Documentation

Author : Active IT zone
Software Framework : Flutter
Addon For: Active eCommerce CMS
Provided by : codecanyon

Documentation

1. What are the prerequisites?

This Flutter app can be hosted into Google Play Store + Apple App Store 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 3.10.4 • 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?

- Extract the scource_code.zip. You will find this inside the main 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?

A) **App Config:**

This helps you connect your app to your server.

Open lib/app_config.dart

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

Do not change the other variables.

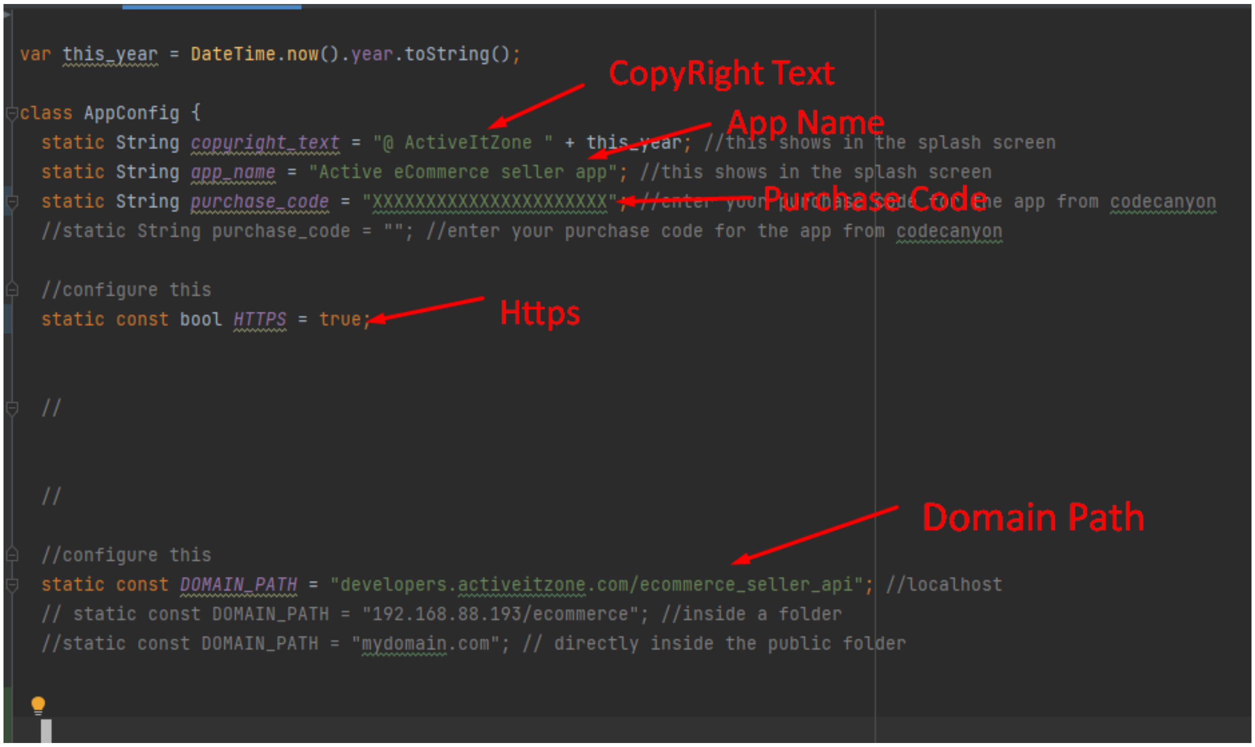
Make sure that purchase_code is given. Otherwise your app will not work properly.

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 **



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

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

  //configure this
  static const bool HTTPS = true;


  //


  //

  //configure this
  static const DOMAIN_PATH = "developers.activeitzone.com/ecommerce_seller_api"; //localhost
  // static const DOMAIN_PATH = "192.168.88.193/ecommerce"; //inside a folder
  //static const DOMAIN_PATH = "mydomain.com"; // directly inside the public folder
```

B) **Theme Config:**

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

Open lib/my_theme.dart

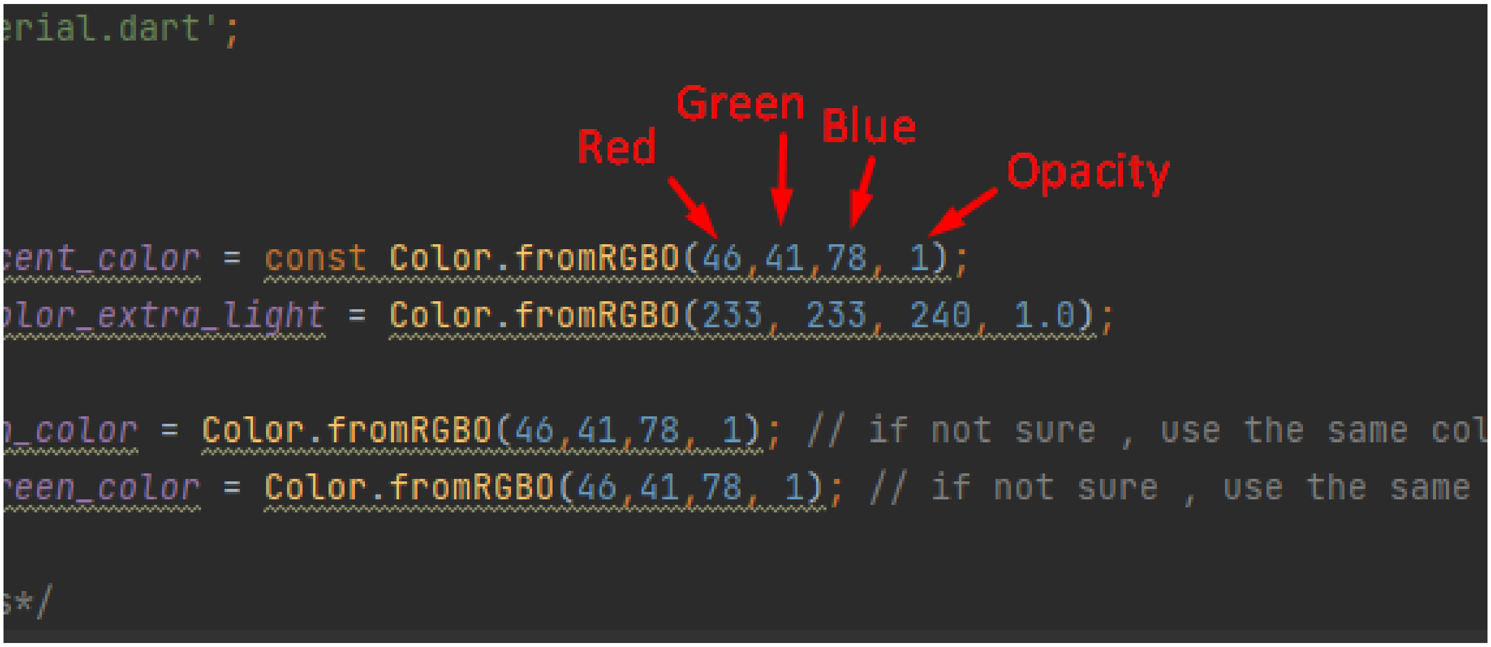
You can change the app_accent_color, app_accent_color_extra_light, splash_screen_color, login_reg_screen_color variable.

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.



```
erial.dart';

cent_color = const Color.fromRGBO(46,41,78, 1);
olor_extra_light = Color.fromRGBO(233, 233, 240, 1.0);

_color = Color.fromRGBO(46,41,78, 1); // if not sure , use the same col
reen_color = Color.fromRGBO(46,41,78, 1); // if not sure , use the same

s*/
```

C) Configure the launcher icon:

This helps you change your app's launcher icon.

Change the app_logo.png in assets/logo 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.

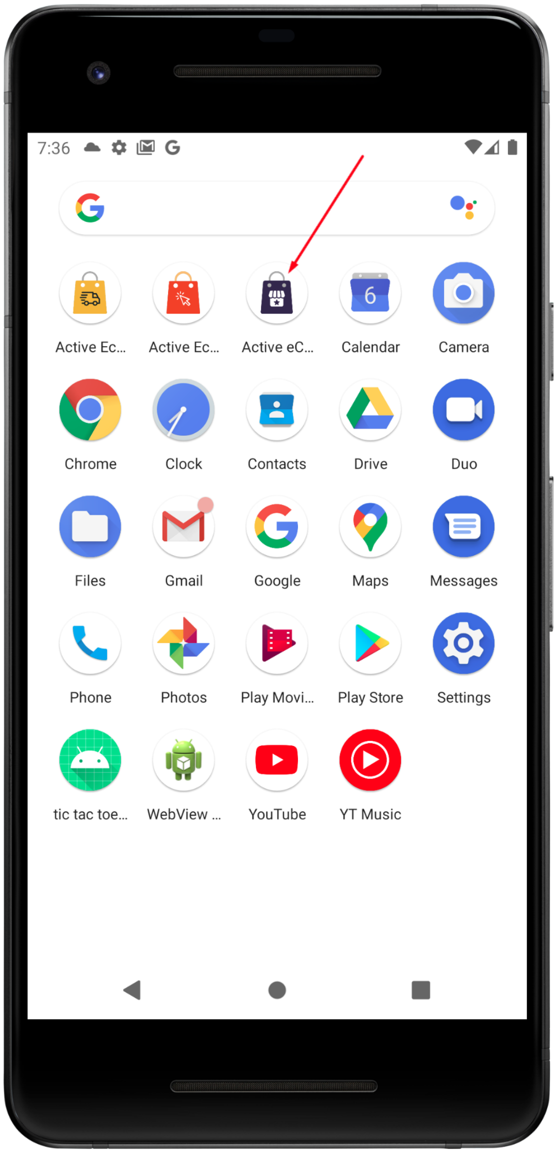
Then in your android studio terminal run:

```
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.

