envato

codecanyon

ACTIVE ECOMMERCE CMS
LARAVEL VERSION OF ACTIVE SUPER SHOP

# Documentation

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

# Documentation

1. What are the prerequisites?
2. How to open the project on 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 run IOS on a simulator?
8. How to activate an app?
9. How to config the Stipe payment gateway?
10. How to generate app store uploadable files?
11. How to Update for Android?
12. How to configure social login?
13. How to configure push notification?
14. How to configure google maps ?
15. How to configure the default language for mobile apps?
16. How to configure multiple languages for mobile apps?
17. My data is not changing. How to clear cache?

## 1. What are the prerequisites?

**Answer:**

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 on your PC/mac.

## 2. How to open the project on Android Studio?

**Answer:**

**Extract** the **scource_code.zip**. You will find this inside the **main zip**.

a. **Open** the folder in your **Android studio**.
b. Even if you are building an app for iOS, use Android Studio for the build.
c. 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?

**Answer:**

**a. App Config:**

**-** This helps you connect your app to your server.

**Open lib/app_config.dart**

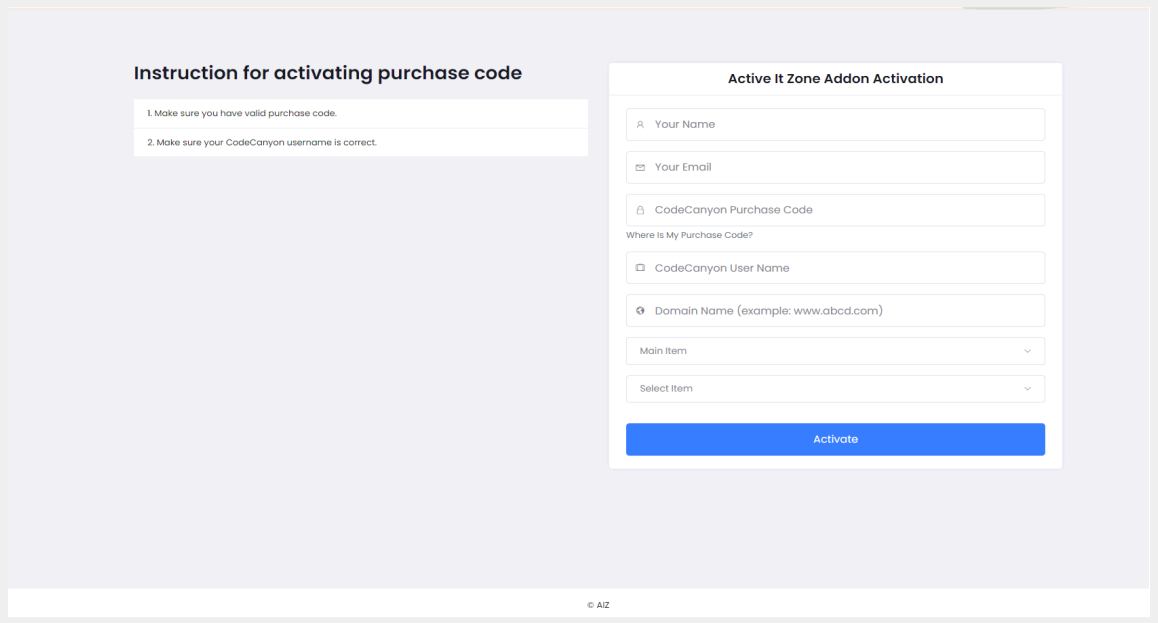
- You can **change** the **copyright_text, app_name,purchase_code,, HTTPS,DOMAIN_PATH** variable.

- Do not change the other variables.

- Make sure that **system_key** and **purchase_code** are given. Otherwise, your app will not work properly.

**b. Activating the Flutter app**:

First, **activate** your Flutter app from here: https://activeitzone.com/activation/addon.



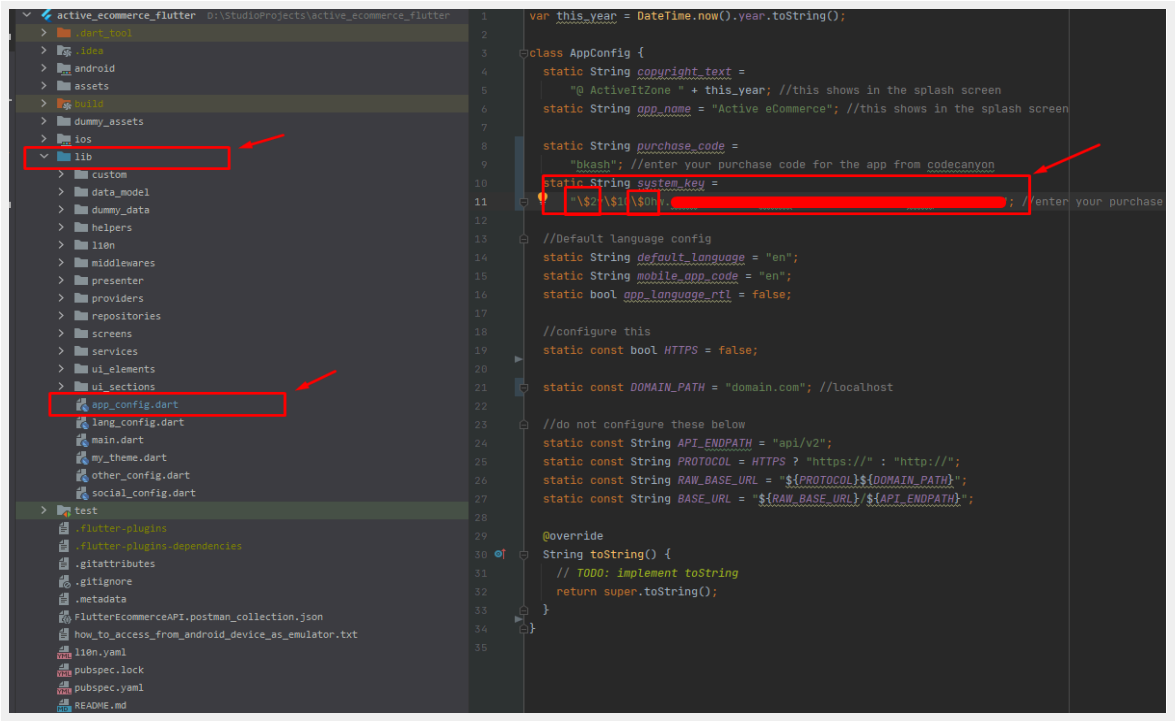

**c. Procedure for getting the system key**:

i. Please make sure that your **Active e-commerce CMS** system is **activated** from this site https://activeitzone.com/activation and that you have collected a **System key**. For more information please check the ecommerce CMS system **documentation**.

ii. Now you can follow the manual process to **set the system key**.

a. Use the system key in your **app_config.dart file**. Find the **app_config.dart** in your **project->lib** folder**.**

Here is the screenshot for a better understanding.



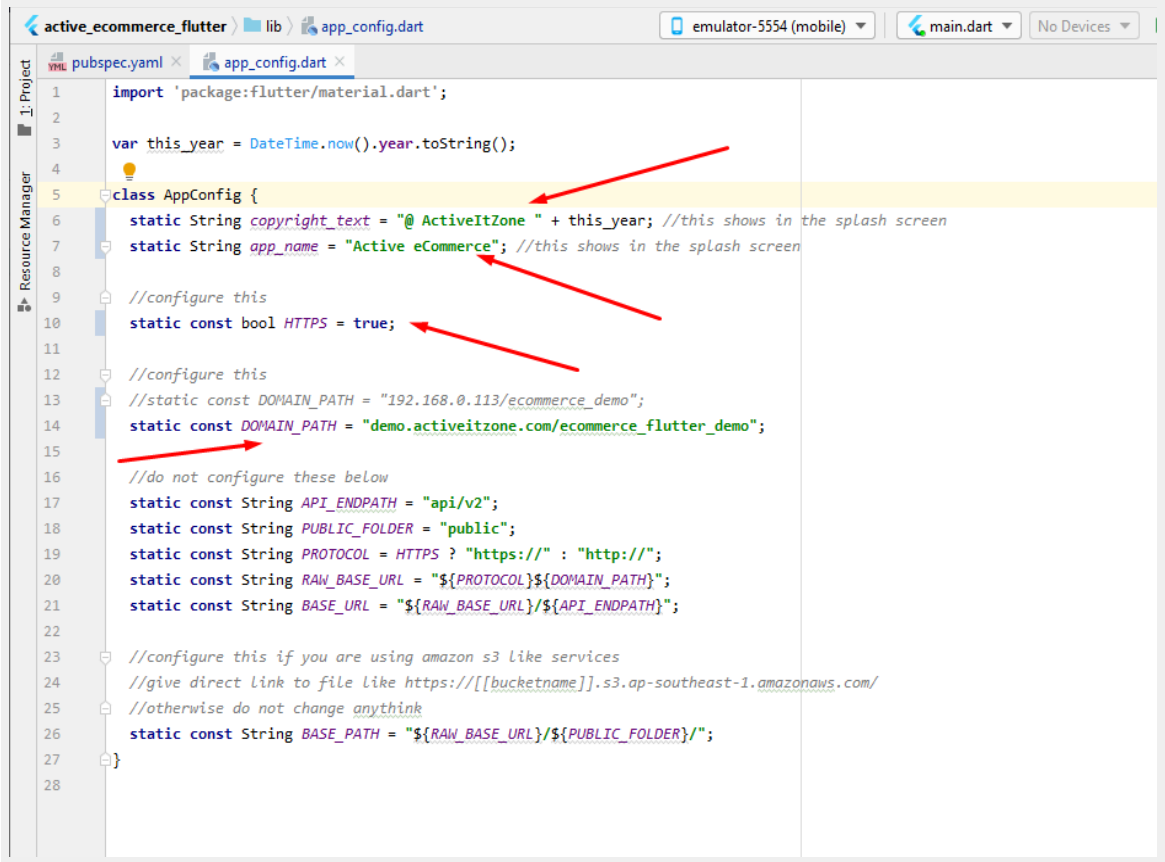

****Make sure that your system uses the same system key otherwise it will not work, check your .env file****


If your site does not have https or you 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 work **




### d. 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:**

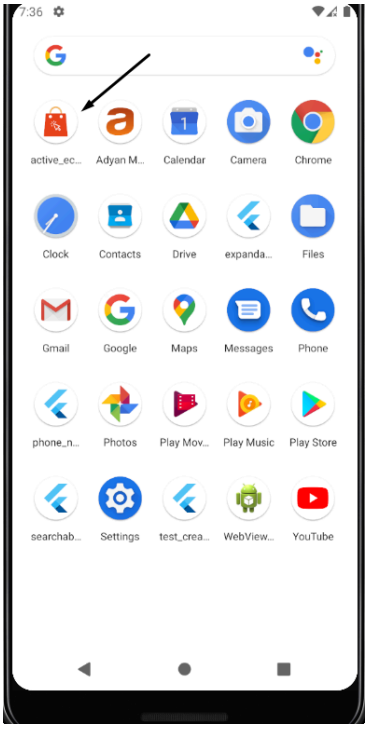
```dart
import 'package:flutter/material.dart';

class MyTheme{
  /*configurable colors stars*/
  static Color accent_color = Color.fromRGBO(230,46,4, 1);
  static Color soft_accent_color = Color.fromRGBO(247,189,168, 1);
  static Color splash_screen_color = Color.fromRGBO(230,46,4, 1); // if not sure , use the same color as accent color
  /*configurable colors ends*/


  /*If you are not a developer, do not change the bottom colors*/
  static Color white = Color.fromRGBO(255,255,255, 1);
  static Color light_grey = Color.fromRGBO(239,239,239, 1);
  static Color dark_grey = Color.fromRGBO(112,112,112, 1);
  static Color medium_grey = Color.fromRGBO(132,132,132, 1);
  static Color grey_153 = Color.fromRGBO(153,153,153, 1);
  static Color font_grey = Color.fromRGBO(73,73,73, 1);
  static Color textfield_grey = Color.fromRGBO(209,209,209, 1);
  static Color golden = Color.fromRGBO(248, 181, 91, 1);
  static Color shimmer_base = Colors.grey.shade50;
  static Color shimmer_highlighted = Colors.grey.shade200;

  //testing shimmer
  /*static Color shimmer_base = Colors.redAccent;
  static Color shimmer_highlighted = Colors.yellow;*/



}
```

Red Green Blue Opacity

**e. 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:

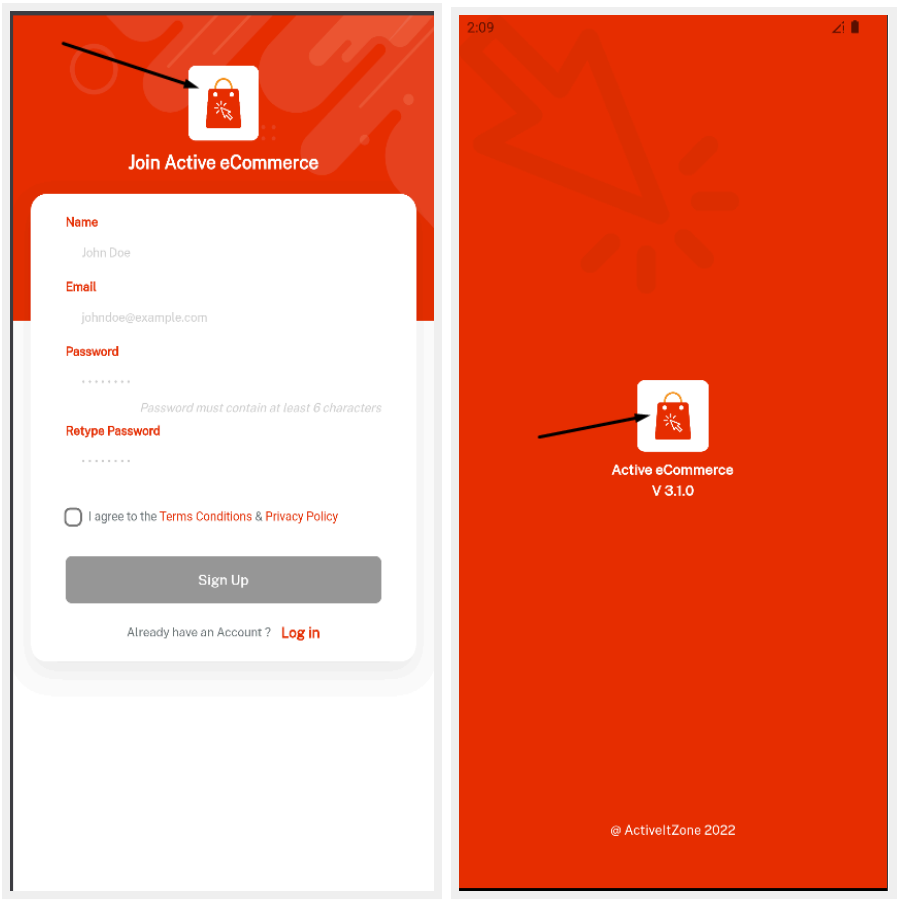
**flutter pub run flutter_launcher_icons:main**

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

f. **Configure other logos:**

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

branding.




This logo will be found in:

assets/login_registration_form_logo.png (512x512)

assets/splash_screen_logo.png (512x512)

Change this logo with your own logo.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?

### Answer:

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

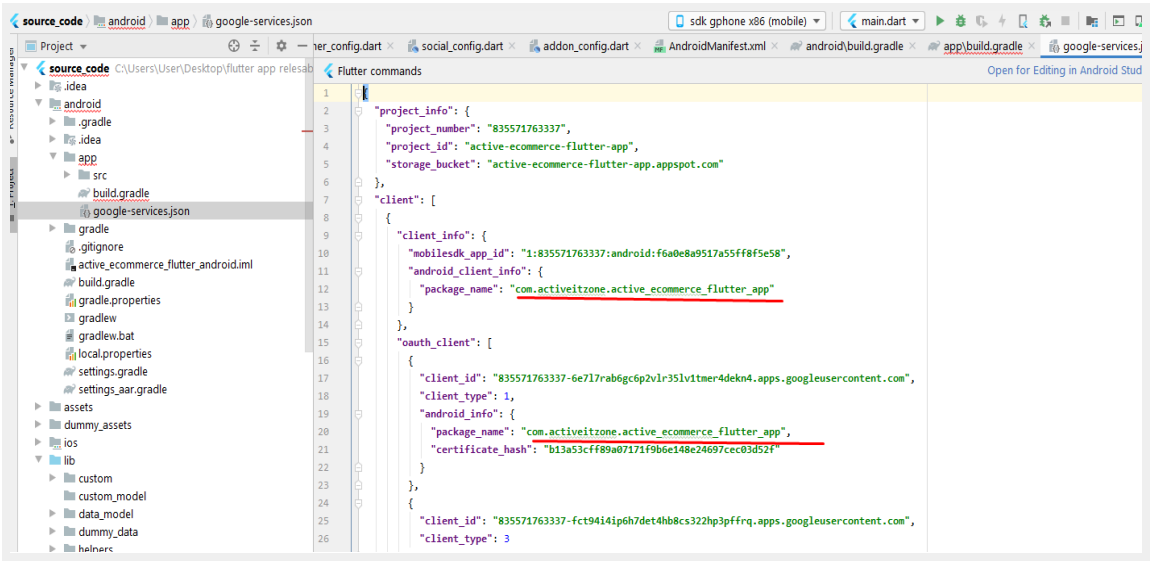
**In Android**

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

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

**However in one place you must have to manually change your package name.

Open android/app/google-services.json , and change your package name manually. Otherwise you will get a build error , even on emulator on debug build




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)?

**Answer:**

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?

