PM	Yaml Source File	3 KB
README.md	11/29/2020 7:19 PM	Markdown Source...	1 KB

Then copy the key.jks from the root folder and paste it in the android/app folder

This PC > Local Disk (C:) > flutter\_projects > active\_ecommerce\_flutter > android > app >

Name	Date modified	Type	Size
src	11/29/2020 7:19 PM	File folder	
build.gradle	4/3/2021 3:50 PM	GRADLE File	3 KB
key.jks	4/1/2021 9:56 PM	JKS File	3 KB

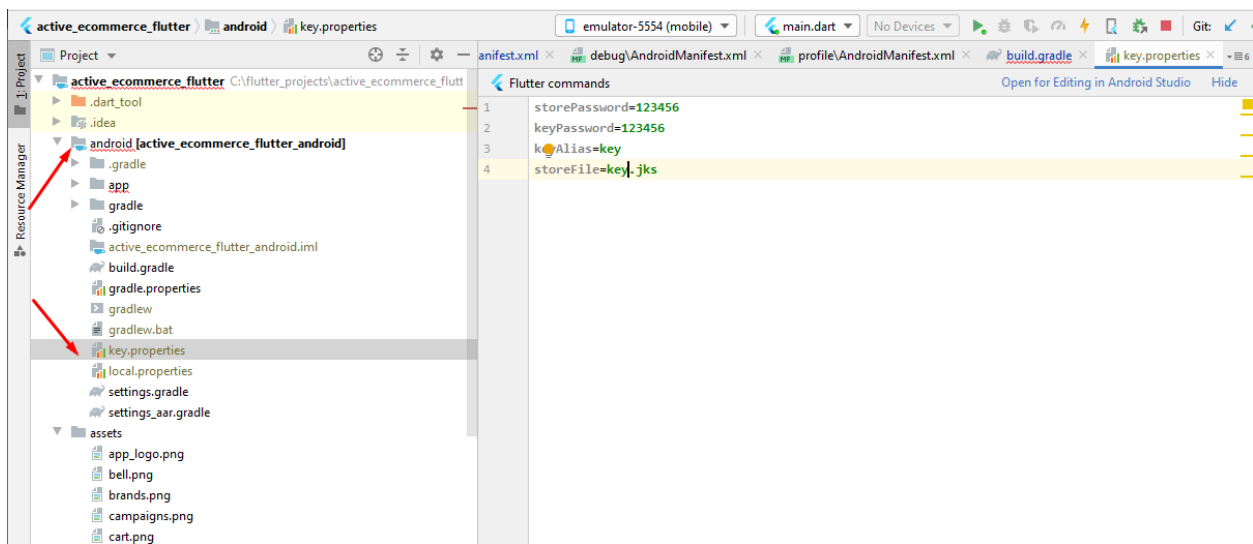
## Reference the keystore from the app

Create a file named `<your app dir>/android/key.properties` that contains a reference to your keystore:

```
storePassword=<password from previous step>
keyPassword=<password from previous step>
keyAlias=key
storeFile=<location of the key store file, such as /Users/<user name>/key.jks>
```

**\*\* If you lose the jks file , you will not be able to release a new update your app in playstore\*\***

Create new file `key.properties` in android folder . Enter the information



Read this

## Configure signing in gradle

Configure signing for your app by editing the `<your app dir>/android/app/build.gradle` file.

1. Add code before `android` block:

```
android {  
  ...  
}
```

With the keystore information from your properties file:

```
def keystoreProperties = new Properties()  
def keystorePropertiesFile = rootProject.file('key.properties')  
if (keystorePropertiesFile.exists()) {  
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))  
}  
  
android {  
  ...  
}
```

Load the `key.properties` file into the `keystoreProperties` object.

2. Add code before `buildTypes` block:

```
buildTypes {  
    release {  
        // TODO: Add your own signing config for the release build.  
        // Signing with the debug keys for now,  
        // so 'flutter run --release' works.  
        signingConfig signingConfigs.debug  
    }  
}
```