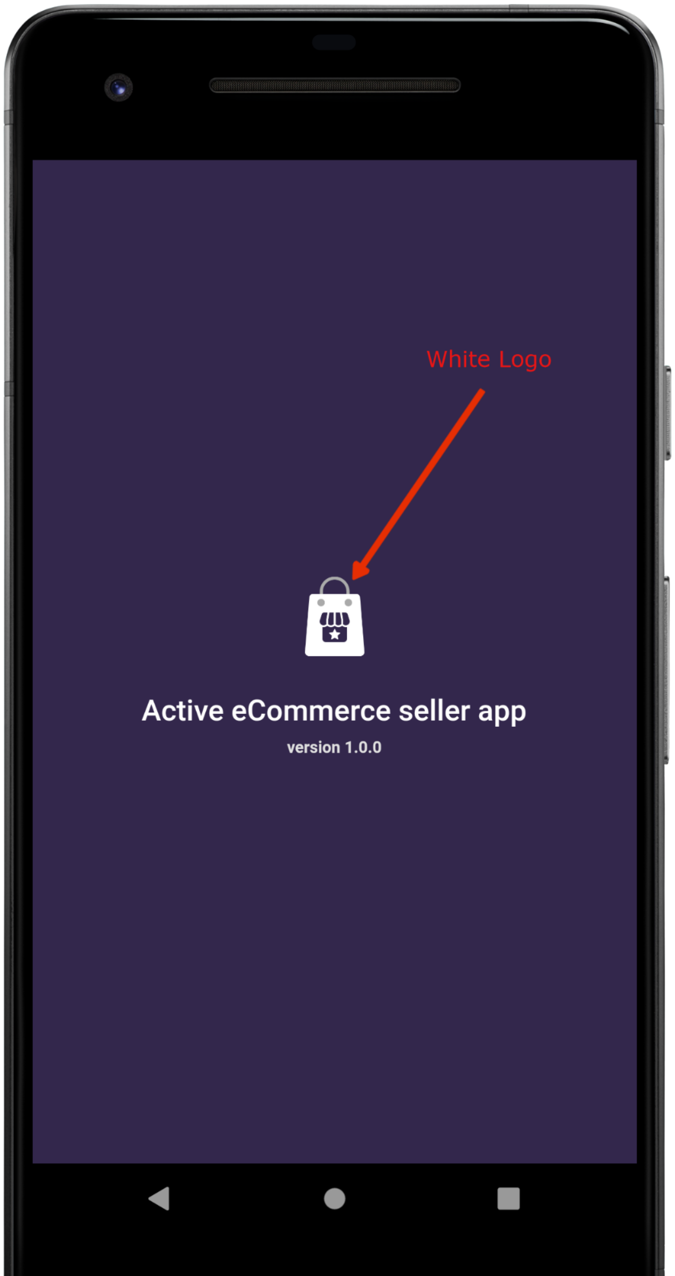


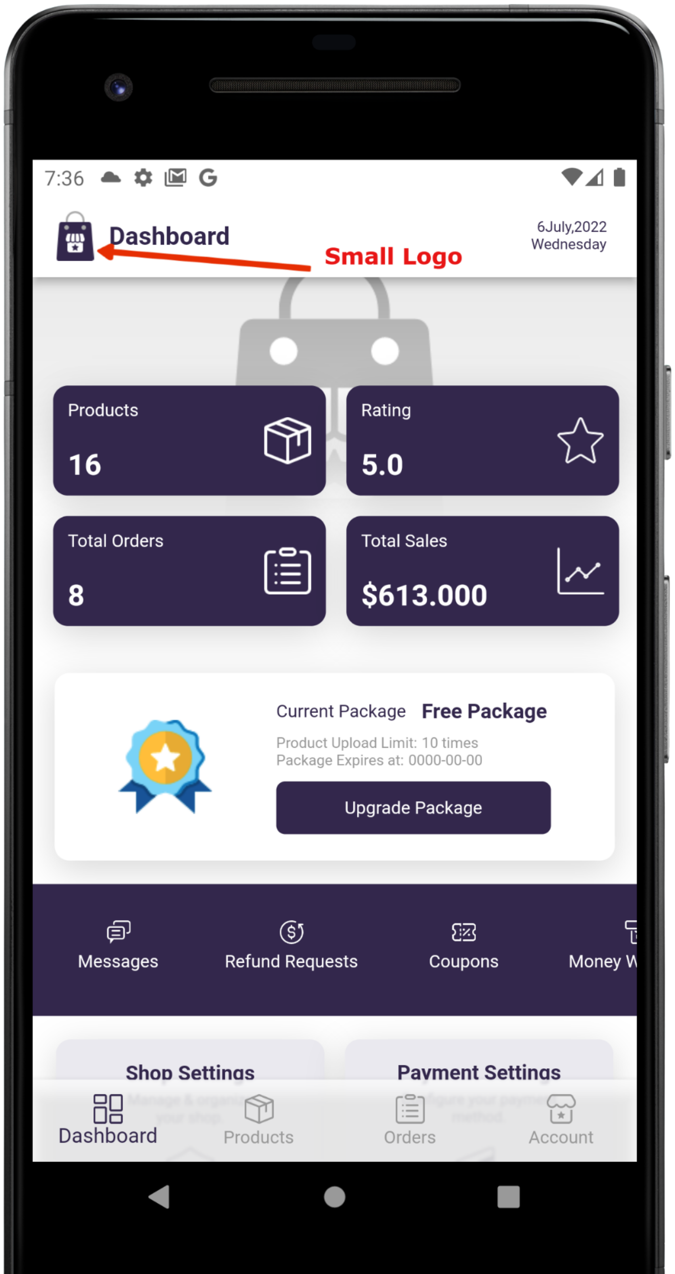
D) Configure other logos:

In the asset/logo folders we have other logos that you may want to change according to your branding.

This logos will be found in :

assets/logo/app_logo.png (512x512)
assets/logo/app_logo_small.png (48x64)
assets/logo/white_logo.png (256x256)
assets/logo/placeholder.png (512x512)





Change these logos with your own logos.File name , image format and size should be the same for each logo.

Then in your android studio terminal run:

```
flutter pub get
```

Then restart your app (shift +10). You should see your own logo in these places.

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 (anddroid/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 activate an app?

Before running the app you need to activate your flutter app otherwise the app will show an inactivated screen.

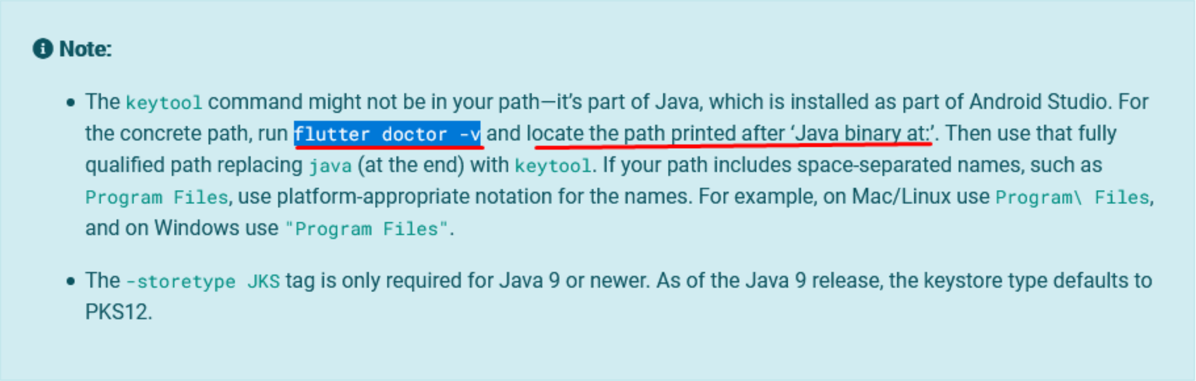
1. Go to this link https://activeitzone.com/activation/addon
2. Put your name in the name field.
3. Put your email in the email field.
4. Put your purchase code in the purchase field.
5. Put your codecanyon user name in the codecanyon user name field.
6. Put your domain name in the domain field.
7. Select **Active Ecommerce** option from main item.
8. Select the **eCommerce Seller App** option from the item.
9. Click the **Activate** button After filling all the fields and activate your app.