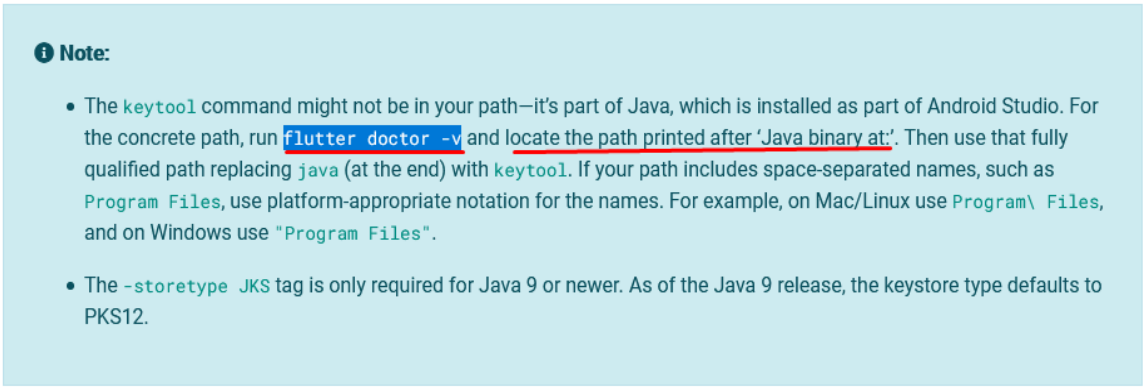
**Answer:**

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

Signing the app:

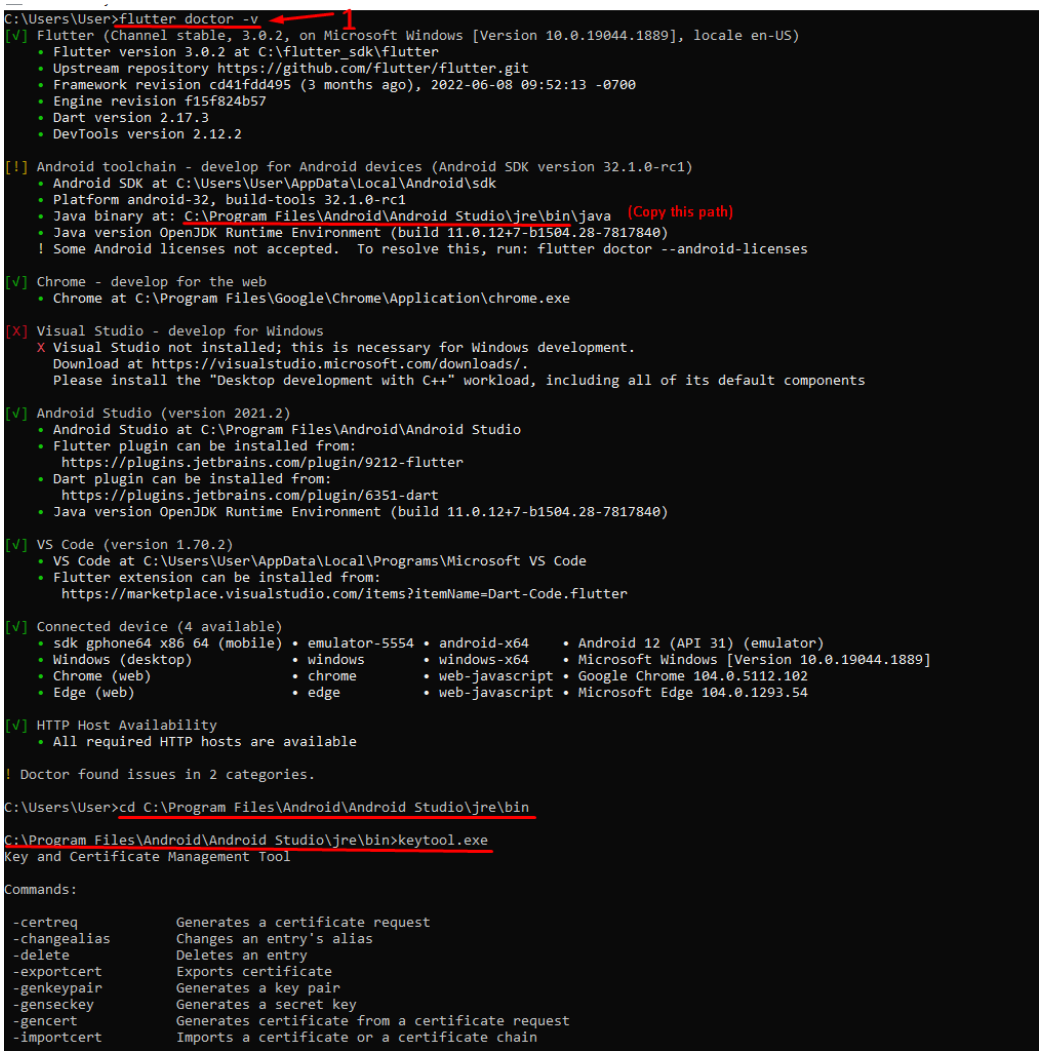
To publish on the Play Store, you need to give your app a digital signature. Use the following instructions to sign your app.

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




Setup environment for keytool:

a. Run this command in your system terminal: flutter doctor -v
b. Find the Java binary at: copy this path except the "java" folder.
c. Change your terminal path by running this command : cd "past here your copied path" then press enter. Example: **cd C:\Program Files\Android\Android Studio\jre\bin**
d. Run the keytool.exe file. Example: **C:\Program Files\Android\Android Studio\jre\bin>keytool.exe**

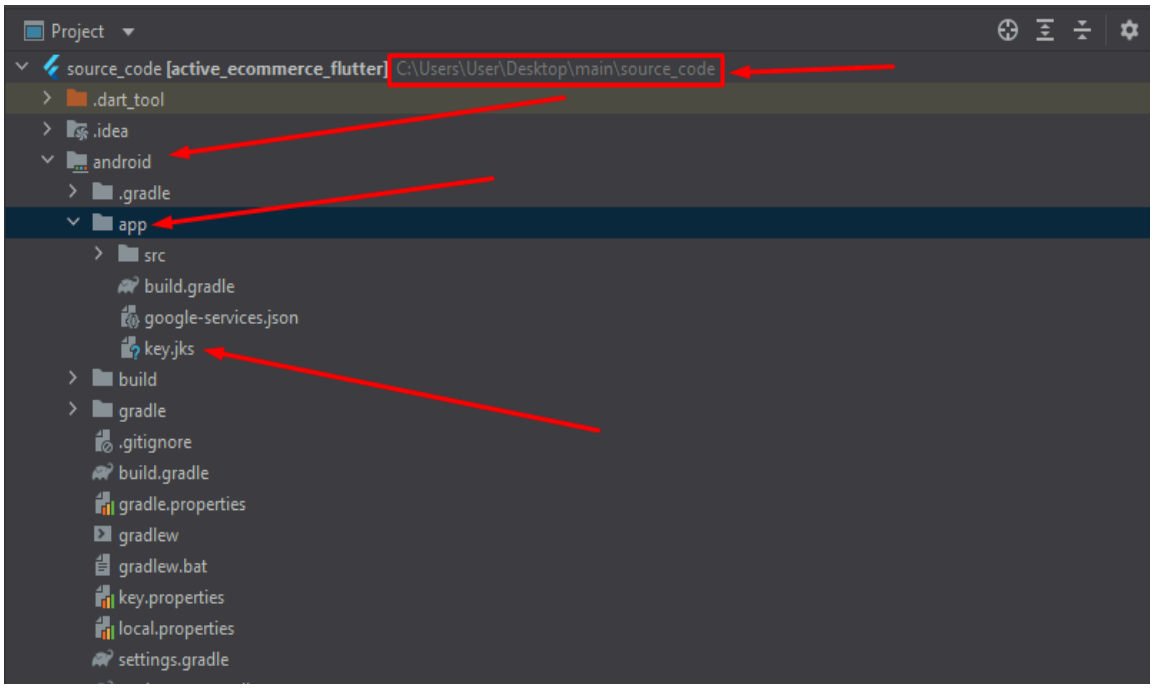



Create jks file for uploading app on playstore:

a. Open your project path and run this command: **keytool -genkey -v -keystore android/app/key.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 100000 -alias key**
b. Provide all information.

```
PS C:\Users\User\Desktop\main\source_code> keytool -genkey -v -keystore android/app/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]:  TS
Is CN=Test, OU=Test, O=Test, L=Test, ST=Test, C=TS correct?
  [no]:  y

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=TS
Enter key password for <key>
```

Your created JKS file is located your project->android->app folder




## 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****

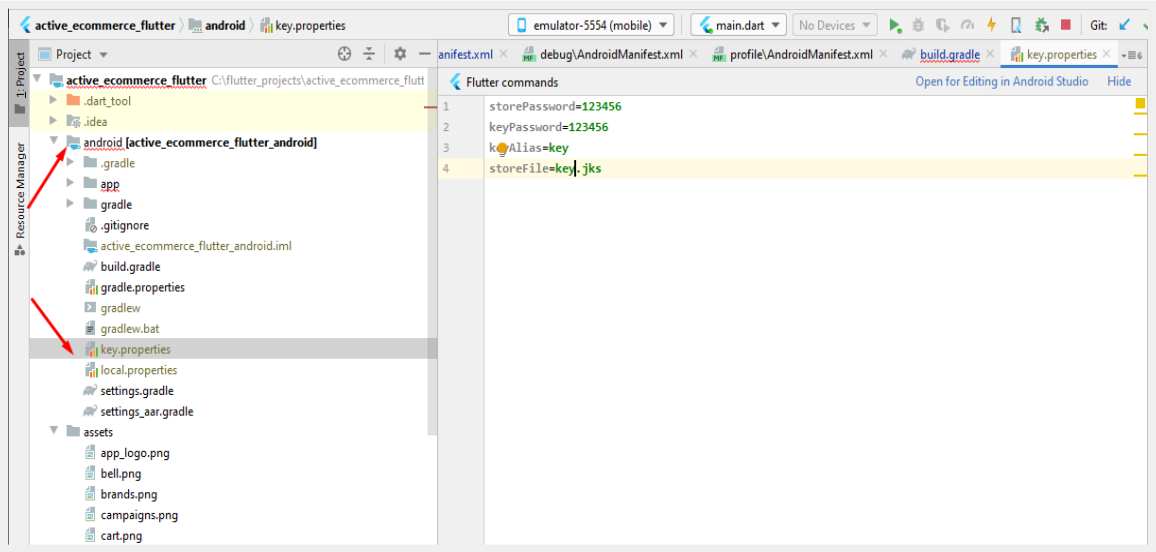
Enter the information in the file key.properties in the android folder.

a. storePassword.

b. keyPassword.

c. keyAlise.

d. storeFile



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
    }
}
```

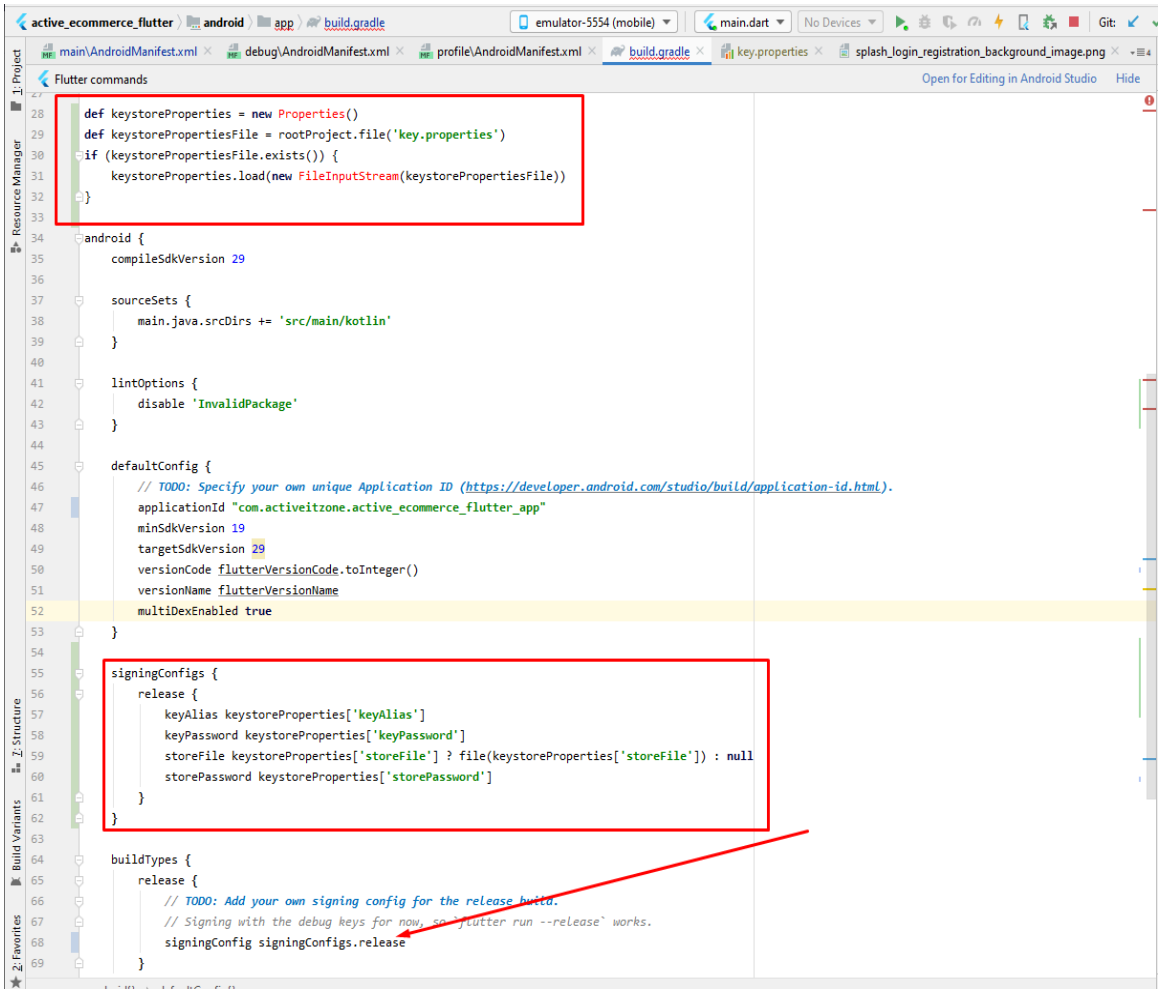
With the signing configuration info:

```
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

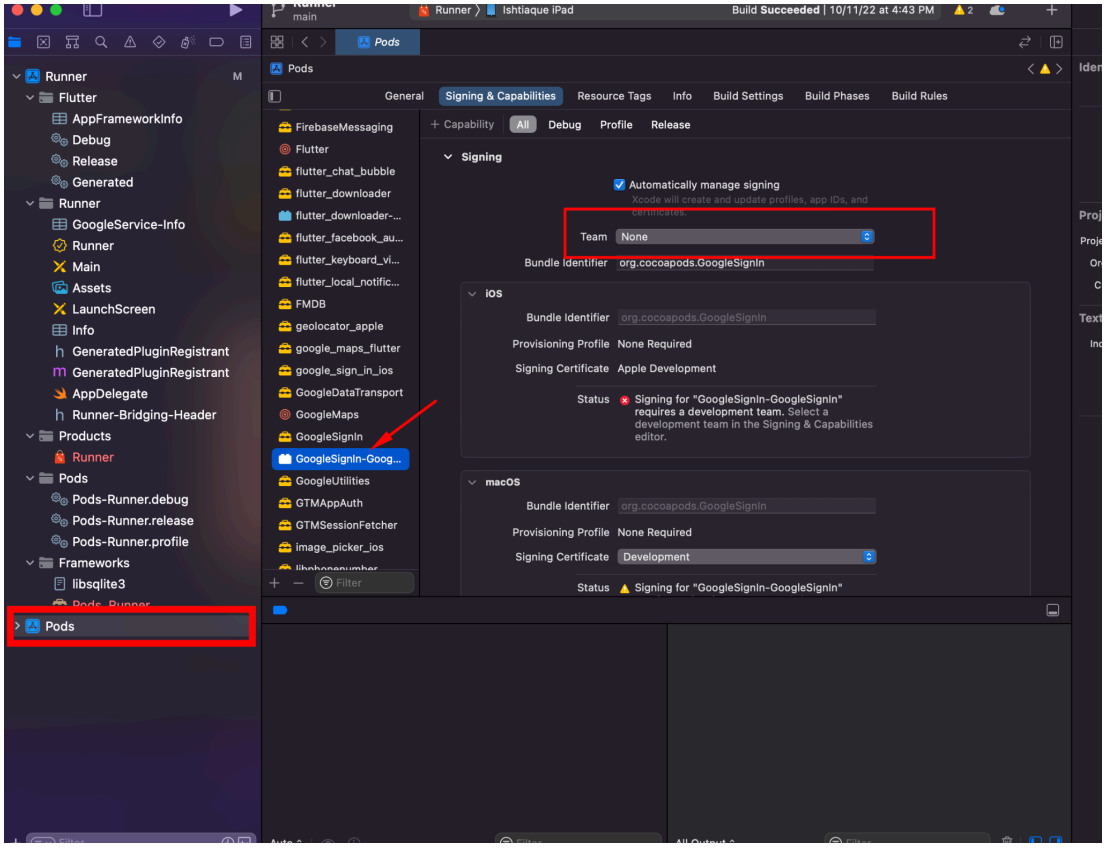
## 7. How to run IOS? **Read all the points carefully before doing anything

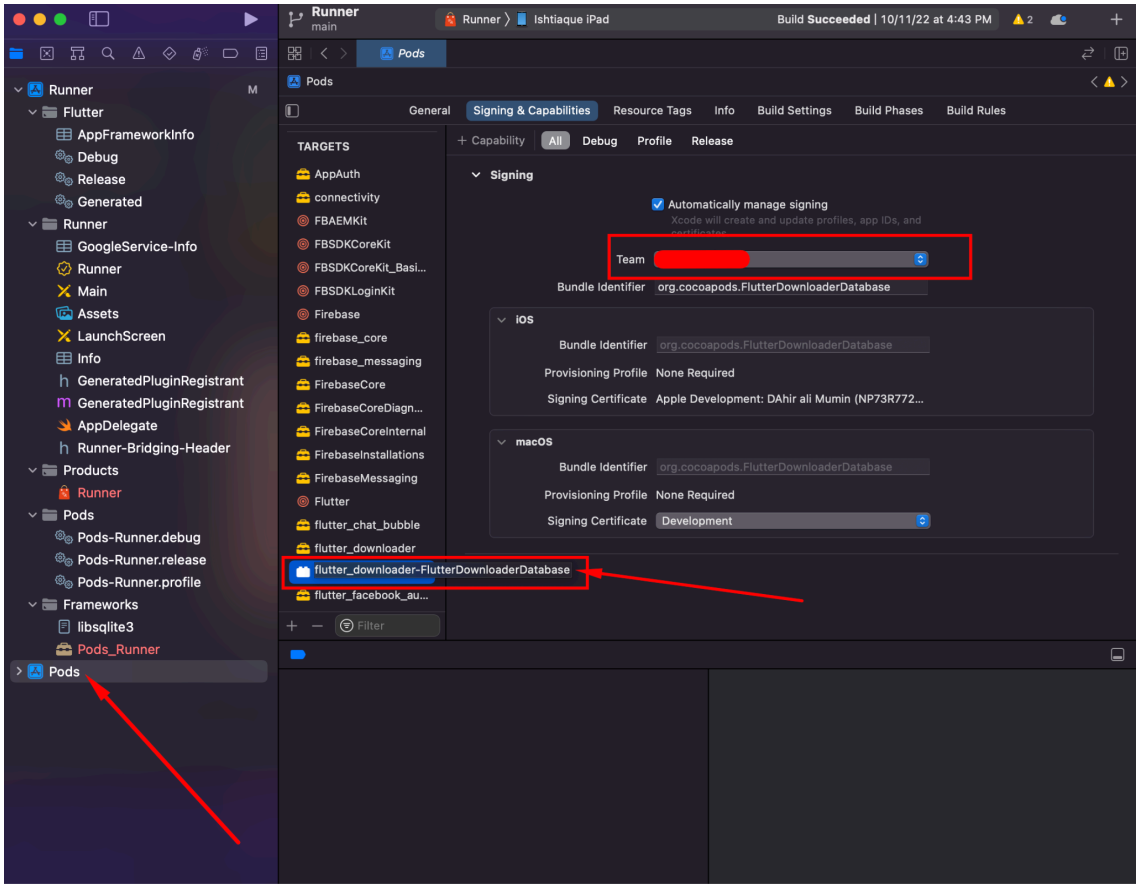
**Answer:**

a. Install Xcode on your mac.