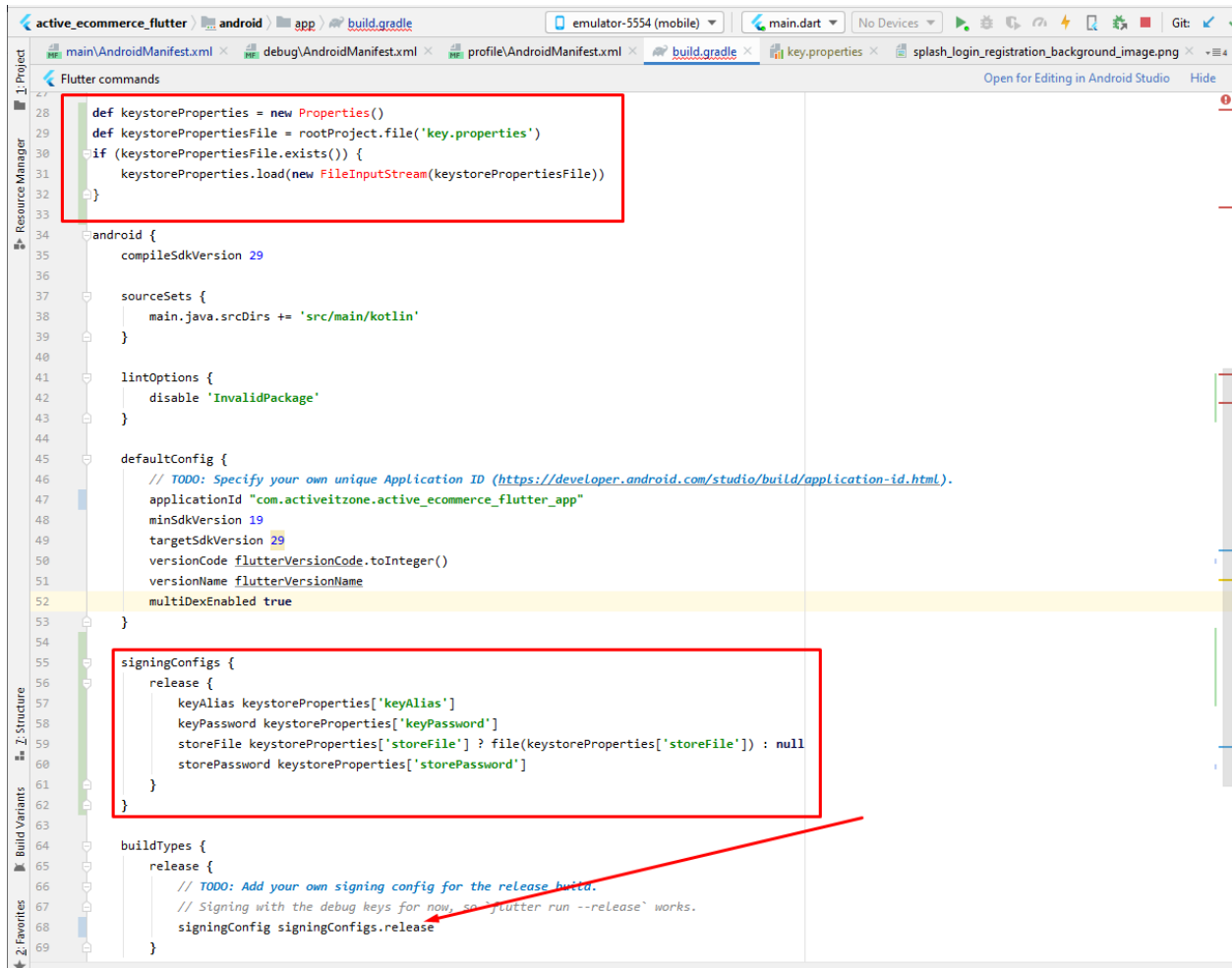
With the signing configuration info:

```
signingConfigs {  
    release {  
        keyAlias keystoreProperties['keyAlias']  
        keyPassword keystoreProperties['keyPassword']  
        storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null  
        storePassword keystoreProperties['storePassword']  
    }  
}  
buildTypes {  
    release {  
        signingConfig signingConfigs.release  
    }  
}
```

Configure the `signingConfigs` block in your module's `build.gradle` file.

Release builds of your app will now be signed automatically.

in app/build.gradle do necessary changes



**Note:** You may need to run `flutter clean` after changing the gradle file. This prevents cached builds from affecting the signing process.

Now you are almost done

In your terminal run : `flutter build appbundle`

The release bundle for your app is created at `<your app dir>/build/app/outputs/bundle/release/app.aab`.