7. How to generate play store uploadable files for release?

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

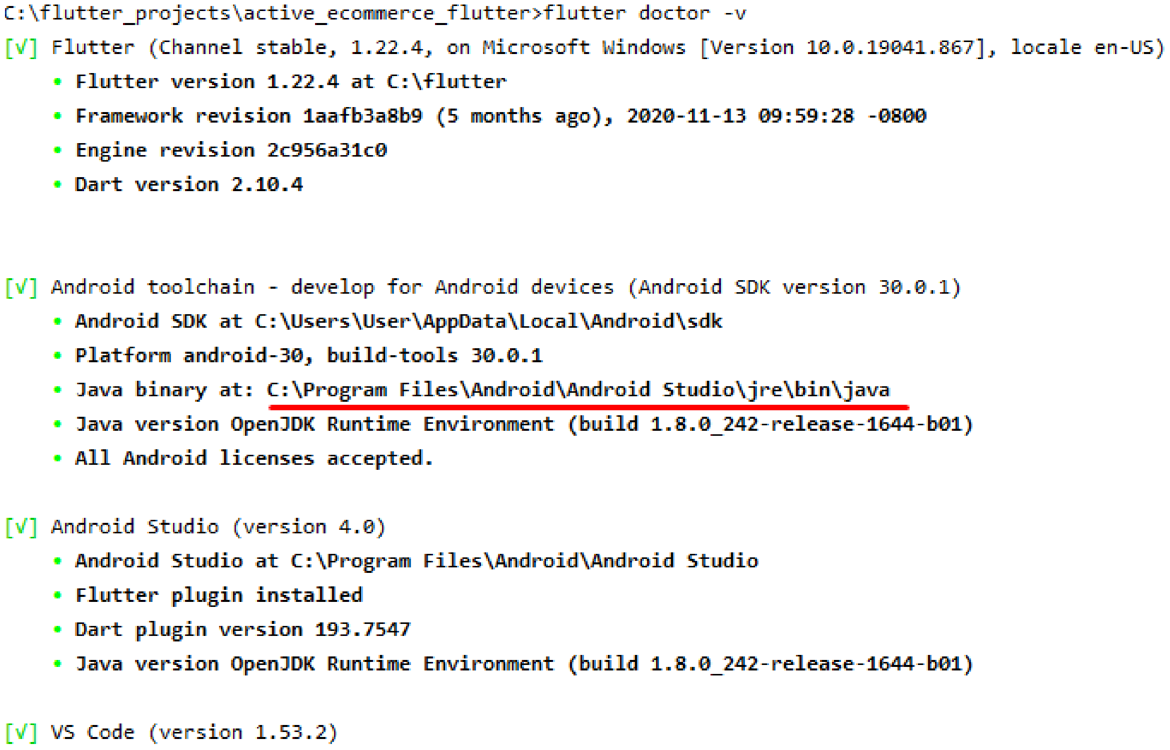
Signing the app:
To publish on the Play Store, you need to give your app a digital signature. Use the following instructions to sign your app.

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



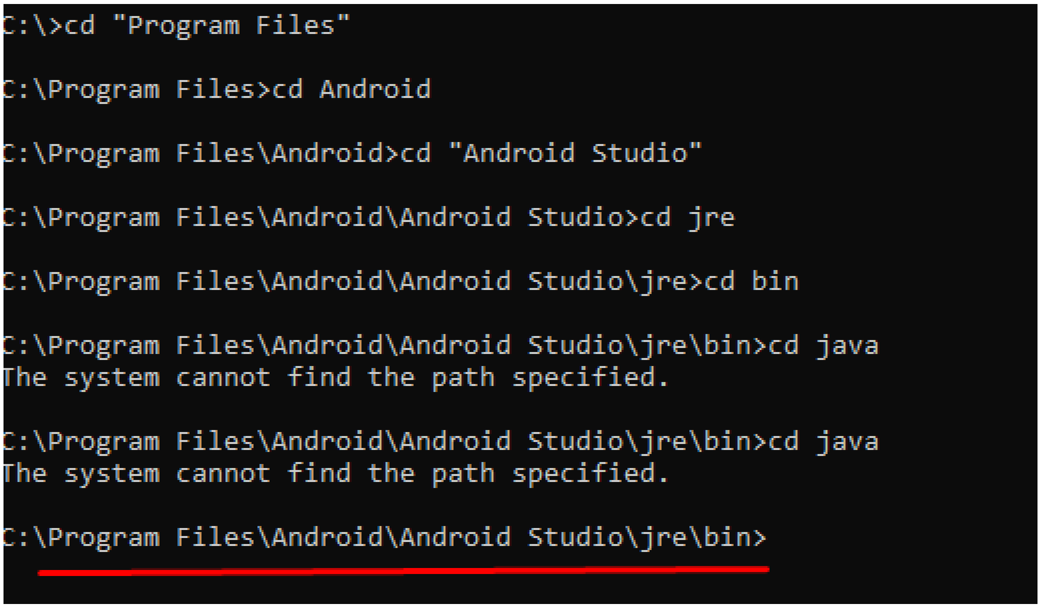
Note:

- 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.



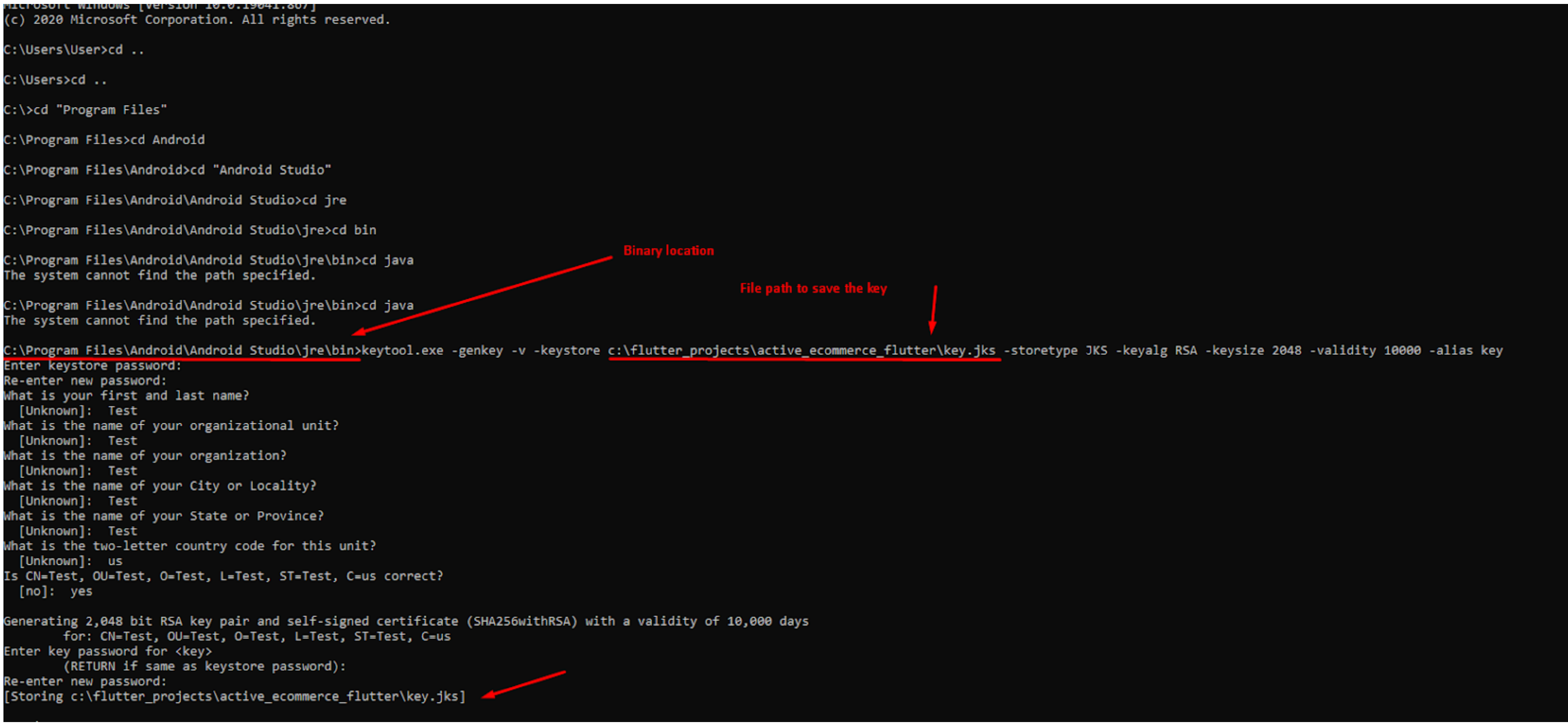
```
C:\flutter_projects\active_ecommerce_flutter>flutter doctor -v
[√] Flutter (Channel stable, 1.22.4, on Microsoft Windows [Version 10.0.19041.867], locale en-US)
    • Flutter version 1.22.4 at C:\flutter
    • Framework revision 1aafb3a8b9 (5 months ago), 2020-11-13 09:59:28 -0800
    • Engine revision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


[√] 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.

[√] 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)

[√] VS Code (version 1.53.2)
```