b. Open your ios folder in Xcode and select your developer team from Signin and capabilities.

c. Choose your team from GoogleSignin-Googlesignin and flutter_downloader-FlutterDownloaderDatabase from Pod->GoogleSignin-Googlesignin and Pod->flutter_downloader-FlutterDownloaderDatabase

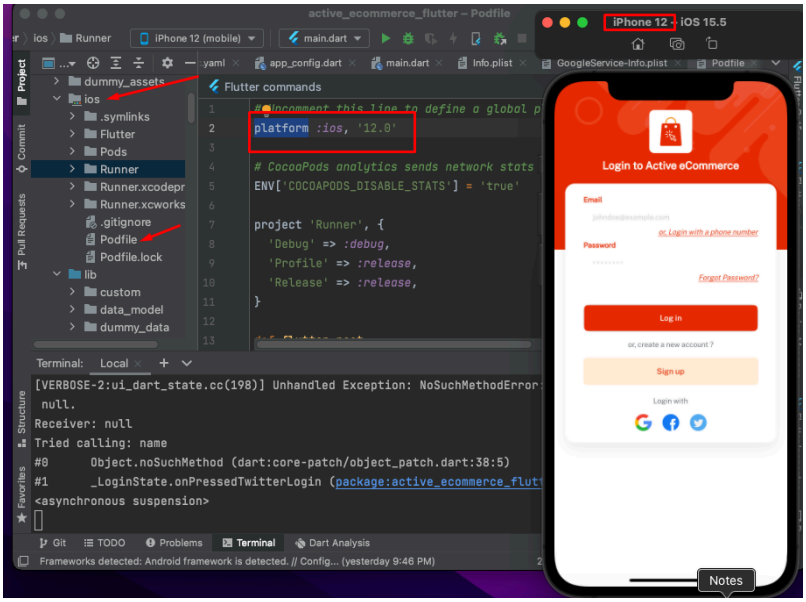





d. You need to open a simulator. Run this command for open a simulator open -a simulator

e. Change the platform version according to your simulator version.

   i) Goto your source_code->ios->Podfile add your ios simulator version.




f. Run this command fluter run.

More info https://docs.flutter.dev/get-started/install/macos

## 8. How to activate an app?

**Answer:**

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

a. Go to this link https://activeitzone.com/activation/addon

b. Put your name in the name field.

c. Put your email in the email field.

d. Put your purchase code in the purchase field.

e. Put your codecanyon user name in the codecanyon user name field.

f. Put your domain name in the domain field.

g. Select **Active Ecommerce** option from main item.

h. Select the **eCommerce Flutter App** option from the item.

i. Click the **Activate** button After filling all the fields and activate your app.

9. **How to config the Stipe payment gateway?**

**Answer:**

Please follow the below procedure:

Just put www before your domain name

Also make sure your server is configured with www

10. **How to generate app store uploadable files?**

**Answer:**

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 create.
- 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.

11. **How to update for Android? **Read all the points carefully before doing anything**

**Answer:**

a. 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.

b. If you are installing and building the release file for the first time this section is not for you.

c. Extract the scource_code.zip. You will find this inside the main zip.

d. Open the folder in your Android studio.

e. **Remember to open this in a separate folder than your old project.

f. Even if you are building an app for iOS, use Android Studio for the build.

g. Then in your Android studio terminal run:

flutter pub get

h. This will fetch all the necessary packages

i. If you are updating, you must have built the key.jks previously

j. Copy the key.jks , key.properties, and the manifest file from your old project and paste in the correct locations

k. See the previous screenshots for the file locations

l. If you are missing your old project, you have to configure key.properties, and the manifest file like described in the installation.

m. 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.

n. But do not create a new key.jks, you have to update your app with the existing key

o. 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.

p. In your terminal run: flutter build appbundle

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

r. Upload this app.aab file to your Google Play console.

## 12. How to configure social login?

**Answer:**

1. Create a firebase project.
    A) First you need to have a google account and login in your browser.
    B) Goto this link: https://console.firebase.google.com/
    C) Click the “Add Project” Button.
    D) Add your project name and continue.
    E) Off the “disable Google Analytics” and Click the “Continue” Button.
    F) Click the “Continue” Button.

**Add Android app on your Firebase project**

1. Click the Android icon.

**Add some information:**

A) Android package name

B) App nickname

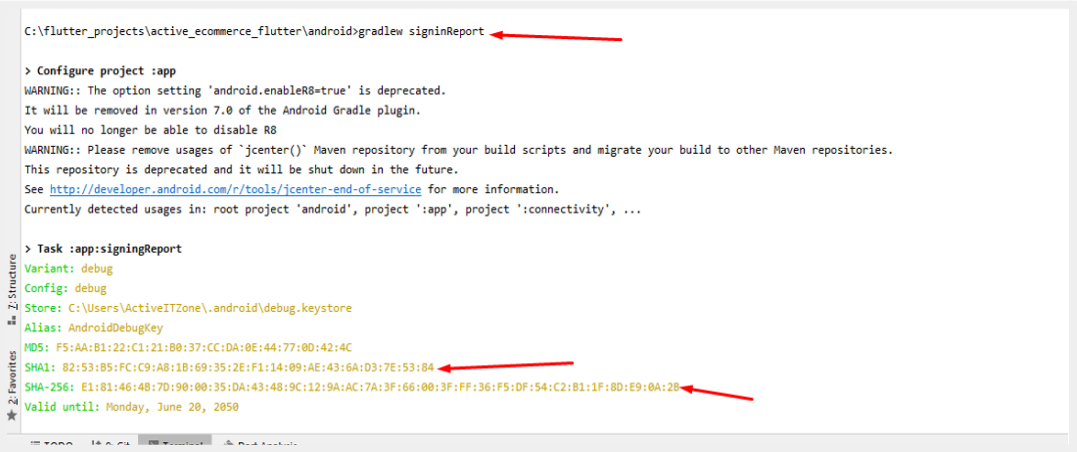
C) Add signing certificate SHA-1 and SHA-256

**How to get SHA-1 code in your project ?**

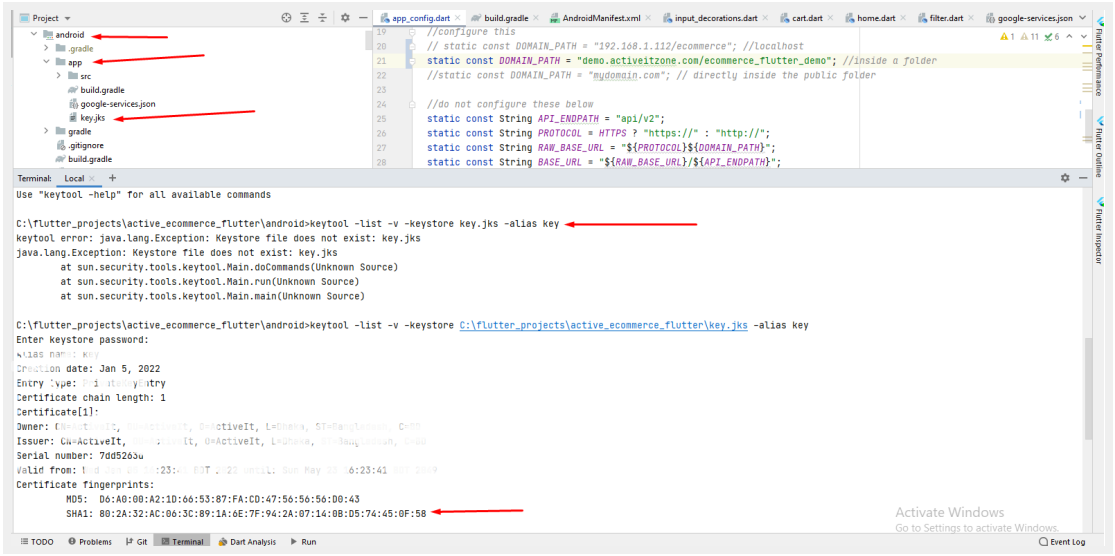
A. Goto android studio and open your project then open terminal and ensure that it shows your project directory in the terminal.

B. Then drive into the android folder(cd android).

C. i) For debug mode: write this command on your Android studio terminal "gradlew signinReport" then press enter.




ii) For release mode: write this command on your android studio terminal "keytool -list -v -keystore YOUR_PROJECT_DIRECTORY\key.jks -alias key" then press enter.



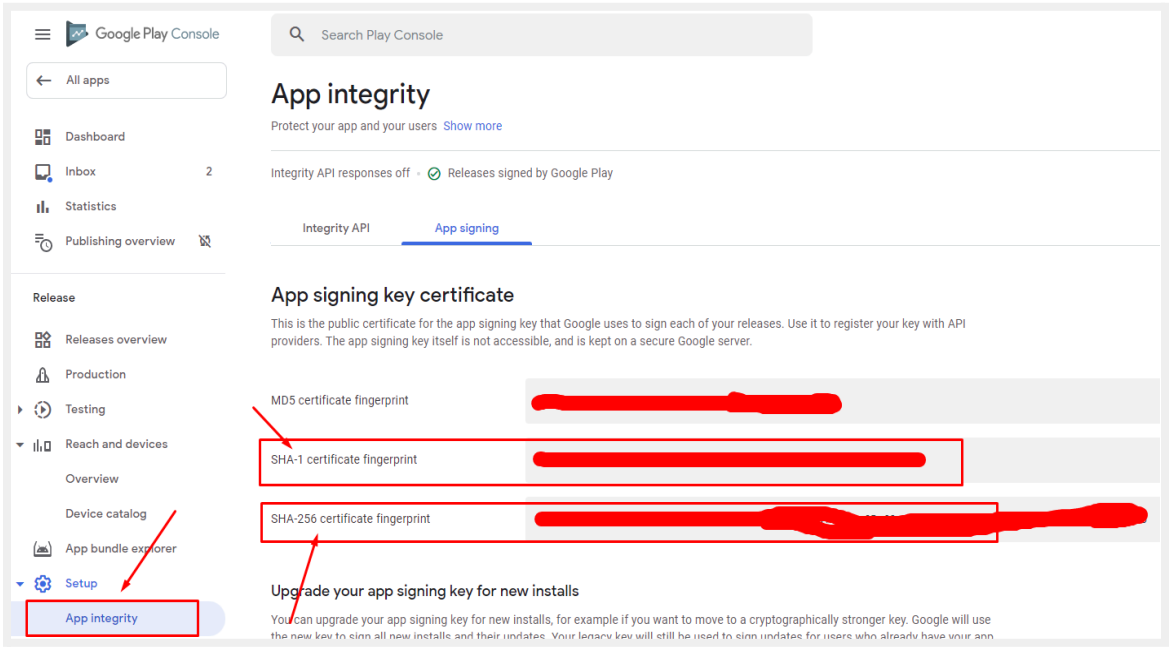

2. Click the “Register App” button

3. Download config file and add this file into your project->android->app folder

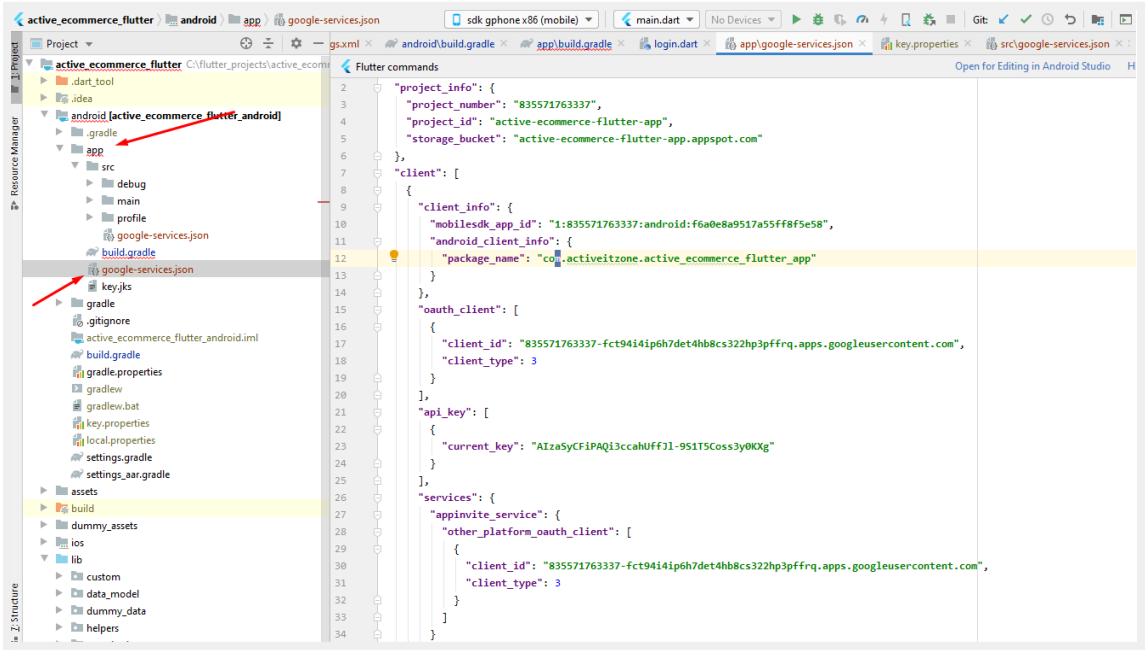
4. Add firebase SDK

5. Click the “Continue to console” button

6. After upload your app on playstore you need to add signing certificate SHA-1 and SHA-256 in firebase project setting : Open your google play console->Your App->Setup->App integrity




You must generate your own google-services.json. Do not use ours - it will not work for you.




**Add the IOS app on your Firebase project**

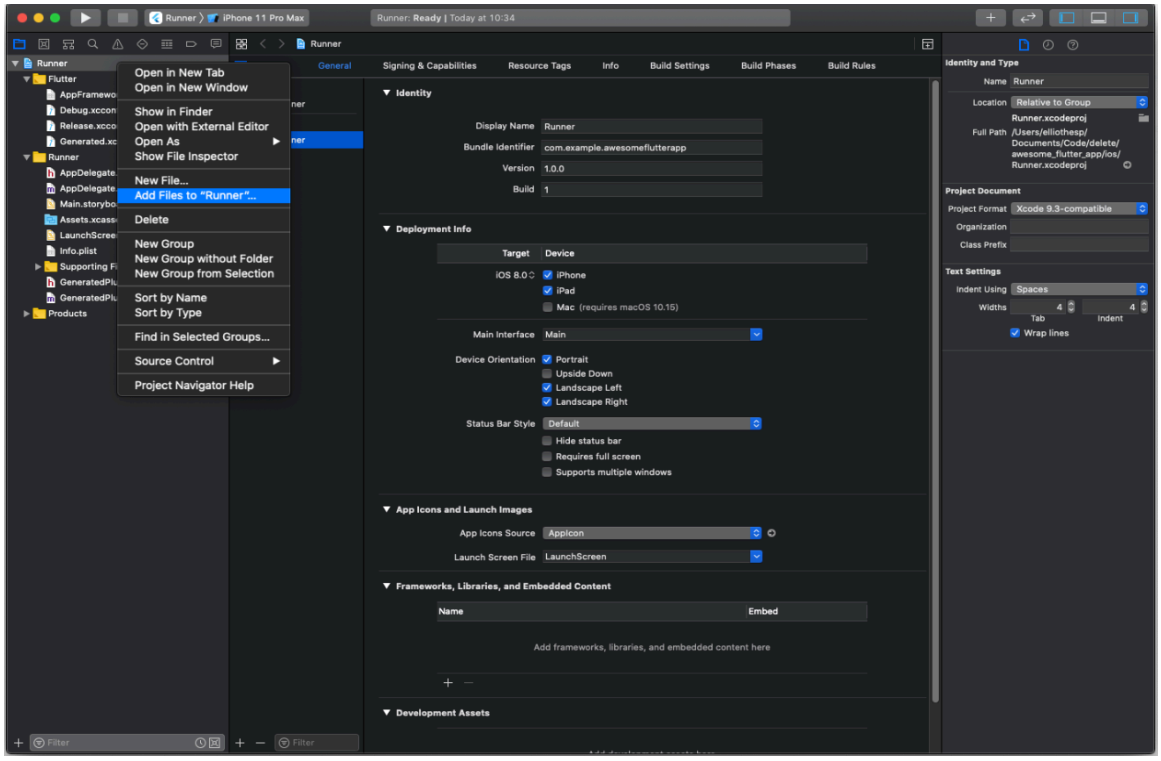
1. Goto your Firebase project setting and click **add app** button.
2. Click IOS icon
3. Add some information.

   i) Apple bundle ID: Insert your app bundle ID.