Upload this app.aab file to your google play console

## 7. How to generate app store uploadable files?

Archive and upload your app using Xcode

- Before you can submit your app for review through App Store Connect, you need to upload the build through Xcode.
- In Xcode, select Generic iOS Device as the deployment target.
- Choose Product from the top menu and click on Archive.
- The Xcode Organizer will launch, displaying any archives you've created.
- Make sure the current build is selected and click on Upload to App Store in the right-hand panel.
- Select your credentials and click Choose.
- In the next window that appears, click on Upload in the bottom right-hand corner.

A success message will appear when the upload has been completed. Click Done.

## 8. How to update for android? **\*\*Read all the points carefully before doing anything**

- This section will help you if you are here for the update and have already generated the signed release apk/appbundle the last time and already have the keytool and the manifest file ready in your old project folder.
- If you are installing and building the release file for the first time this section is not for you.
- Extract the source\_code.zip. You will find this inside the main zip.
- Open the folder in your android studio.
- **\*\*Remember to open this in a separate folder than your old project.**
- Even if you are building an app for ios, use android studio for the build.
- Then in your android studio terminal run:  
`flutter pub get`
- This will fetch all the necessary packages

- If you are updating, you must have build the key.jks previously
  - Copy the key.jks , key.properties, and the manifest file from your old project and paste in the correct locations
  - See the previous screenshots for the file locations
  - If you are missing your old project, you have to configure key.properties, and the manifest file like described in the installation.
  - As our source code is made ready for the fresh installation , you will have to do all your configuration (like domain path, app color, package name etc ) shown in the previous steps.
  - But do not create a new key.jks, you have to update your app with the existing key
  - If you have somehow lost your previous key , you have to release a totally new app to the play store. You will not be able to release an update.
- 
- In your terminal run : `flutter build appbundle`
  - The release bundle for your app is created at `<your app dir>/build/app/outputs/bundle/release/app.aab`.
  - Upload this app.aab file to your google play console

## 9. How to configure google map? (Read the whole thing before implementing)

Go to your flutter project->lib->app\_config.dart

```
static String default_language ="en";
static String mobile_app_code ="en";
static bool app_language_rtl =false;
```



```
import 'package:flutter/material.dart';

var this_year = DateTime.now().year.toString();

class AppConfig {
  static String copyright_text = "@ ActiveItZone " + this_year; //this shows in the splash
  static String app_name = "Active eCommerce"; //this shows in the splash screen
  static String purchase_code = ""; //enter your purchase code for the app from codecanyon
  //static String purchase_code = ""; //enter your purchase code for the app from codecany

  //Default language config
  static String default_language = "en";
  static String mobile_app_code = "en";
  static bool app_language_rtl = false;

  //configure this
  static const bool HTTPS = false;
}
```

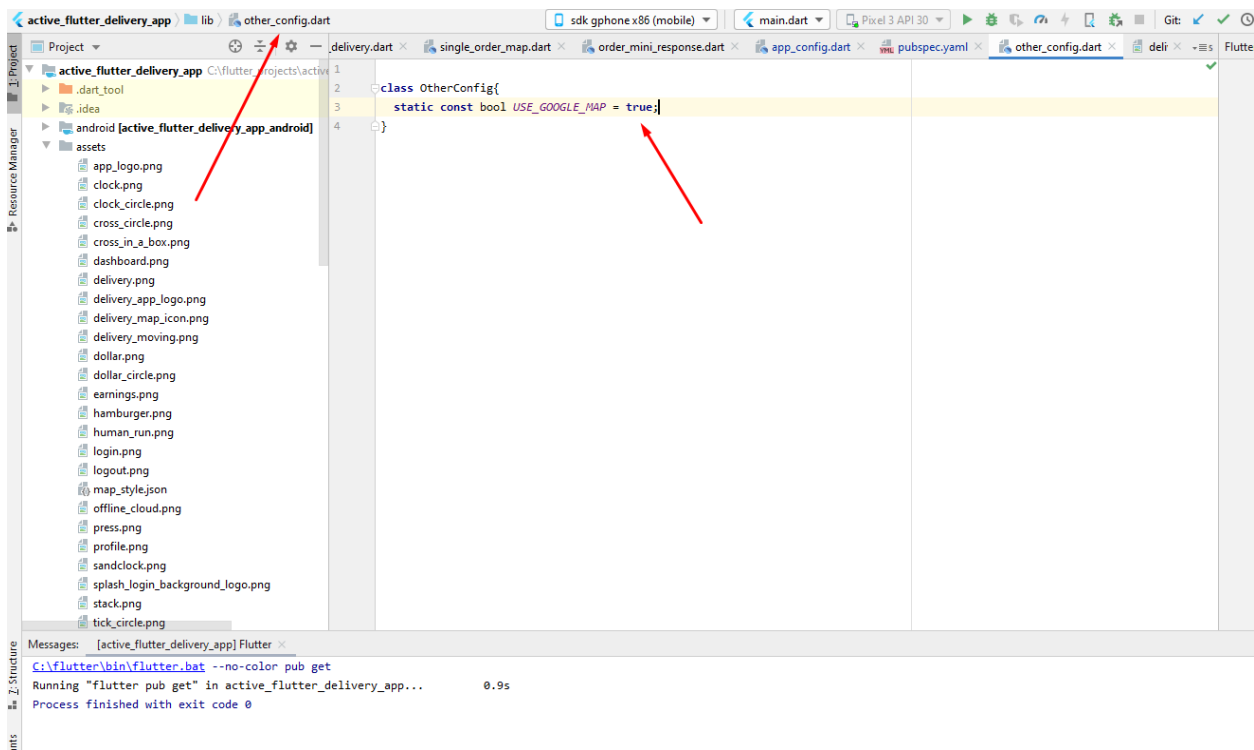
This value you can find in your admin panel. Go to your admin panel->setup & configurations->languages.