Find binary path

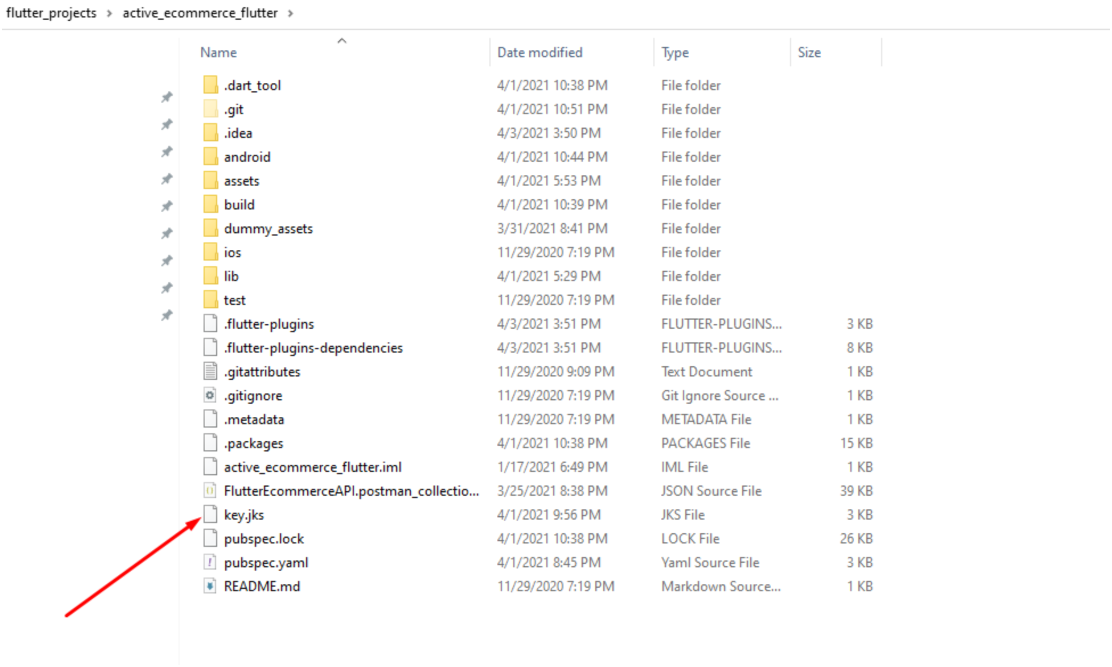


```
C:\>cd "Program Files"

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)

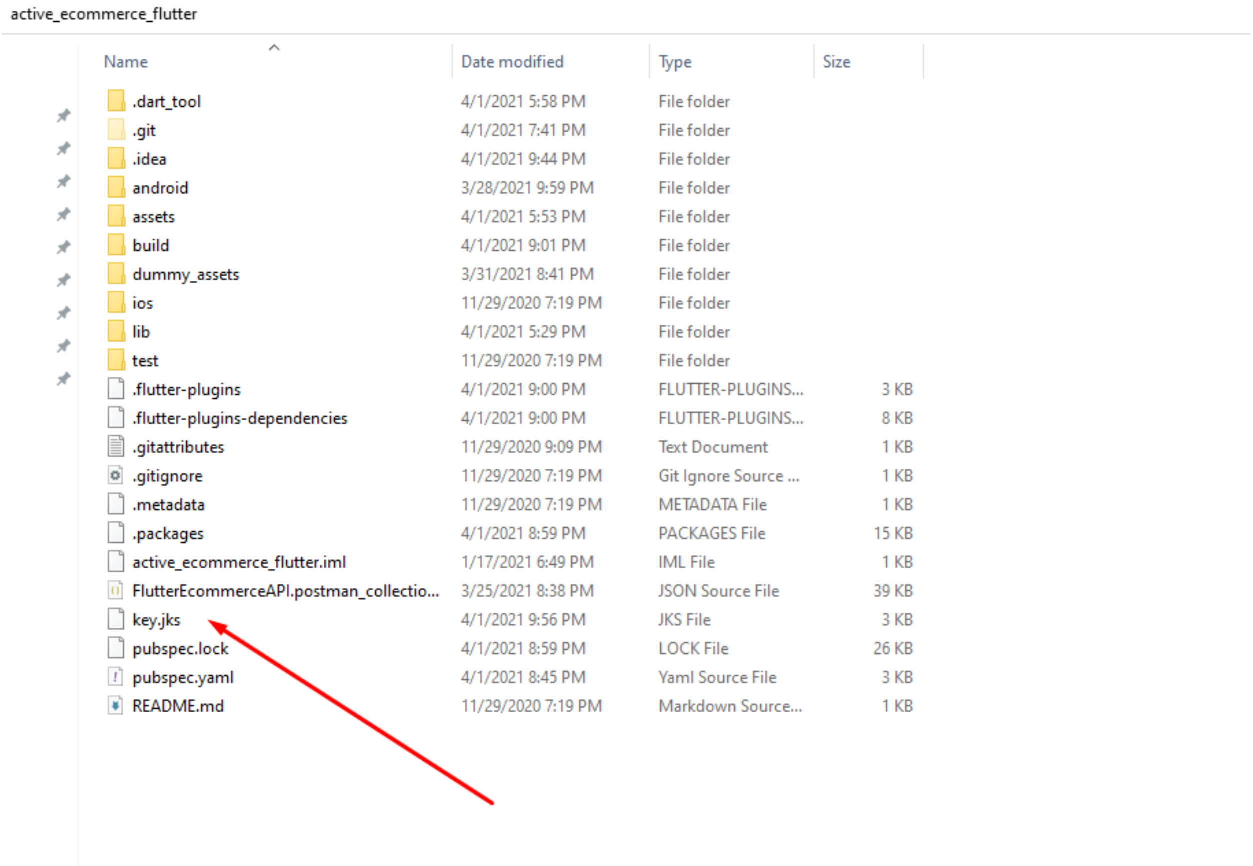


```
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. All rights reserved.