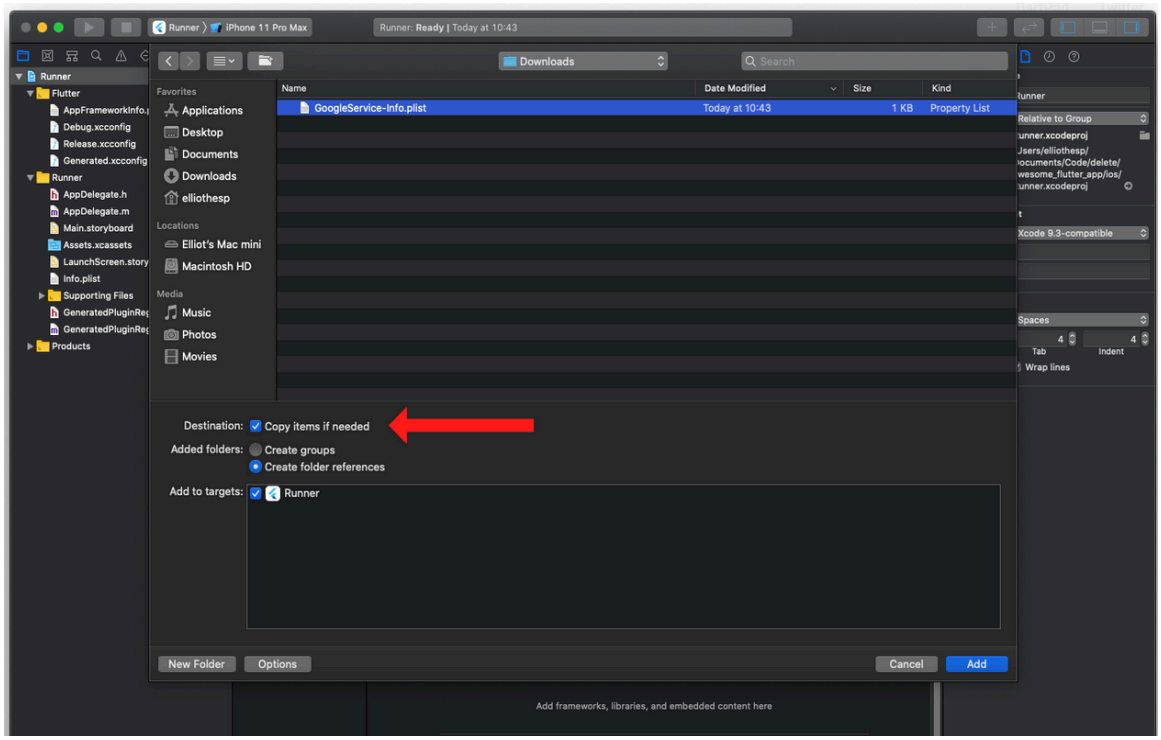
   ii) App nickname (optional).

   iii) App Store ID (optional).
4. Click the Register app button.
5. Download GoogleService-Info.plist file

Next you must add the file to the project using Xcode (adding manually via the filesystem won't link the file to the project). Using Xcode, open the project's ios/{projectName}.xcworkspace file. Right click Runner from the left-hand side project navigation within Xcode and select "Add files", as seen below:



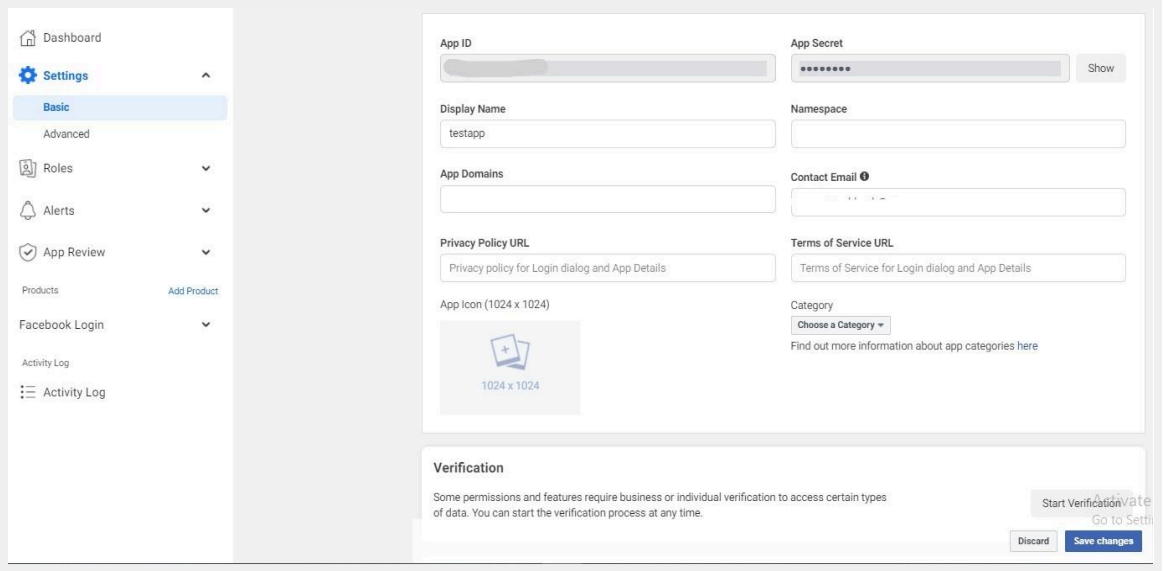

Select the GoogleService-Info.plist file you downloaded, and ensure the "Copy items if needed" checkbox is enabled:



**Firebase Authentication :**

1. Goto signin method and then enable Google, facebook and twitter and apple.
2. For facebook you need a facebook app id and app secret, we will create the next step.

   Add these documents.




**I) Facebook**: Package Used

https://pub.dev/packages/flutter_facebook_auth

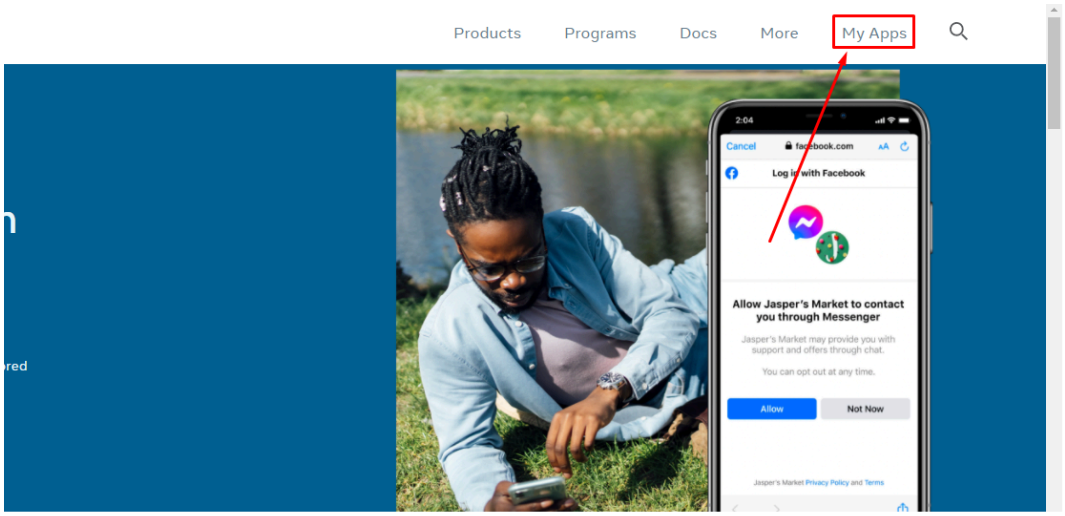
See its documentation and steps

How to create a Facebook app?

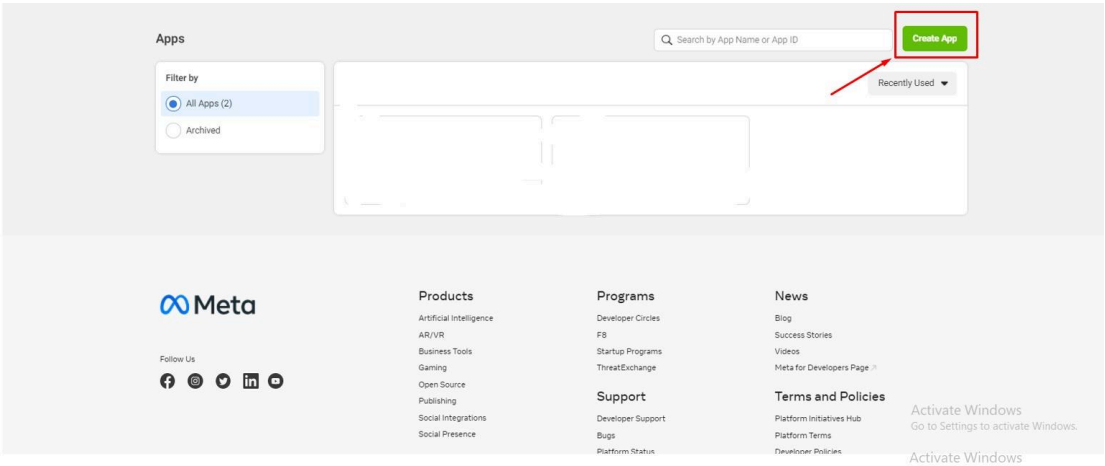
A) First you have to log in to facebook in your browser.

B) Goto https://developers.facebook.com/

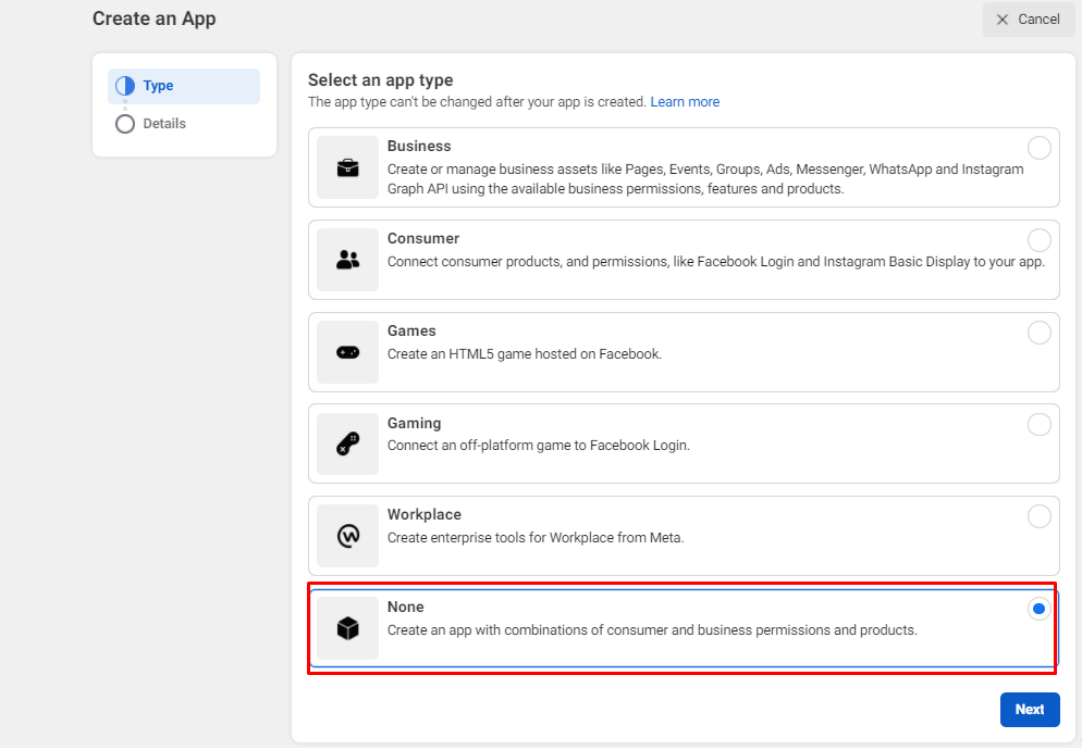
C) Goto My Apps.




D) Create an app.

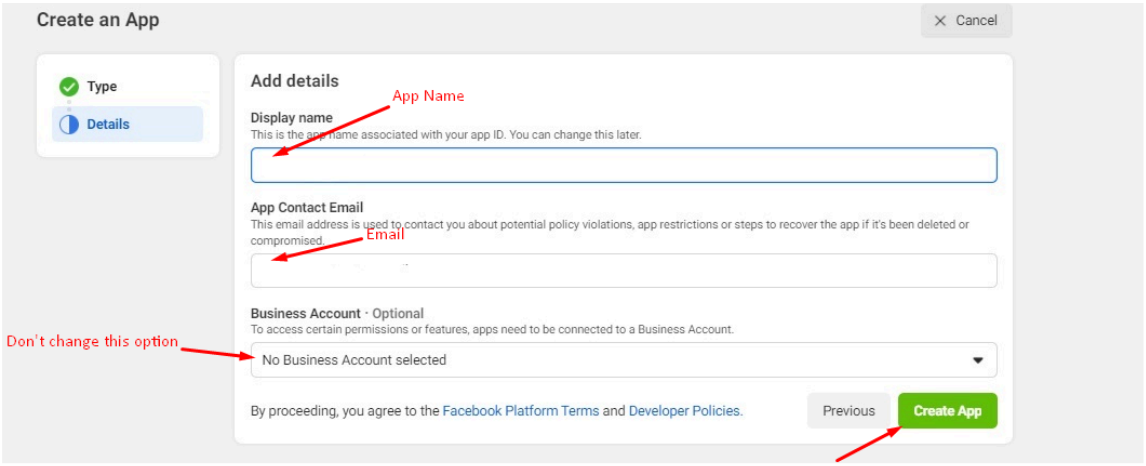



Select App Type None.



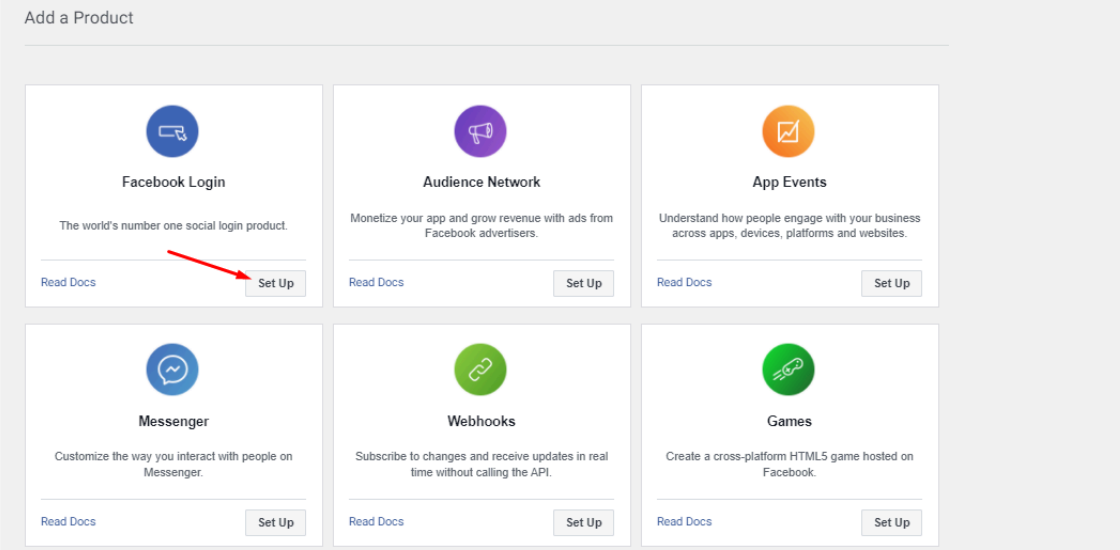

E) Add your app details

- App Name.
- App Contact Email.
- Business Account. (Don't Change it).

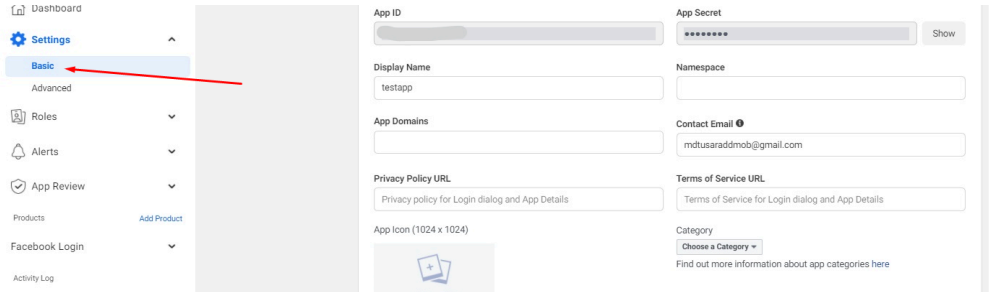



F) Add product "Facebook Login".

- ❖ Click the "Set Up" button.




D) Click the “Basic” Option.



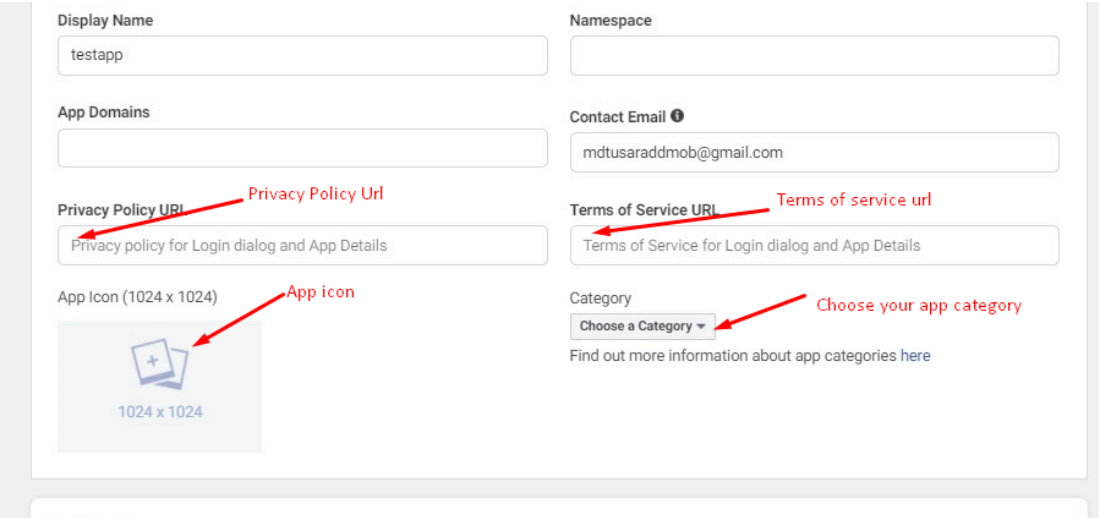

E) Add some information and save it.