#	Name	Code	Flutter App Lang Code	RTL	Options
1	English	en	en	<input type="checkbox"/>	
2	Bangla	bd	bn	<input type="checkbox"/>	
3	Arabic	sa	ar	<input checked="" type="checkbox"/>	

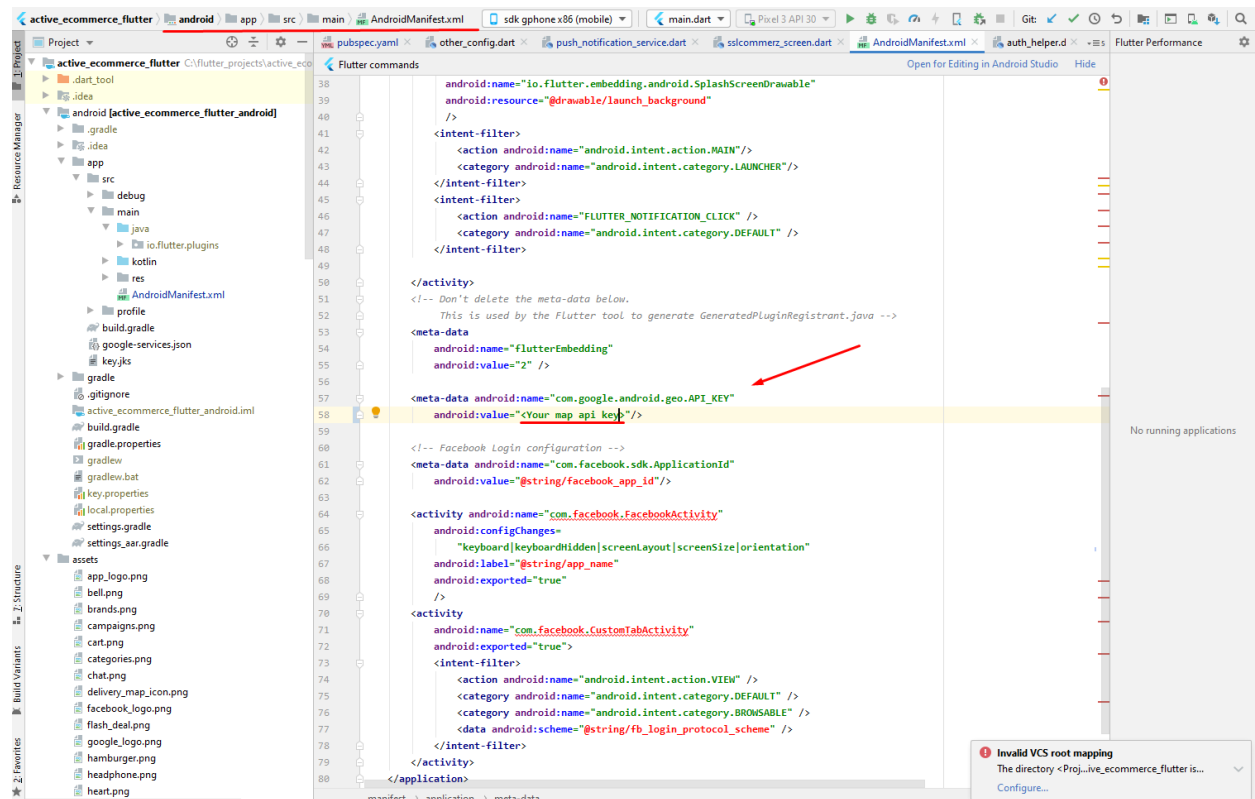
## 10. How to configure google maps? (Read the whole thing before implementing)