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]
```



flutter_projects › active_ecommerce_flutter ›

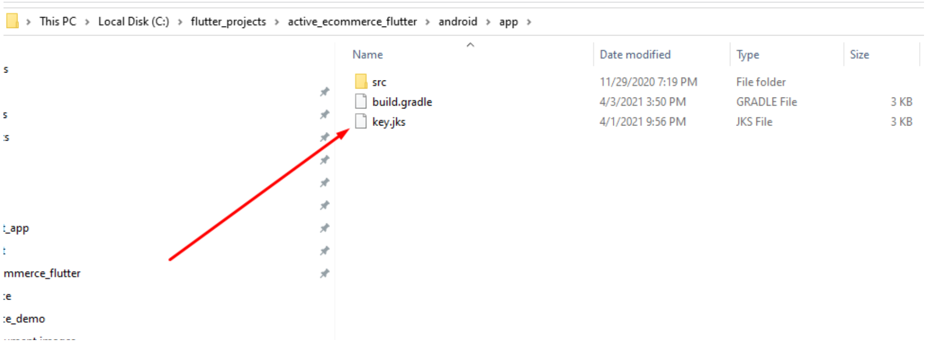
Name	Date modified	Type	Size
.dart_tool	4/1/2021 10:38 PM	File folder	
.git	4/1/2021 10:51 PM	File folder	
.idea	4/3/2021 3:50 PM	File folder	
android	4/1/2021 10:44 PM	File folder	
assets	4/1/2021 5:53 PM	File folder	
build	4/1/2021 10:39 PM	File folder	
dummy_assets	3/31/2021 8:41 PM	File folder	
ios	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



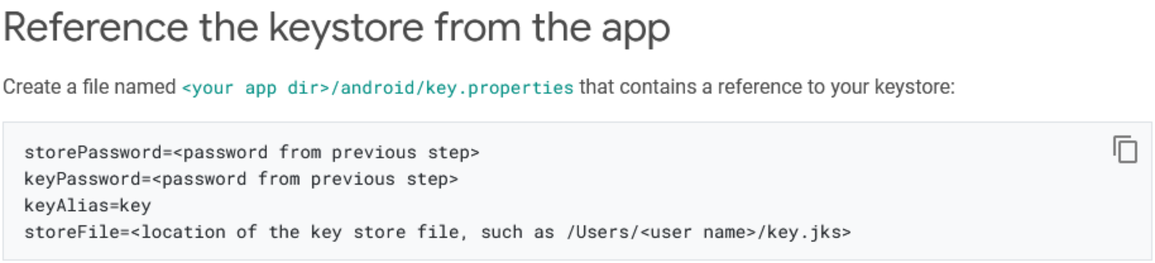
active_ecommerce_flutter

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

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



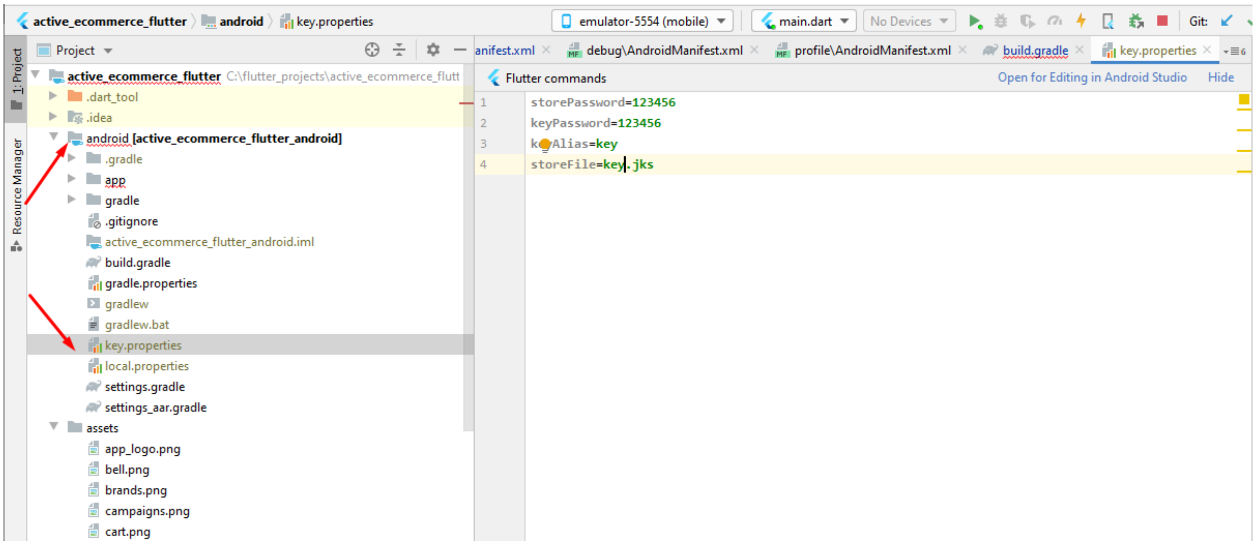
Reference the keystore from the app

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>
```

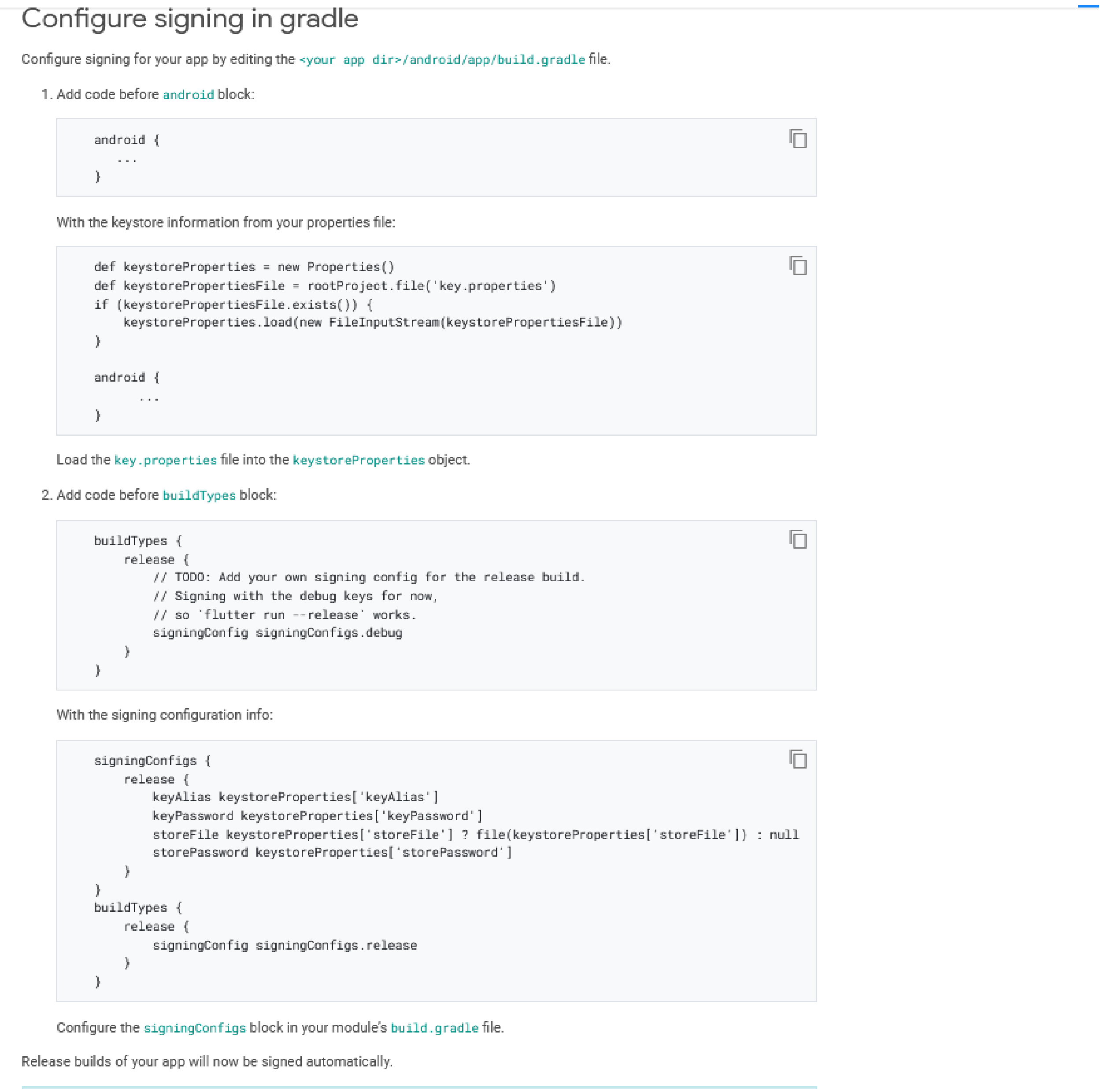
**** If you lose the jks file , you will not be able to release a new update your app in playstore****

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



```
storePassword=123456
keyPassword=123456
keyAlias=key
storeFile=key.jks
```