I. This app privacy policy url.

II. This app terms of service url.

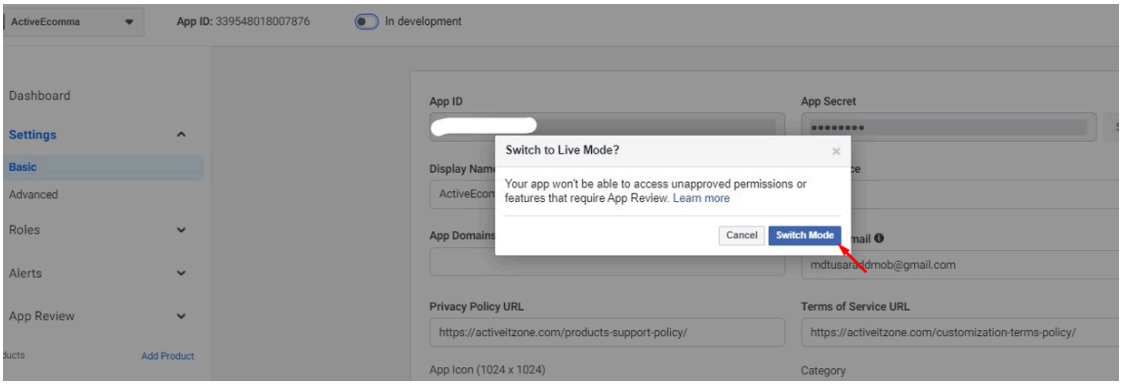
III. This app icon (Icon size also 1024 X 1024 Or 512 X 512 ).

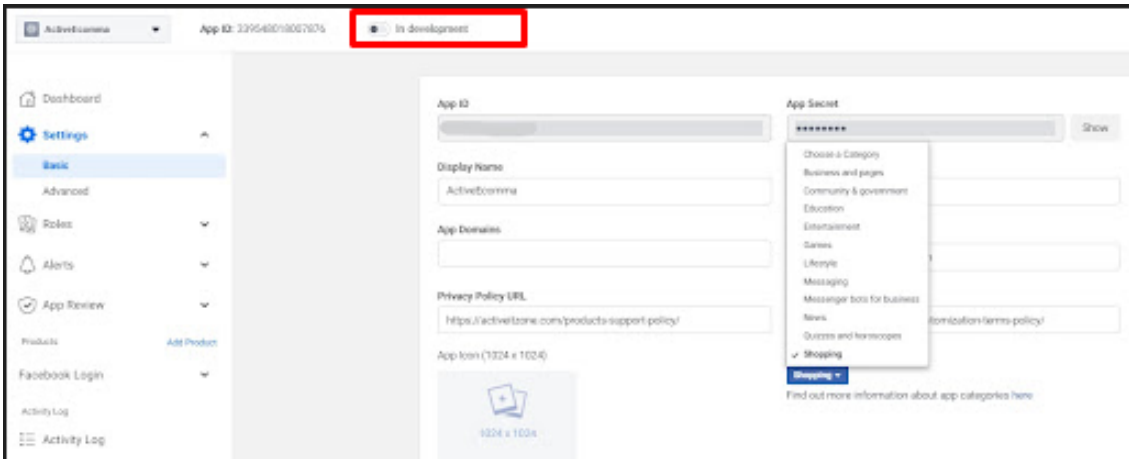
IV. This app category.




J) Now activate your app.

K) Click the “Switch Mode” Button.



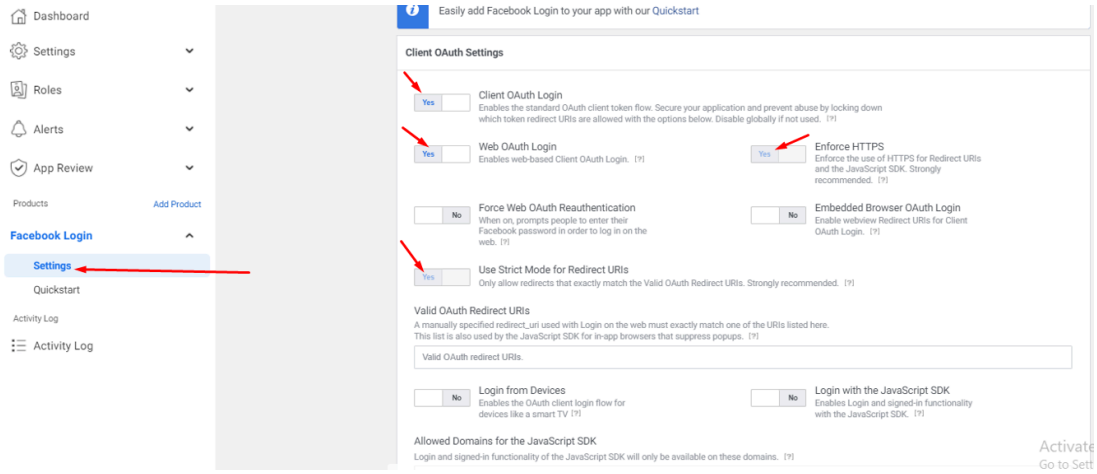

L) Make sure your app facebook login settings are on these options.

A) Client OAuth Login.

B) Web OAuth Login.

C) Enforce HTTPS

D) Use Strict Mode for Redirect URIs




**13.**

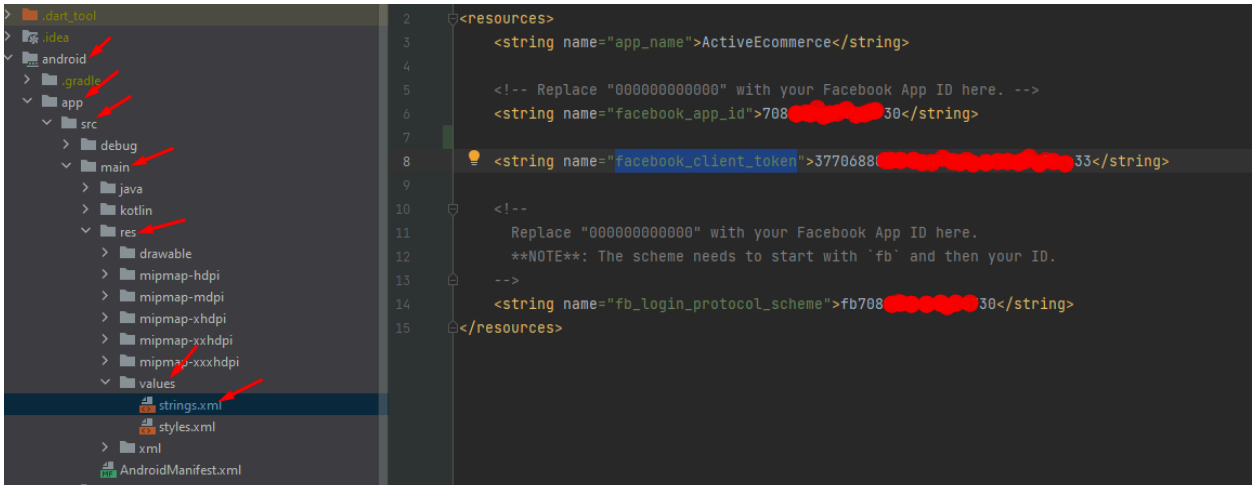
## How to configure social login?

# Integration

### For Android:

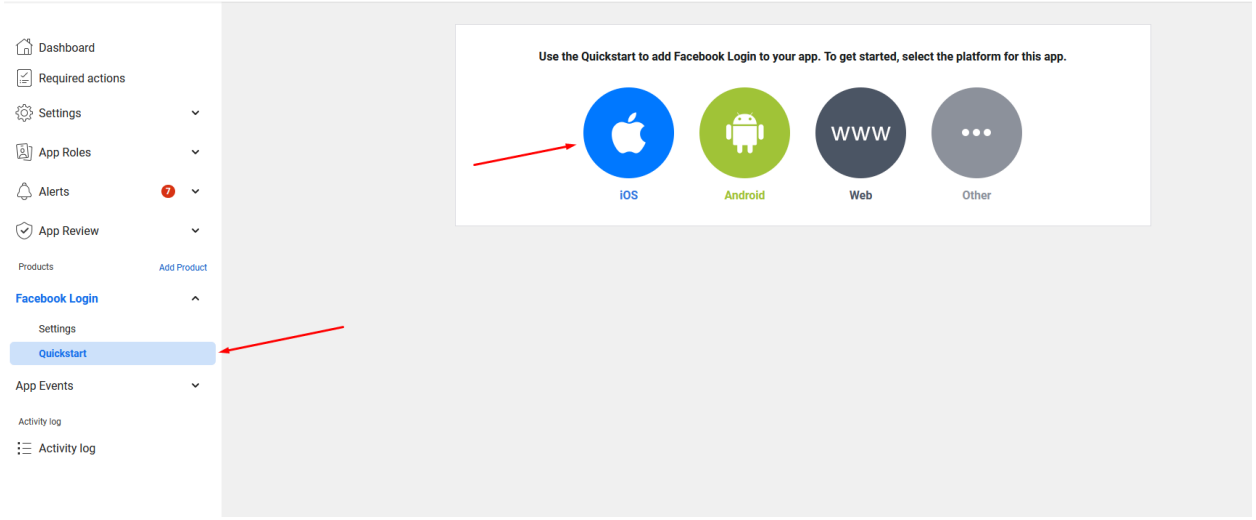
M) Add your facebook app id, app name, `facebook_client_token` and fb login protocol scheme (***NOTE**: The scheme needs to start with `fb` and then your ID.example: fb123456789*)

into your project->android->app->src->main->res->values->string.xml file.

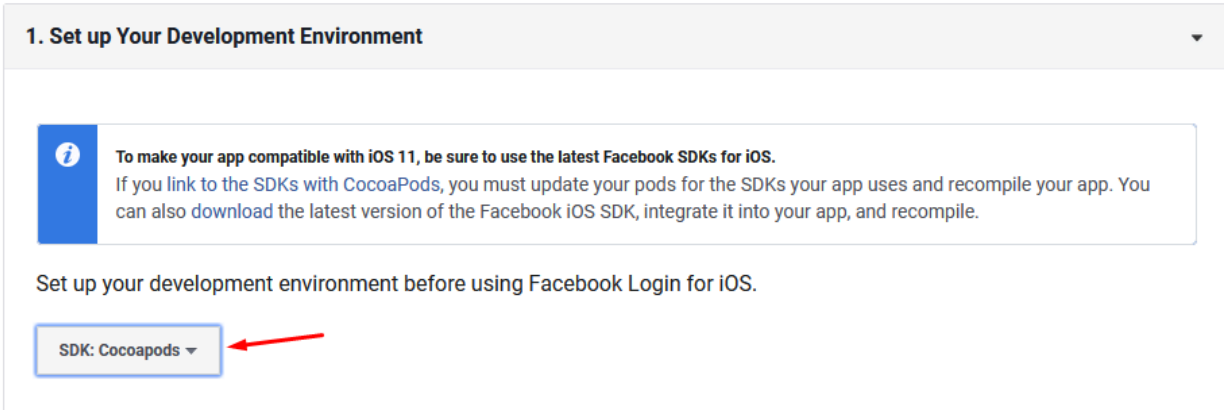



### For IOS:

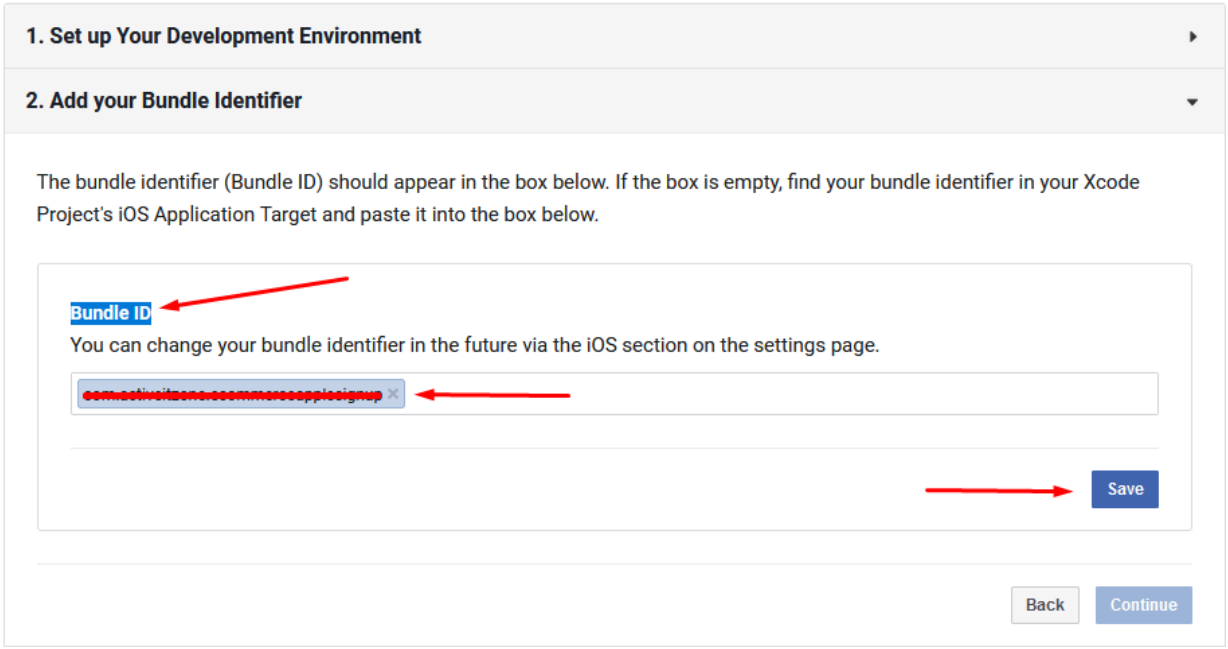
Goto Facebook Login->Quickstart then select iOS platform and just follow 1 and 2 steps.



Then  choose your environment SDK: Cocoapods



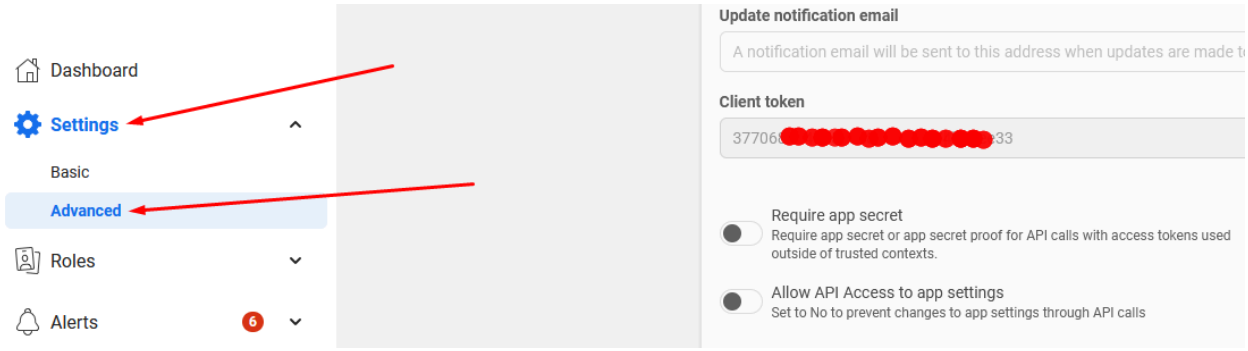

Then add your Bundle ID then save it and exit from the quick start page.




Do not need to follow the others step in the quick start page.

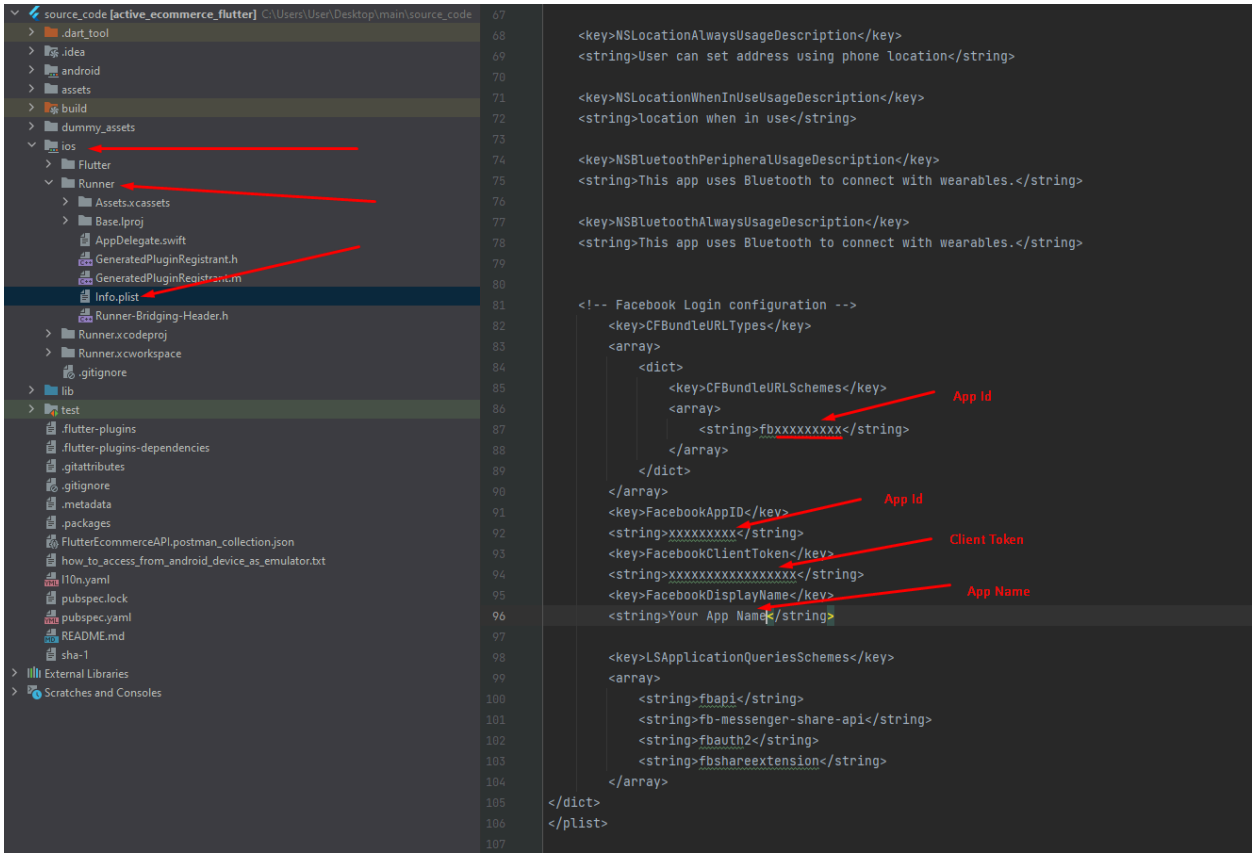
**Client Token:** Where do you get the client token?