1. Go to <https://console.developers.google.com/> and generate api keys separately for ios and android. No restrictions are needed

## 1. In lib/other\_config.dart make, use google map = true



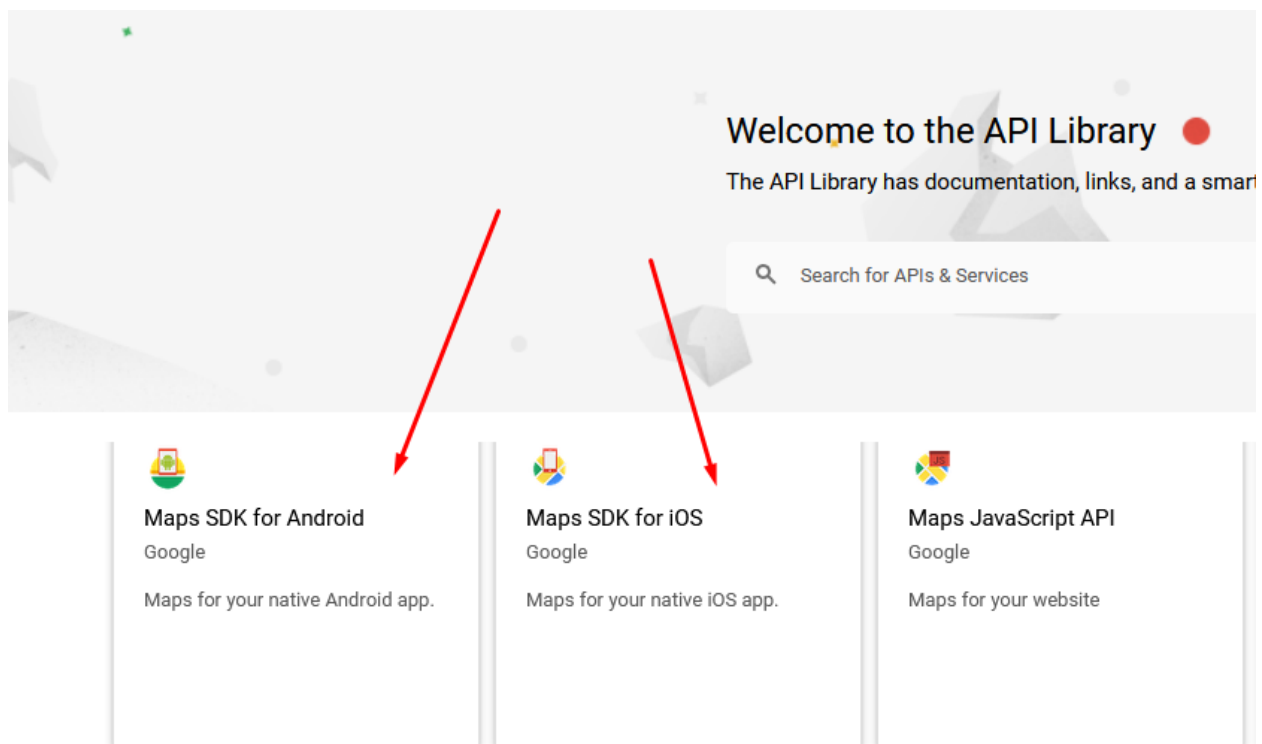
## 2. In main AndroidManifest.xml put the map api key



3. For ios follow this

<https://blog.logrocket.com/adding-google-maps-to-a-flutter-app/#addinggooglemapstoflutterios>

4. Enable android and ios api. These are free.



Machine learning

5. If you need to style the map then go to here <https://mapstyle.withgoogle.com/>  
Generate style according to your needs.

Copy the generated json and **replace** the copied code in assets/map\_style.json

In console run:

**flutter clean**

**flutter pub get**