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
    }
}
```

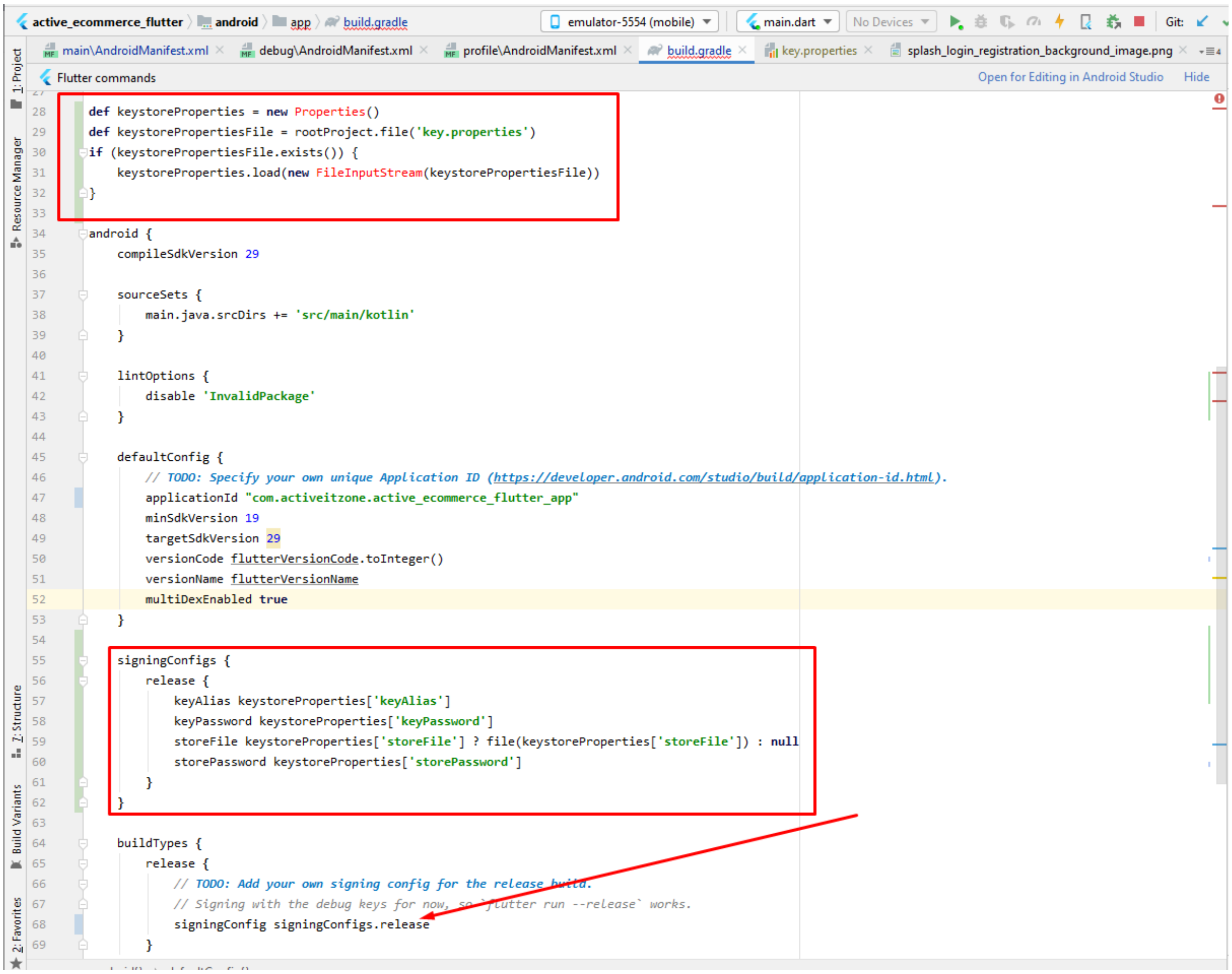
With the signing configuration info:

```
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



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

android {
    compileSdkVersion 29

    sourceSets {
        main.java.srcDirs += 'src/main/kotlin'
    }

    lintOptions {
        disable 'InvalidPackage'
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
        applicationId "com.activeitzone.active_ecommerce_flutter_app"
        minSdkVersion 19
        targetSdkVersion 29
        versionCode flutterVersionCode.toInteger()
        versionName flutterVersionName
        multiDexEnabled true
    }

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

    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.release
        }
```

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

8. 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.

9. 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 scource_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:

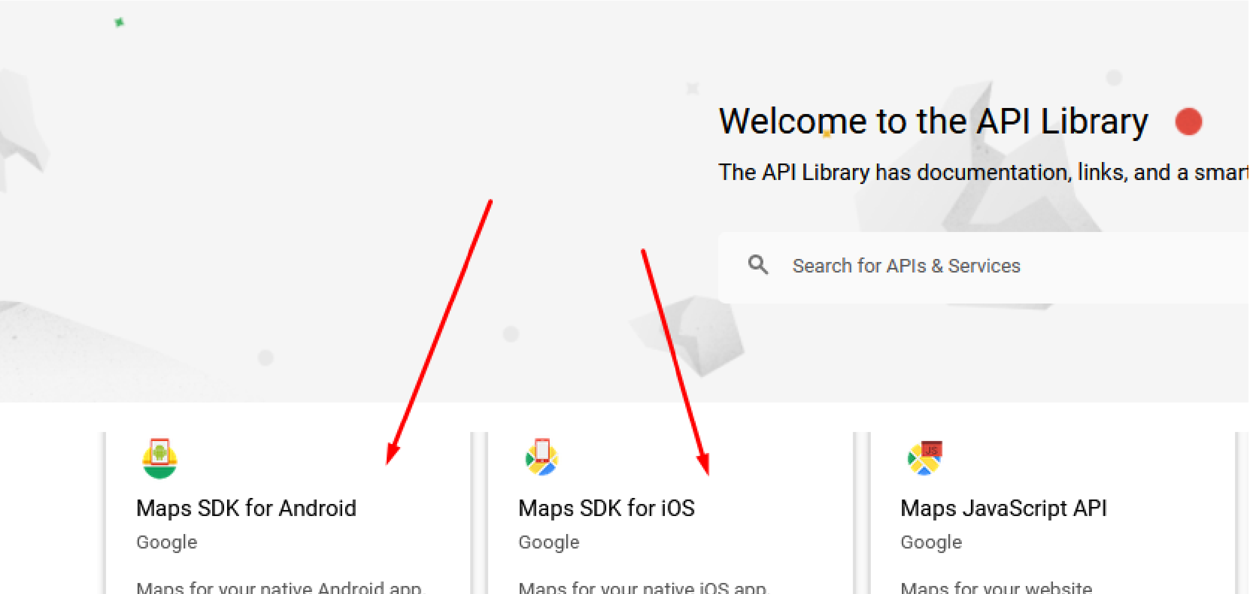
  ```
  flutter pub get
  ```
- This will fetch all the necessary packages

- 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

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

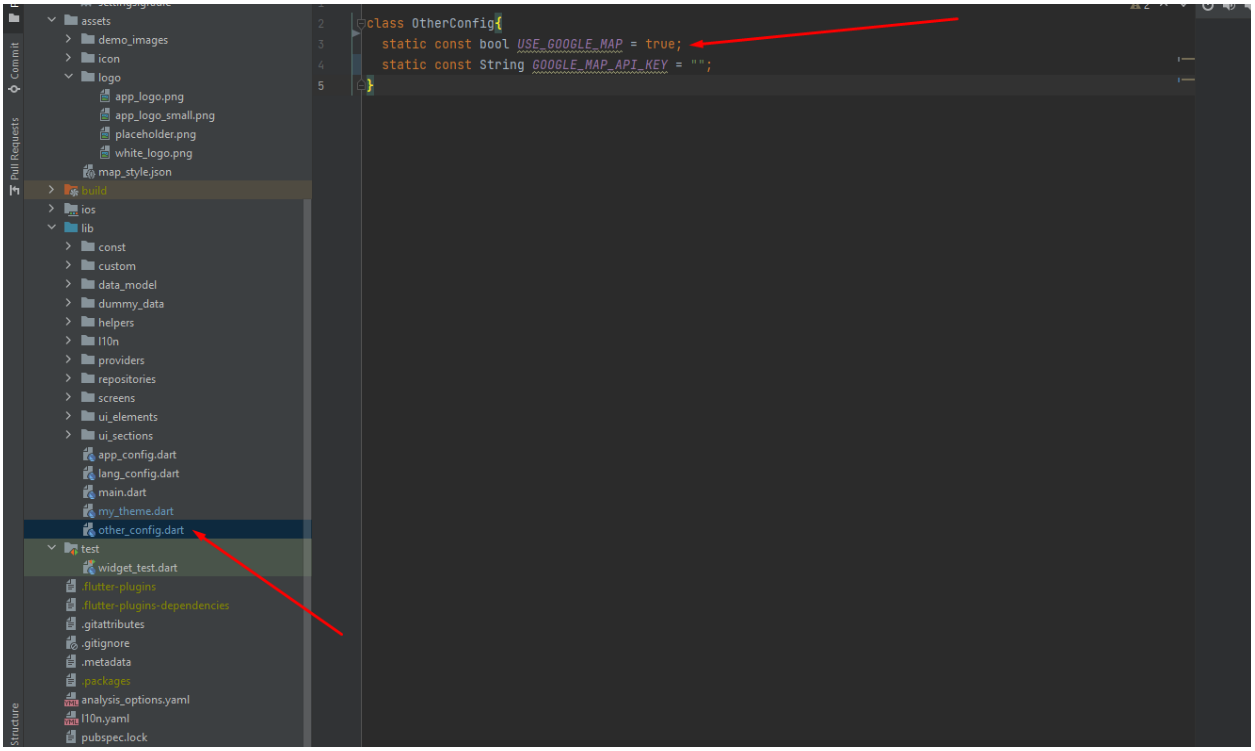
1. Go to https://console.developers.google.com/ and generate api keys separately for ios and android. No restrictions are needed



Welcome to the API Library

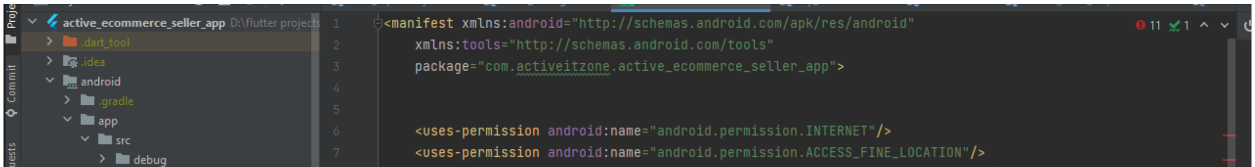
The API Library has documentation, links, and a smart

2. In lib/other_config.dart make, use google map = true and put your google map api key