Ans: Goto your facebook app setting->advance finde the Client token




**In your project's ios folder:** Add your facebook app id, app name, `facebook_client_token` and fb login protocol scheme (*`**NOTE**: The scheme needs to start with `fb` and then your ID.example: fb123456789`*)

into your project->ios->Runner->info.plist file.



**Google**: Package Used

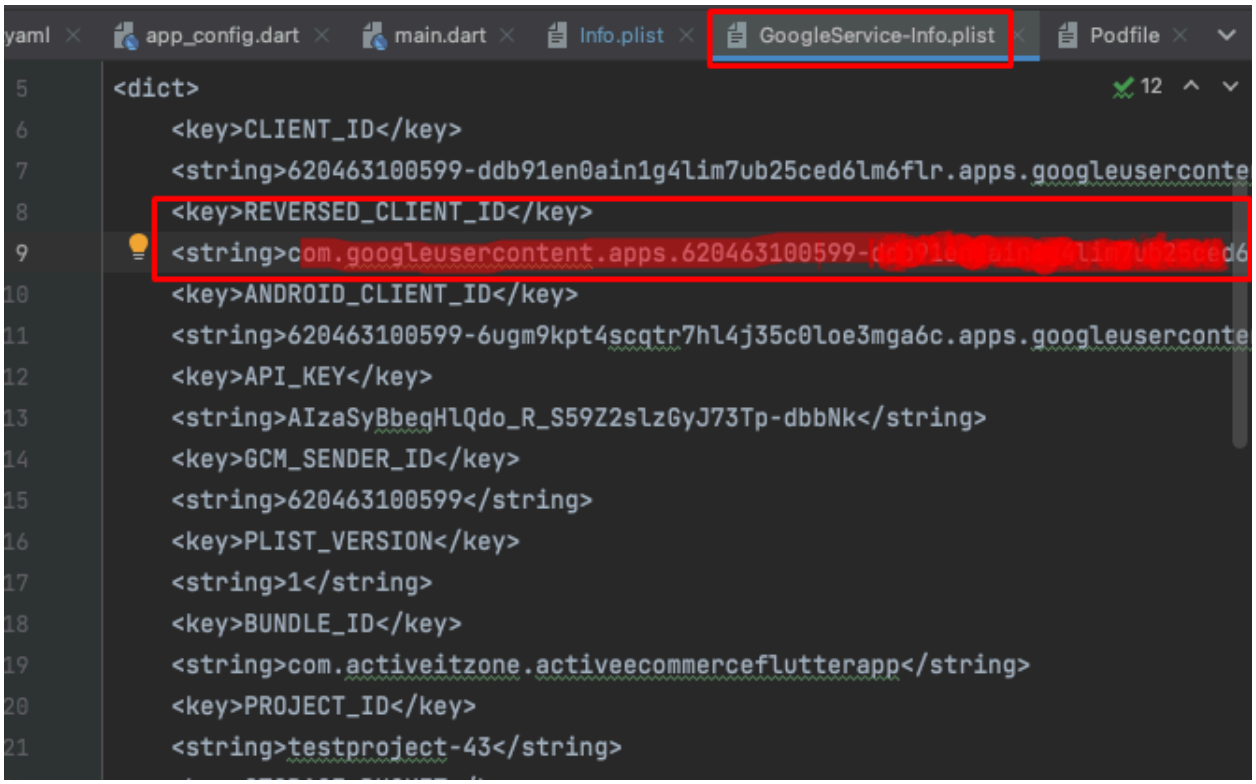
https://pub.dev/packages/google_sign_in

# Integration

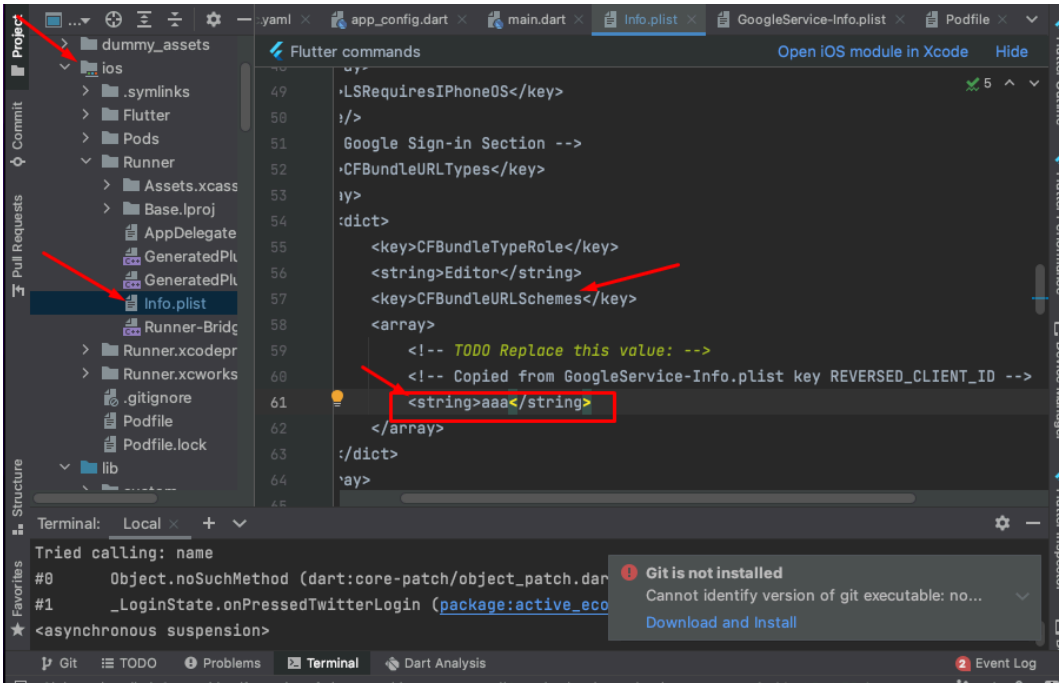
**For IOS:**

1. First register your application.
2. Make sure the file you download in step 1 is named GoogleService-Info.plist.
3. Move or copy GoogleService-Info.plist into the [my_project]/ios/Runner directory.
4. Open Xcode, then right-click on Runner directory and select Add Files to "Runner".
5. Select GoogleService-Info.plist from the file manager.
6. A dialog will show up and ask you to select the targets, select the Runner target.
7. Then add the CFBundleURLTypes attributes below into the [my_project]/ios/Runner/Info.plist file.

**Open your GoogleService-Info.plist file and copy *REVERSED_CLIENT_ID***




Put ***REVERSED_CLIENT_ID*** in your source_code->ios->info.plist files
**CFBundleURLSchemes-> array-> string**




Follow the guideline from here https://pub.dev/packages/google_sign_in

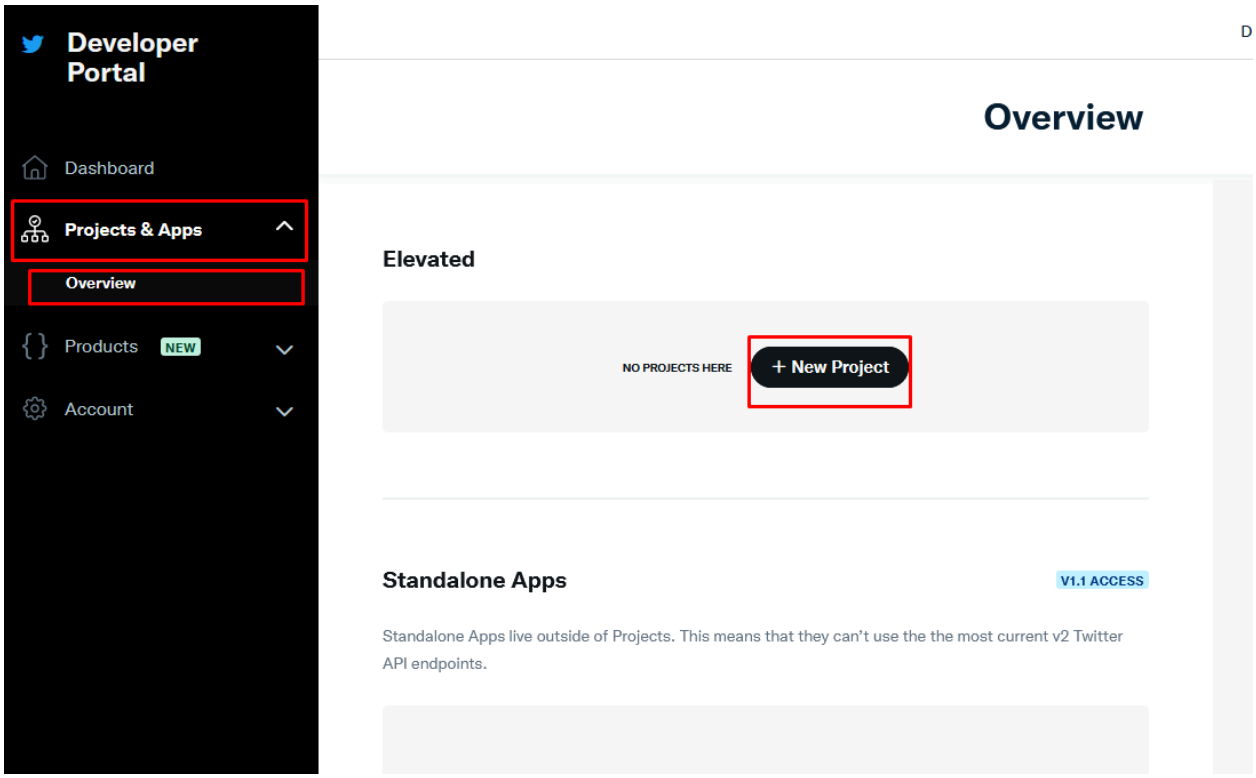
**Twitter** : package used

https://pub.dev/packages/twitter_login

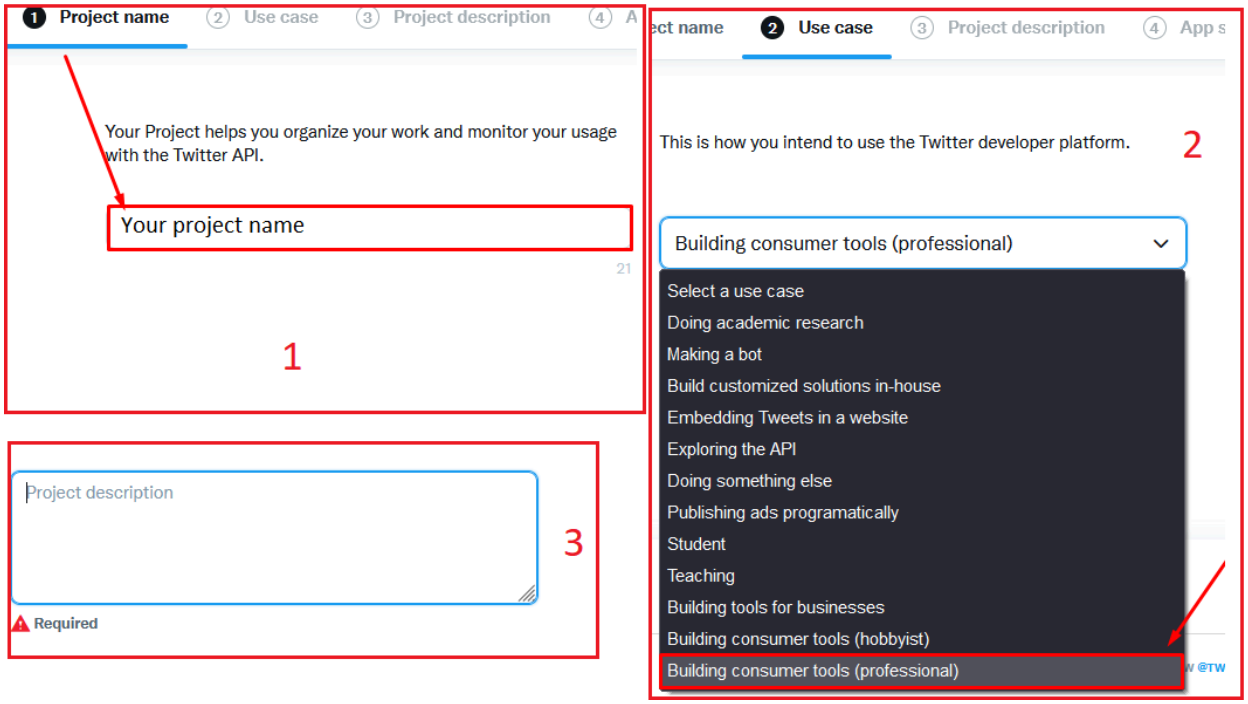
How to create a Twitter app.

1.1. First you have to login twitter in your browser.

1.2. Go to https://developer.twitter.com/
1.3. Go to Developer Portal.
1.4. Go to Project & Apps->Overview create a new project.




1.5. Filup some information for your project. Project Name, Use case, Project description.

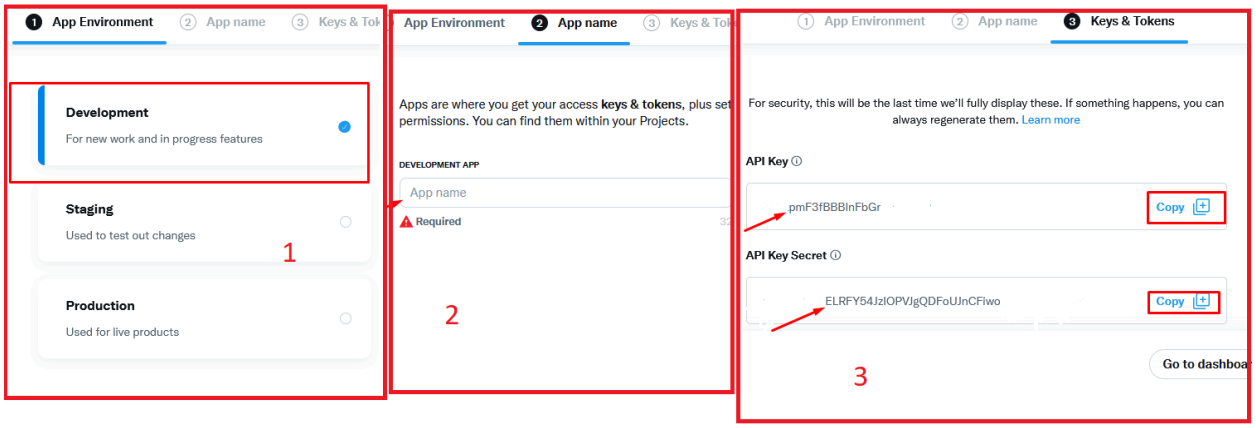



2. Setup your app

2.1. Select your app environment.

2.2. Enter your app name.

2.3. Collect your **API Key** and **API Key Secret.**(you will have to collect it.)




3. Setup user authentication setting.

3.1. Go to Project & Apps-> select your project and then select your created app.

3.2. Find **User authentication settings.**

3.3. Click the **Setup** button.

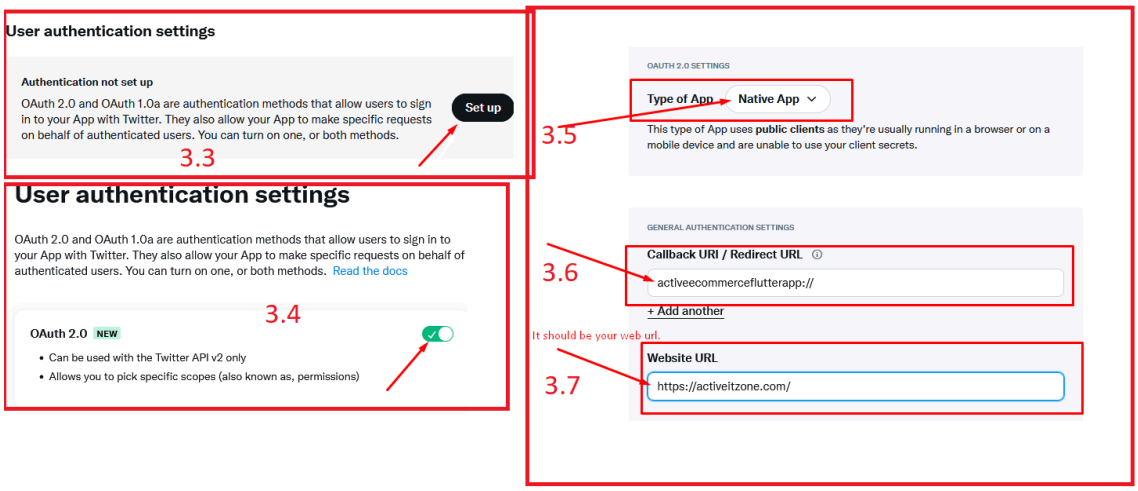
3.4. Enable OAuth 2.0

3.5. Select **Type of App.(Native app)**

3.6. Callback URI.(**`activeecommerceflutterapp://`**).

3.7. Web site URL.(It's your web URL).

3.8. Save it.




4. Setup in your flutter e-commerce code.

4.1. Go to your flutter project->lib->social_config.dart

4.2. Enter your `twitter_consumer_secret` **and** `twitter_consumer_key`.

**Apple SignIn**: package used

https://pub.dev/packages/sign_in_with_apple

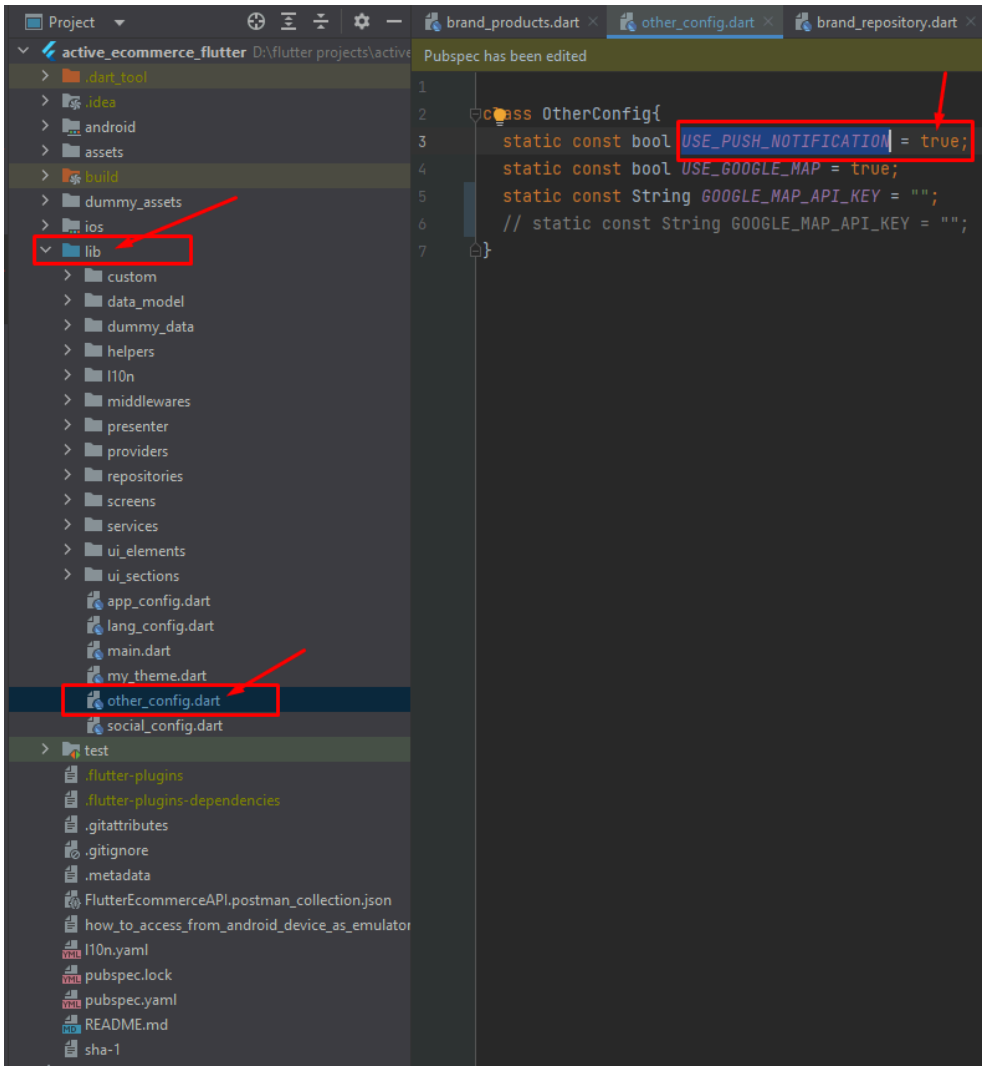
Configure properly as we mention in our ecommerce cms documentation. You don't need extra configuration for the app.

Note: Use the same Bundle ID name for IOS bundle id as you created an apple developer account for web apple signUp.

## 12. How to configure push notification?

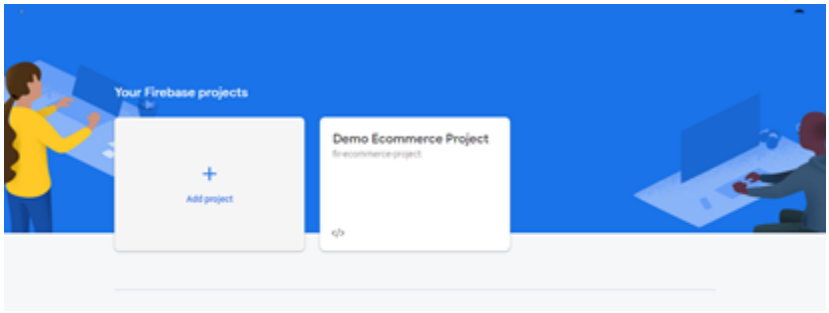
To use firebase follow the procedure which are mentioned below

1. Open other_config.dart file from your project_file->lib->other_config.dart file and put the true in variable USE_PUSH_NOTIFICATION = true;

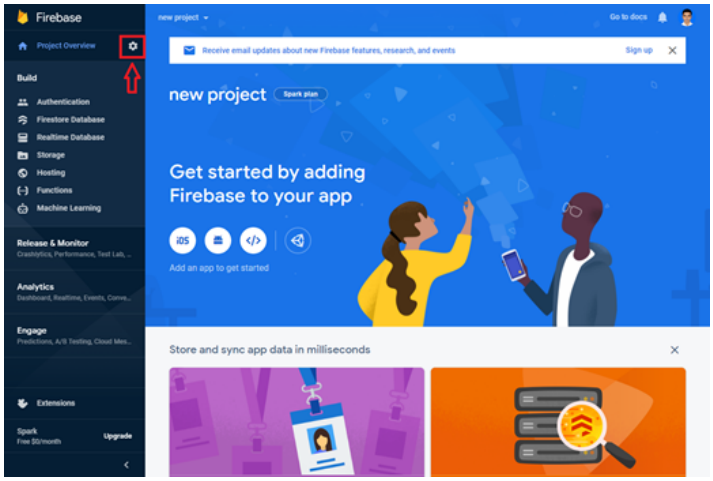


2. Go to this URL to create project https://console.firebase.google.com/u/0/

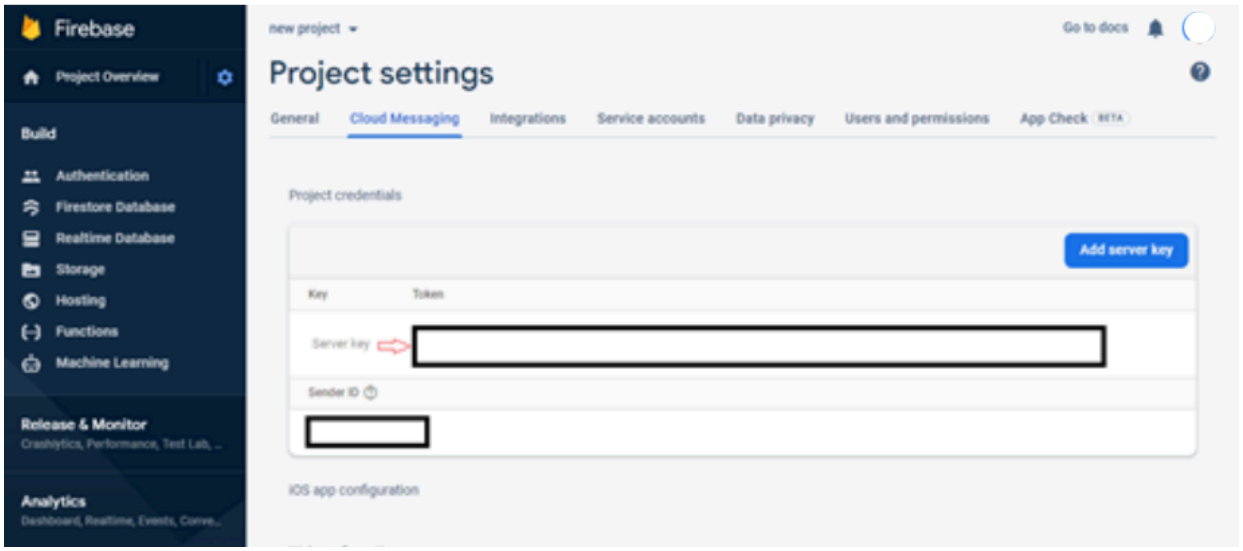
   If you already have a project then continue with that.



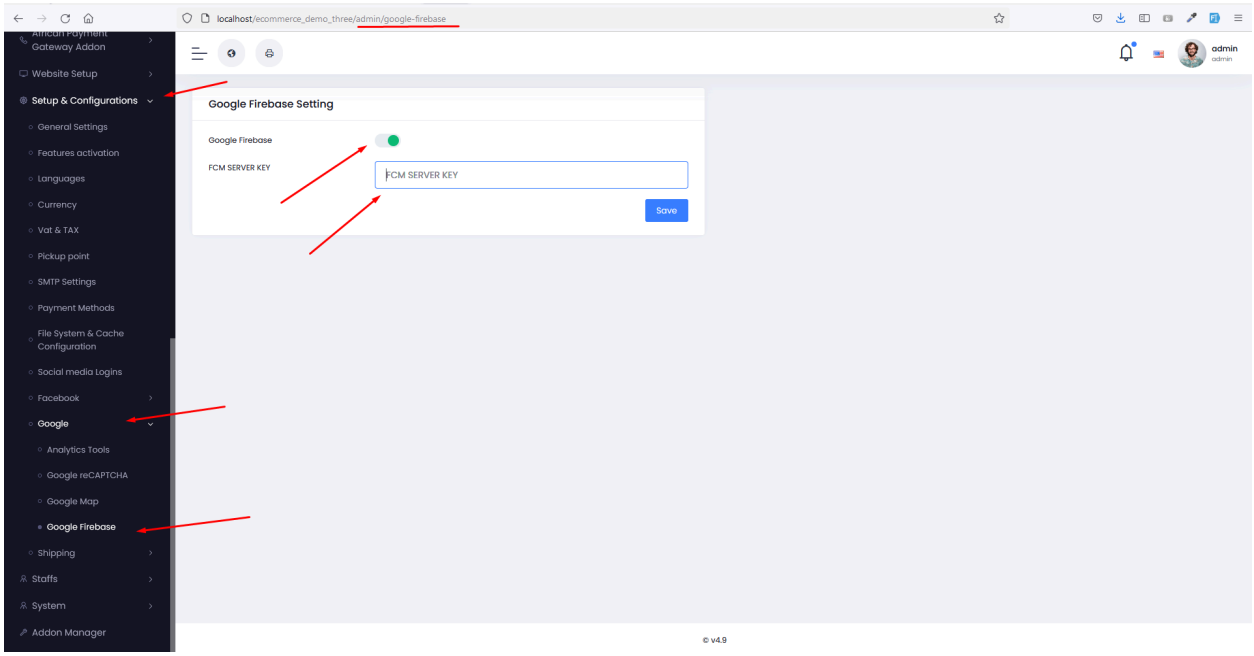

2. Now go to project settings to get server key



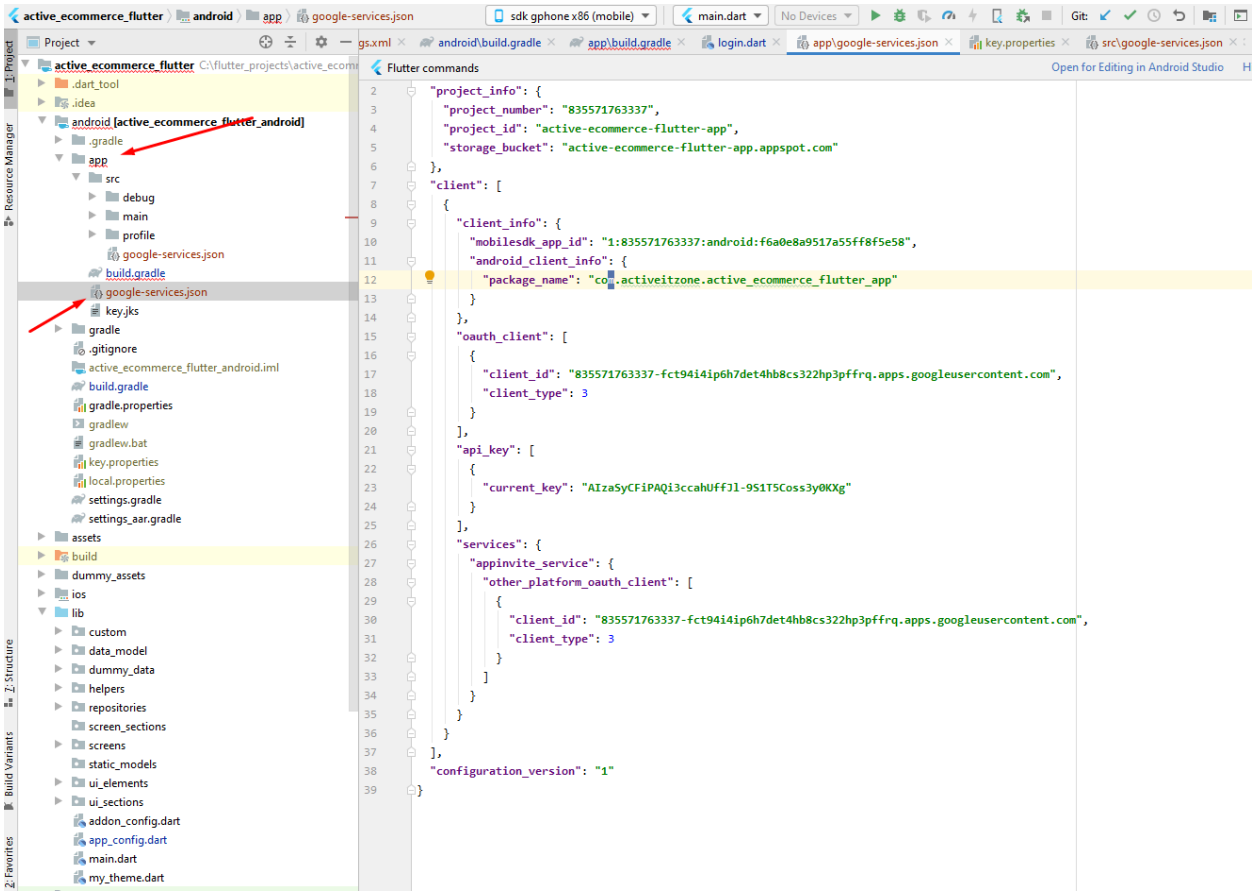

3. To get server key click on Cloud Messaging option




4.Turn on the switch and put the server key in admin panel



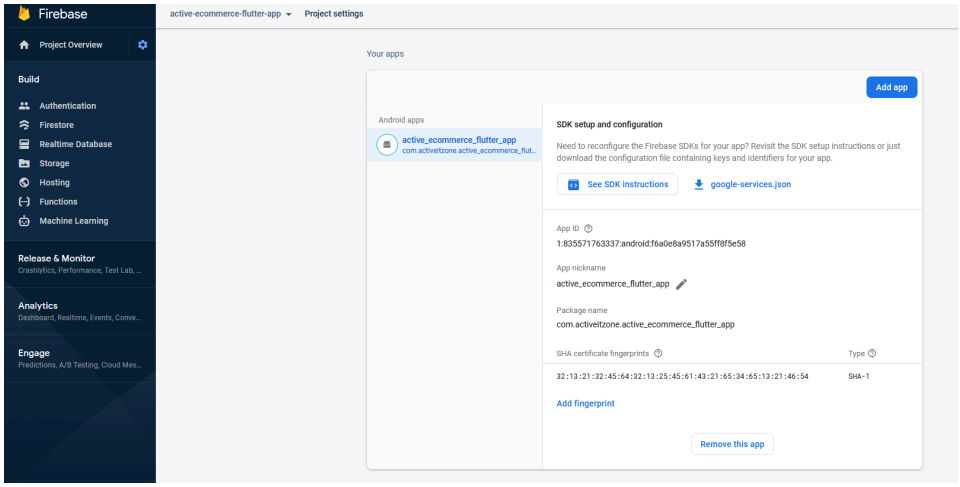

5.You will need to generate your own google-services.json. Do not use ours - it will not work for you



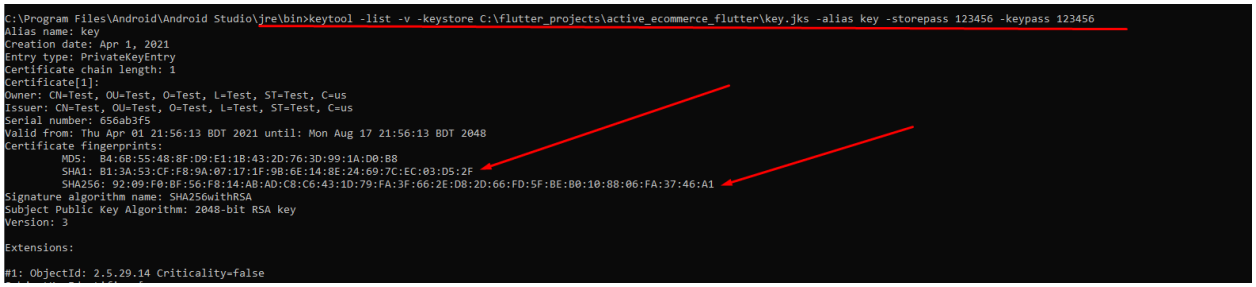

Firebase console:

https://console.firebase.google.com

You need to provide your fingerprints here (sha1 and sha 256)



You will find your signature/fingerprints from here ( Provided that you already have generated the key). You will also need the path of your key.jks. You may have already kept it in the root folder.