```
class OtherConfig{
  static const bool USE_GOOGLE_MAP = true;
  static const String GOOGLE_MAP_API_KEY = "";
}
```

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



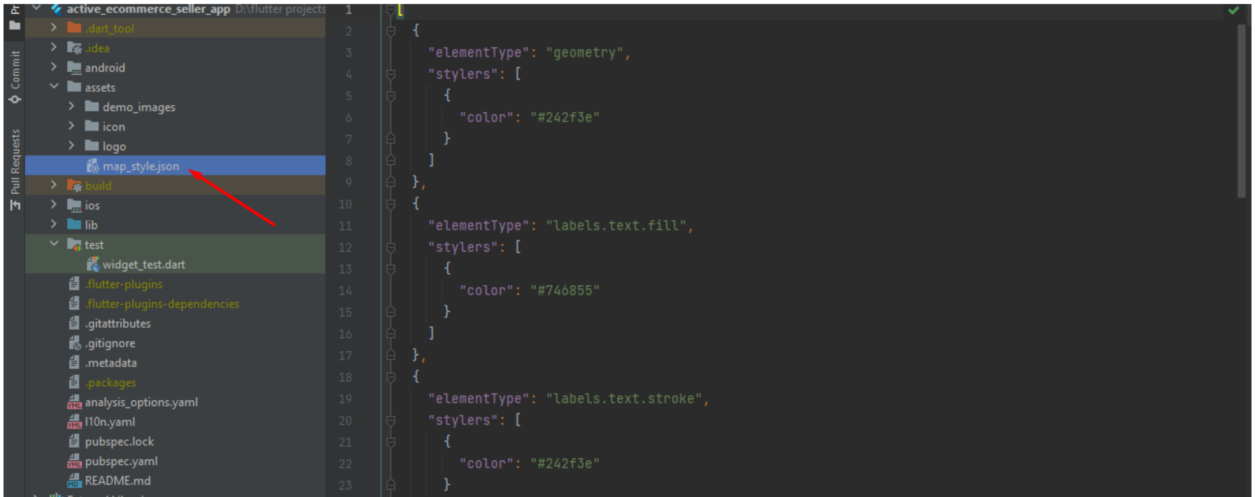
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.activeitzone.active_ecommerce_seller_app">


    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

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.

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



```
[
  {
    "elementType": "geometry",
    "stylers": [
      {
        "color": "#242f3e"
      }
    ]
  },
  {
    "elementType": "labels.text.fill",
    "stylers": [
      {
        "color": "#746855"
      }
    ]
  },
  {
    "elementType": "labels.text.stroke",
    "stylers": [
      {
        "color": "#242f3e"
      }
```

In console run:

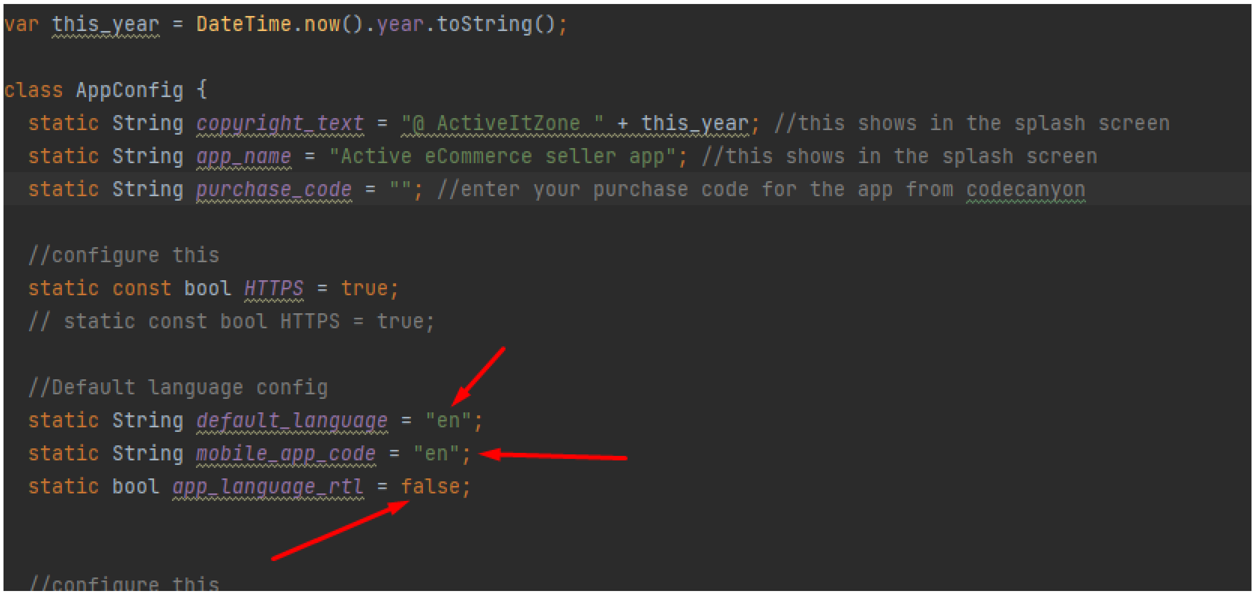
```
flutter  clean
flutter pub get
```

11. How to configure the default language for mobile apps?

Go to your flutter project->lib->app_config.dart

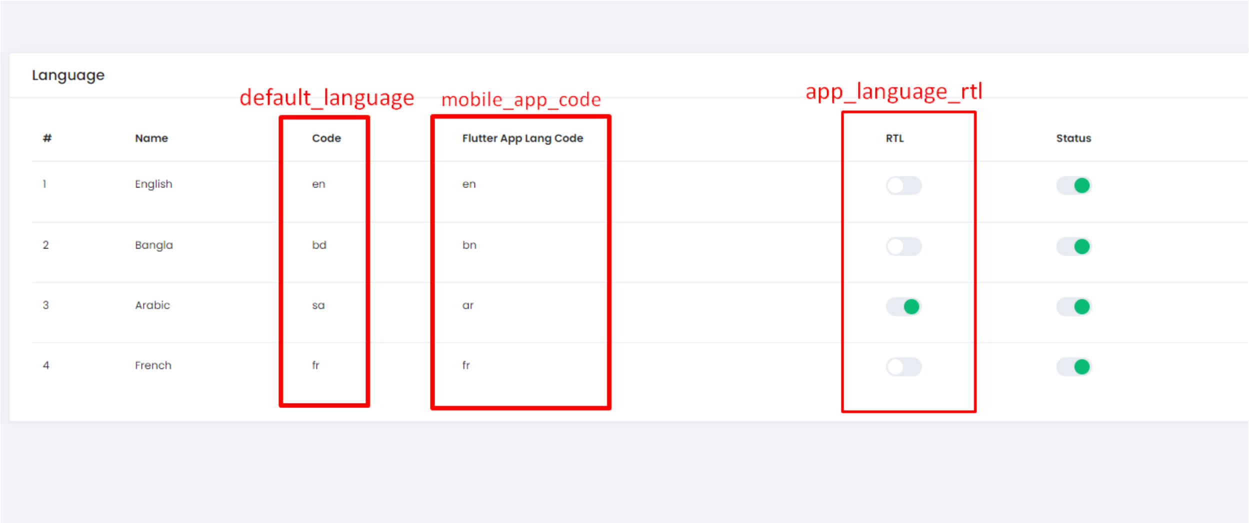
Change variables value `//Default language config`

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



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

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

  //configure this
  static const bool HTTPS = true;
  // static const bool HTTPS = true;

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


  //configure this
```