7. Although  most of the configuration for android is done you can check guidelines from here.

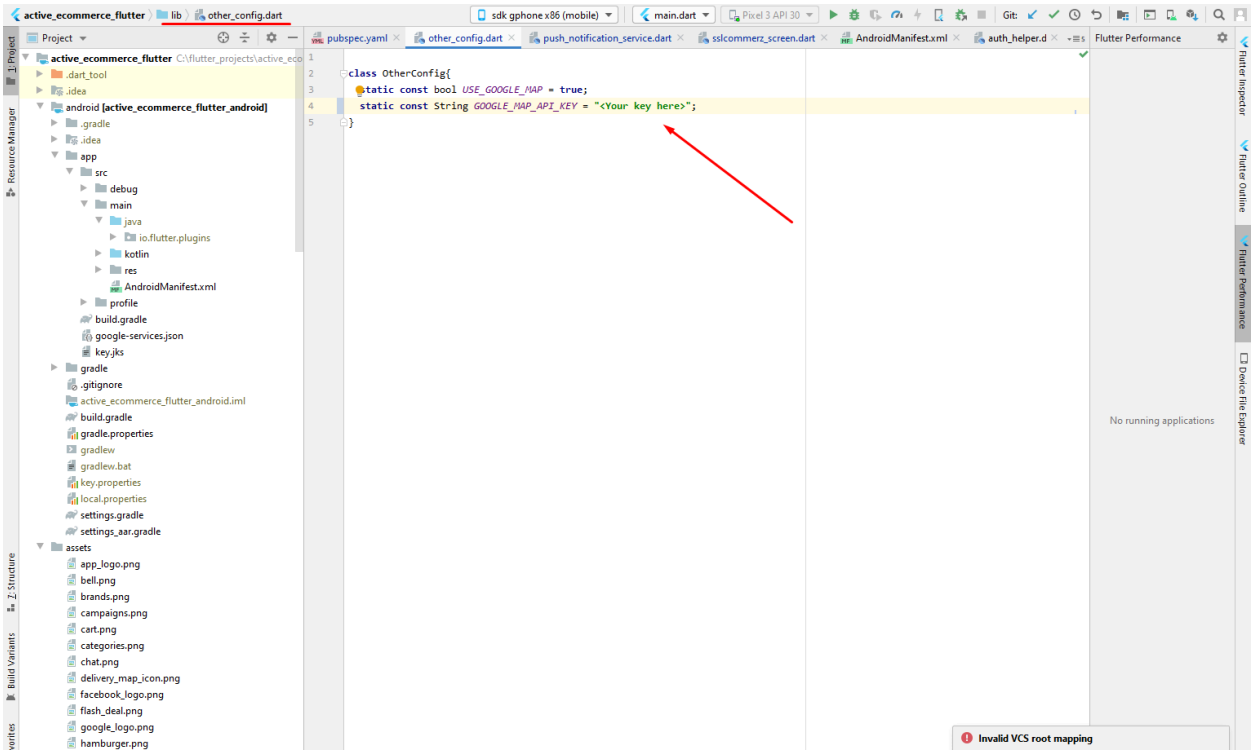
## Apple Integration

Please Follow this URL's documentation
https://firebase.flutter.dev/docs/messaging/apple-integration/

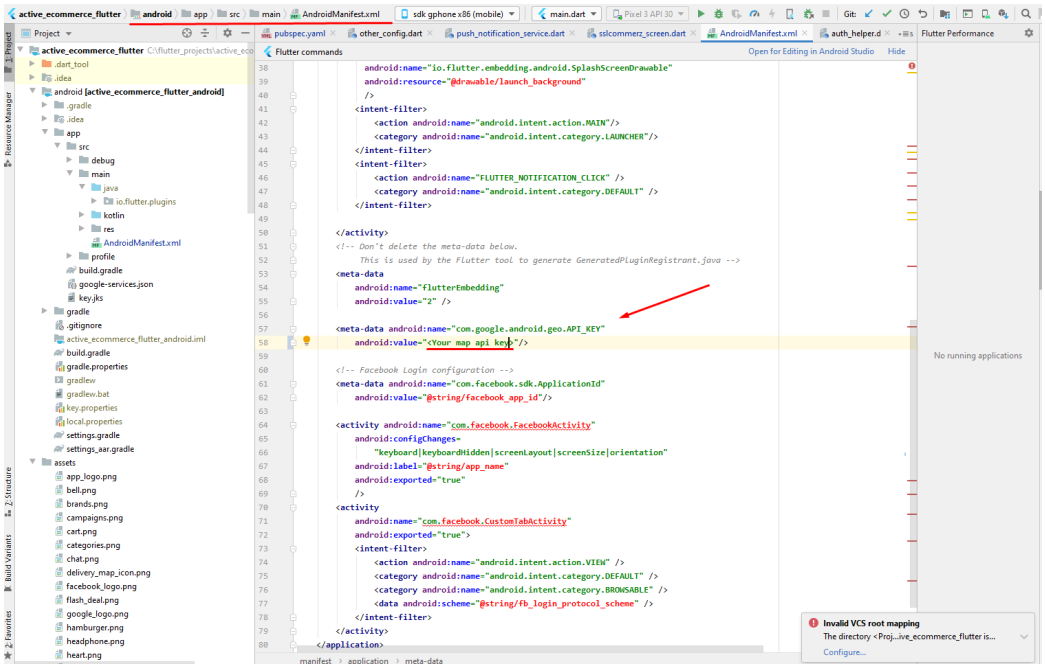
### 13. How to configure google map? (Read the whole thing before implementing)

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 and put google map api key



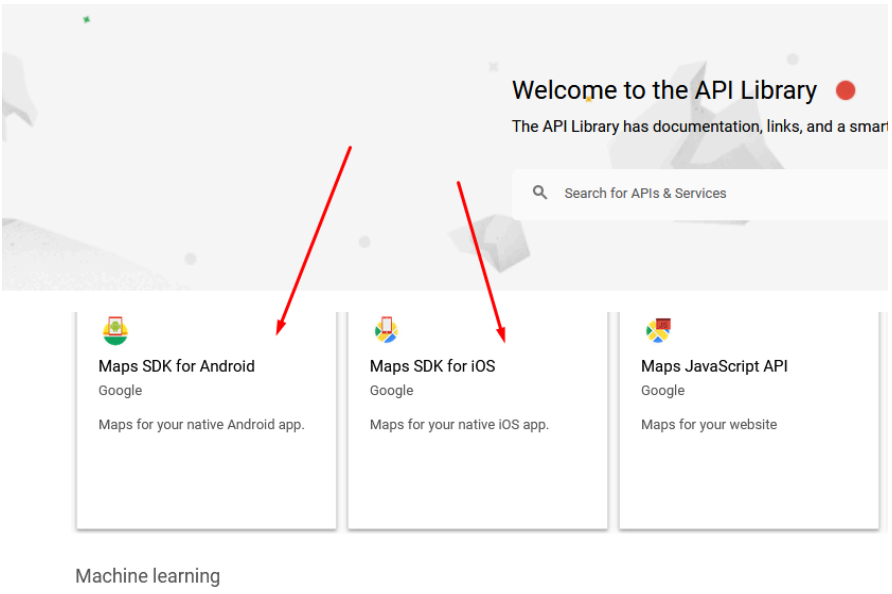

2. In main  AndroidManifest.xml put the map api key

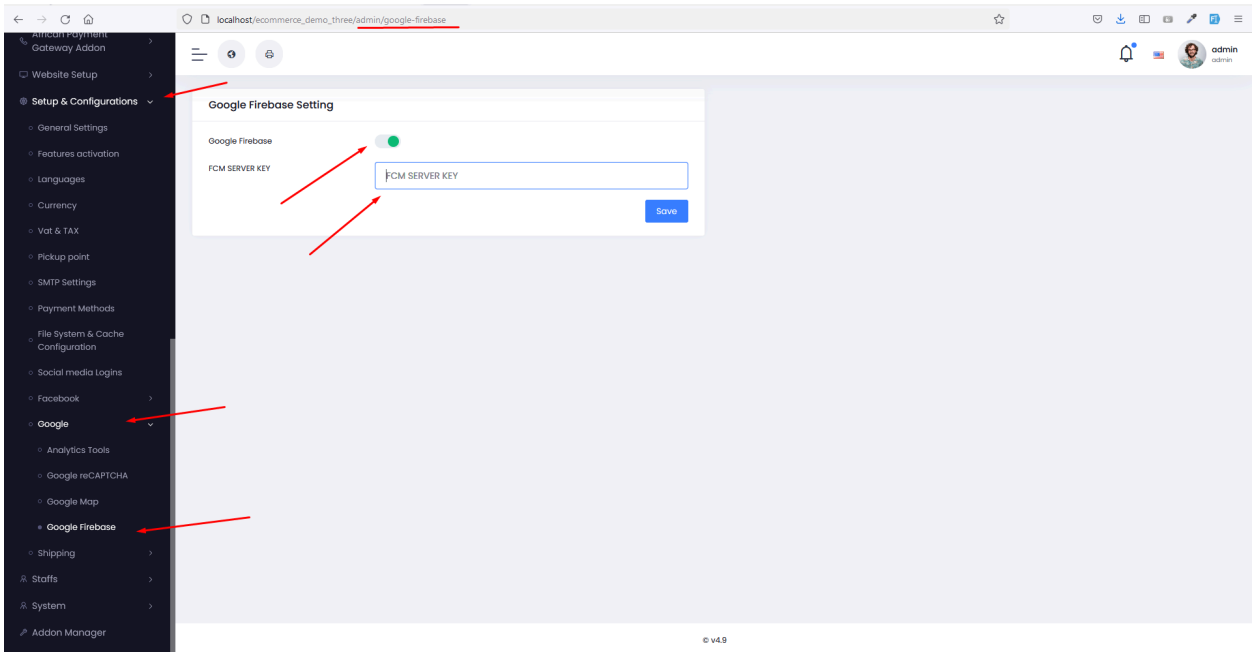



3.For ios follow this

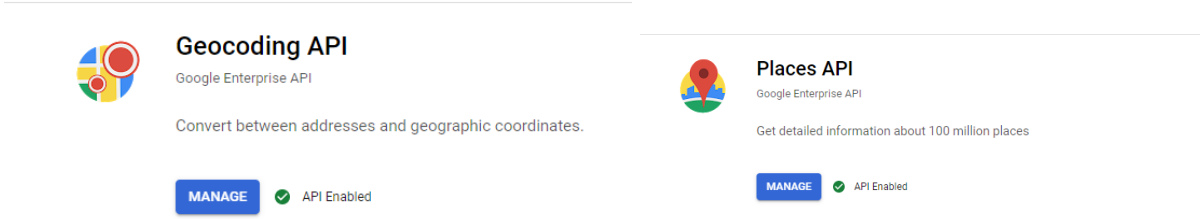
https://blog.logrocket.com/adding-google-maps-to-a-flutter-app/#addinggooglemapstoflutterios

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






5. In the customer app we are searching location via text .And while setting pin to location taking information from the location. For these we would need these apis enabled. Unfortunately these api are not free, you will need to add card.If you do not want to spend money you cannot use google map in the customer app




## 14. How to configure the default language for mobile apps?

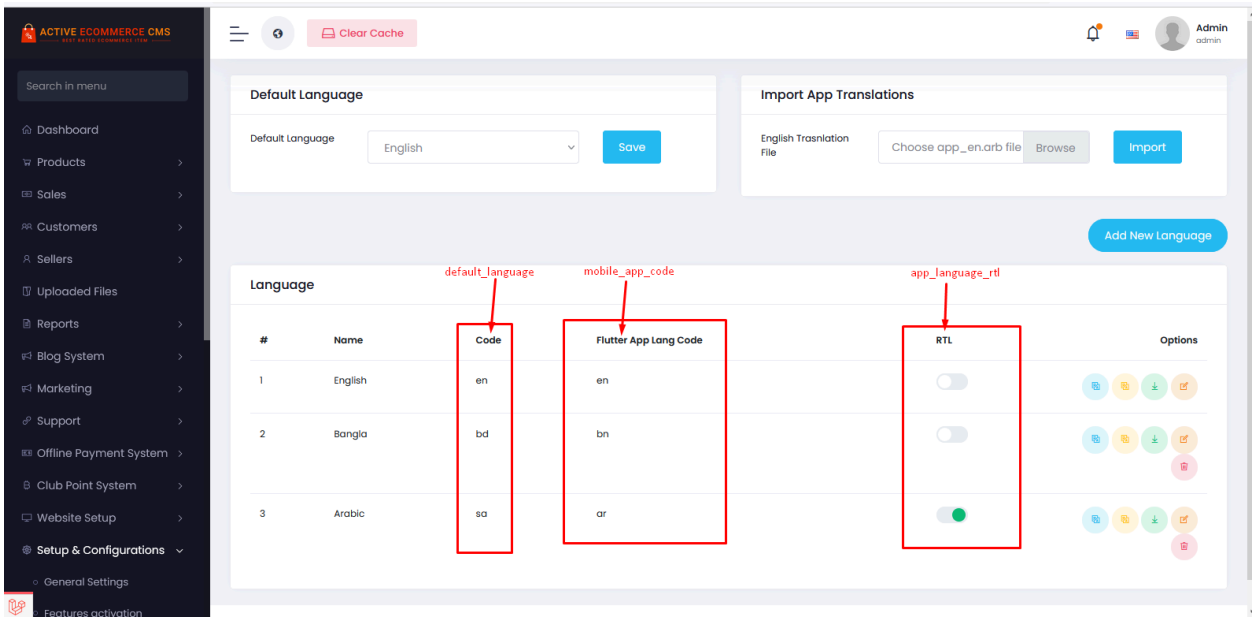
Go to your flutter project->lib->app_config.dart

Change variables value *//Default language config*

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

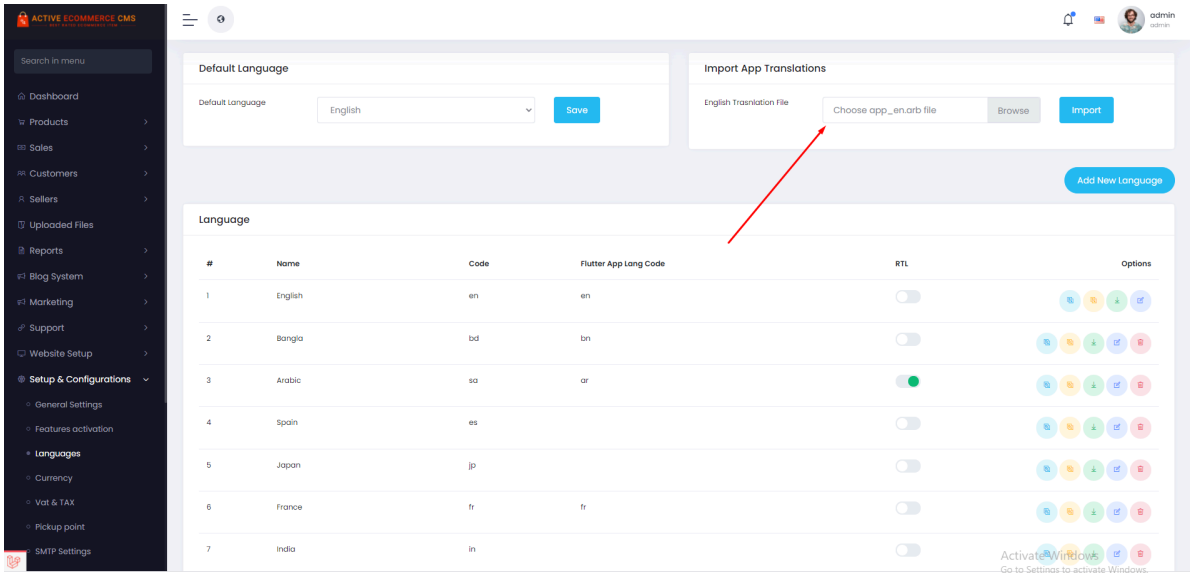
```dart
import 'package:flutter/material.dart';

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.

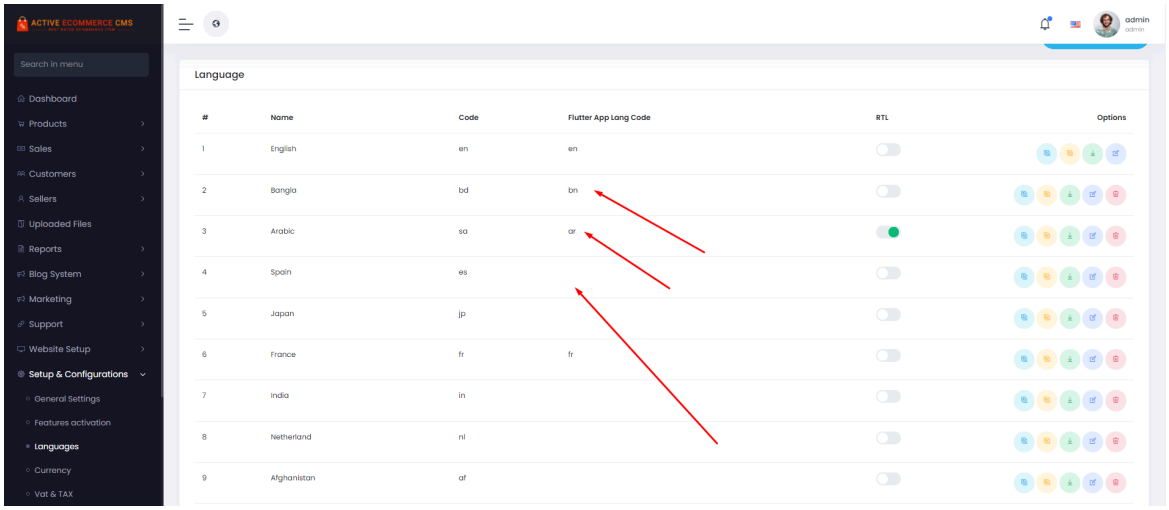


## 15. How to configure multiple languages for mobile app? (Read the whole thing before implementing)

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 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.

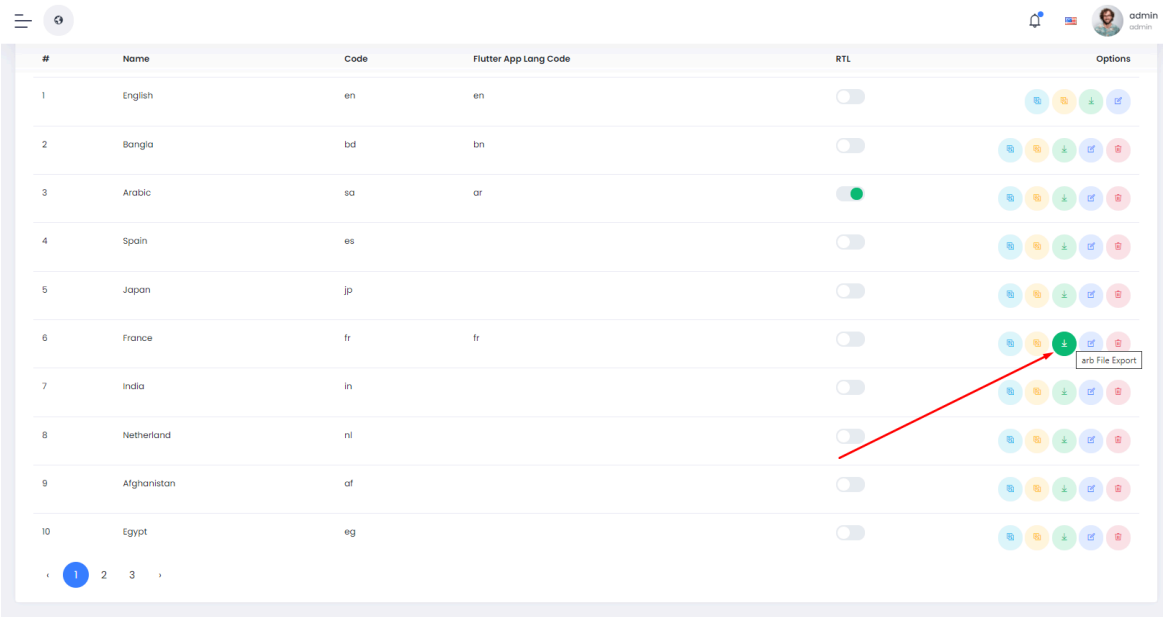



5. Make sure while adding/editing a language , your flutter app language code exists. The code must be in iso 639-1 format. Without a valid code , you will not see a translated output in the app.




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.




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.

## 16. 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.