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



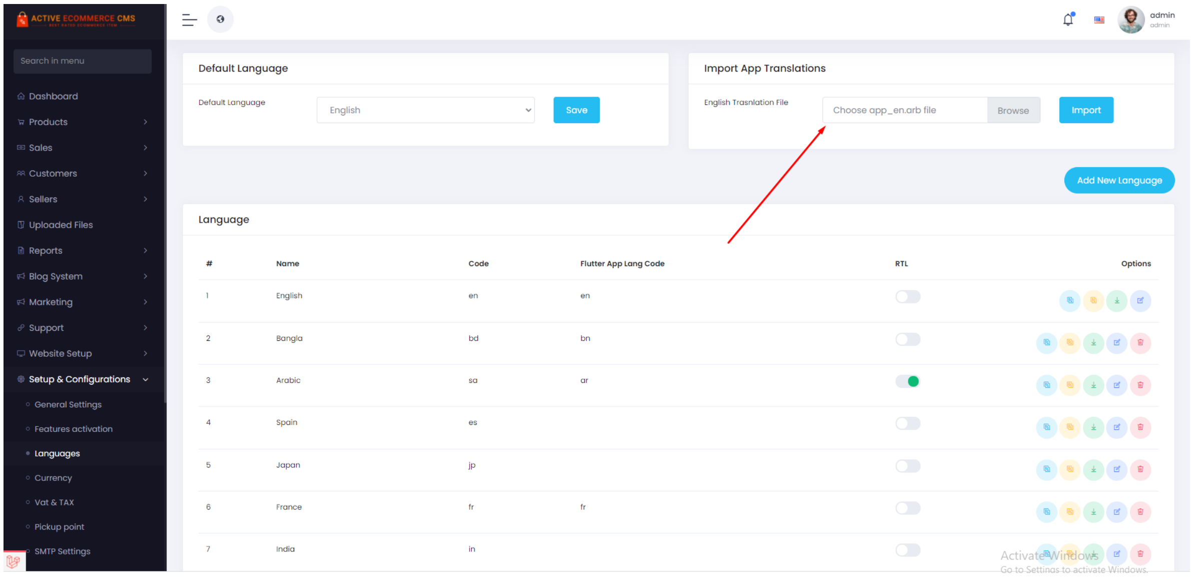
Language

default_language mobile_app_code app_language_rtl

#	Name	Code	Flutter App Lang Code	RTL	Status
1	English	en	en	○	●
2	Bangla	bd	bn	○	●
3	Arabic	sa	ar	●	●
4	French	fr	fr	○	●

12. How to configure multiple languages for mobile apps? (Read the whole thing before implementing)

1. In your **lib/l10n** folder you will see an **app_en.arb** file. This is your main translation + interpretation file. Never delete this.NEVER.
2. If you want another language file you can copy the app_en.arb file and make another language file like app_fr.arb and so on. But we will suggest that you use our translation generator from the admin panel.
3. Always make sure your language code is valid. https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes Use iso 639-1 codes. By default flutter localization uses 78 major language codes from here.
4. Upload **app_en.arb** in the admin panel.It will fetch strings from the file and uploads to your database.



Default Language

Default Language: English

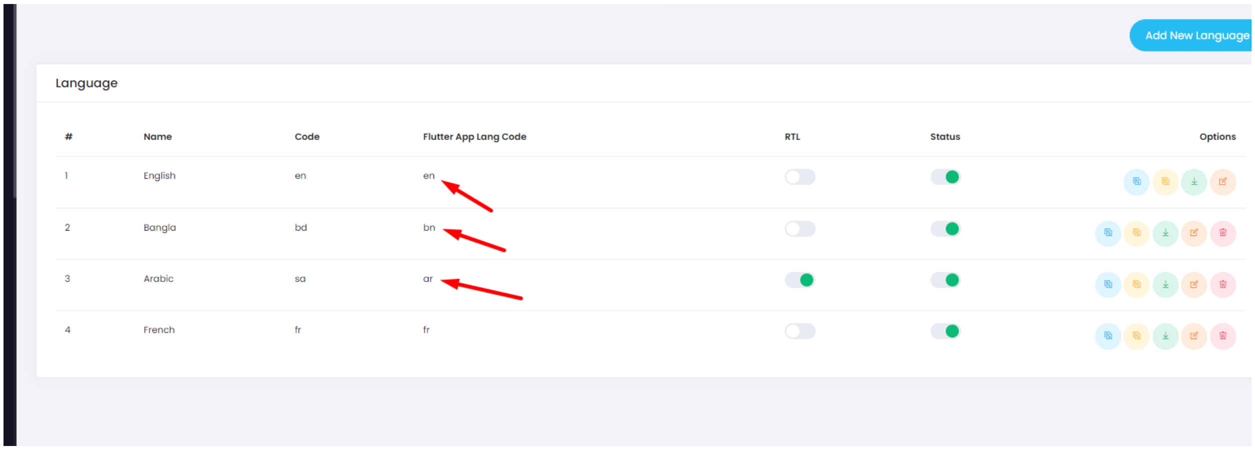
Import App Translations

English Trasnlation File: Choose app_en.arb file

Language

#	Name	Code	Flutter App Lang Code	RTL	Options
1	English	en	en	○	
2	Bangla	bd	bn	○	
3	Arabic	sa	ar	●	
4	Spain	es		○	
5	Japan	jp		○	
6	France	fr	fr	○	
7	India	in		○	

5. Make sure while adding/editing a language , your flutter app language code exists. The code must be in iso 639-1 format. Without a valid code , you will not see a translated output in the app.



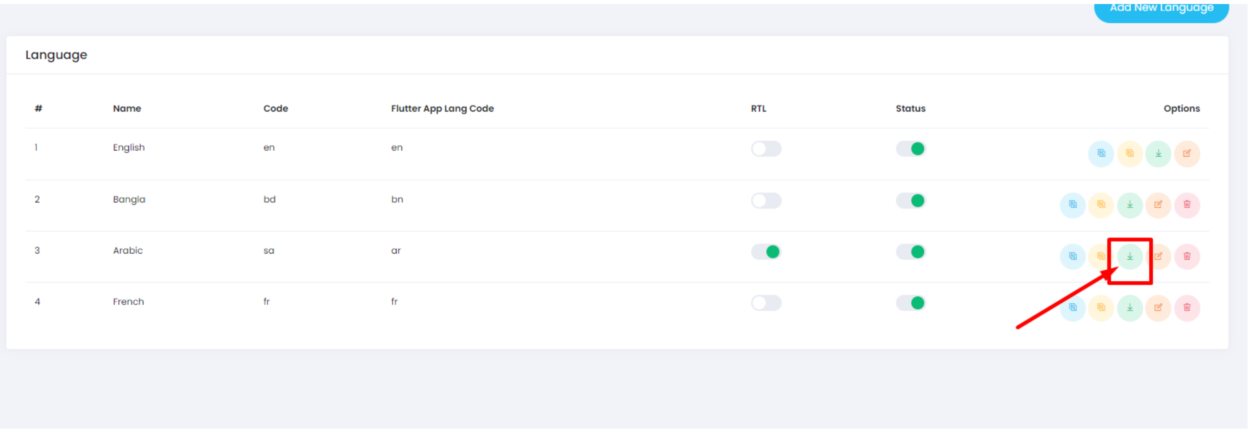
Add New Language

Language

#	Name	Code	Flutter App Lang Code	RTL	Status	Options
1	English	en	en	○	●	
2	Bangla	bd	bn	○	●	
3	Arabic	sa	ar	●	●	
4	French	fr	fr	○	●	

6.Then translate your app strings like you did for your web. You can use google chrome's translation extension and the copy button for a faster output. See, our documentation on translation is provided with the cms. Remember the translations for web and app are kept separate, so even if you did create the translation, for the web , you have to create it for the mobile app too.

7.Once all the strings are converted for a particular language , say for example french, you can download the app_fr.arb file from the panel and put this arb file in your flutter apps **lib/l10n** folder along with your **app_en.arb** file.You can also change the main **app_en.arb** file this way but we encourage you **not to do it** . If you face any error due to app_en.arb file changes , we will not provide you any support.
Make sure the file you pasted in the **lib/l10n** is not empty.If you provide an empty file you will get errors.



Add New Language

Language

#	Name	Code	Flutter App Lang Code	RTL	Status	Options
1	English	en	en	○	●	
2	Bangla	bd	bn	○	●	
3	Arabic	sa	ar	●	●	
4	French	fr	fr	○	●	

8. For the same language , your language code for app and web can be different. This is not an issue. But you have to make sure the code for the app is in 639-1 format.

9.The language list to the app is shown from the backend api, so if you are using a lot of languages , make sure you provide translation for all of them.If you don’t , by default the text from app_en.arb will be shown.

13. How to remove cache data.

To enrich user experience we have cached (Mostly for a day) a lot of api responses. If you think your app data is not changing even after your data has been changed from the backend, try clearing cache from the admin panel.There is a big red button on the top navbar in the admin panel to clear cache.