



# Documentation

**Author :** Active IT zone

**Software Framework :** Flutter

**Addon For:** Active eCommerce CMS

**Name:** Active Ecommerce Flutter App

**Version:** 5.5.0

**Provided by :** codecanyon



## **Documentation**

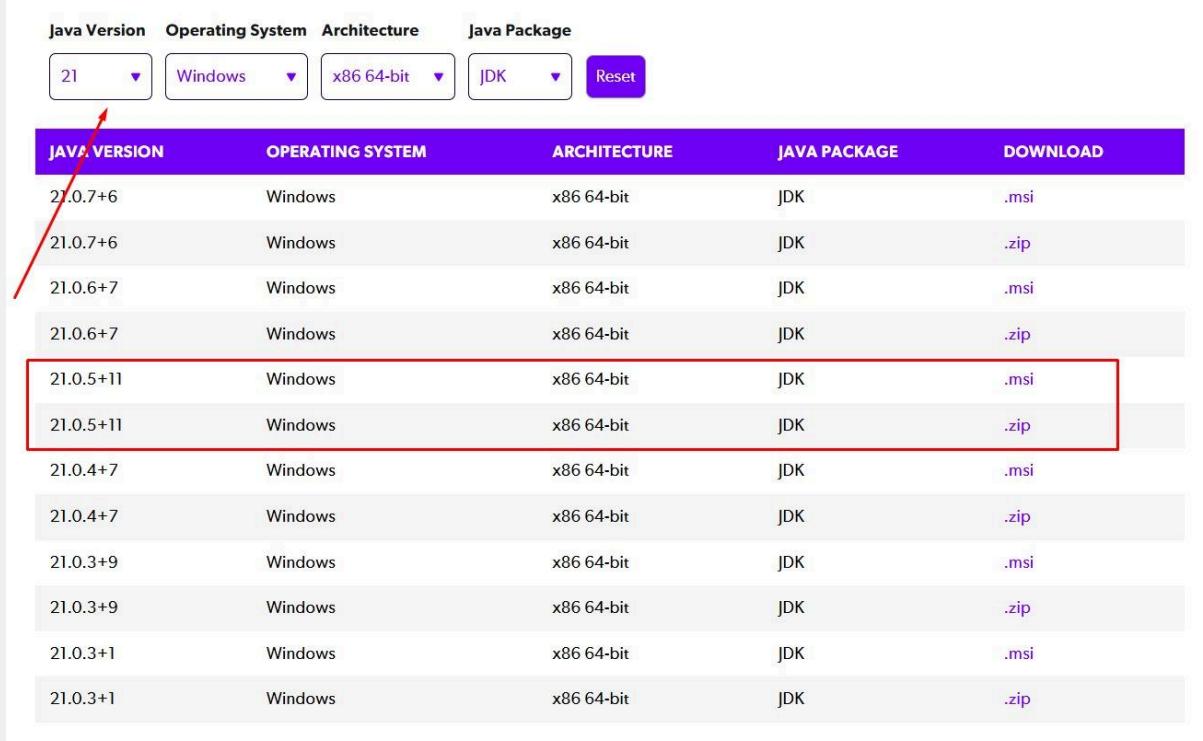
- 1.** What are the prerequisites?
- 2.** How to open the project on Android Studio?
- 3.** How to configure the App according to your setup?
- 4.** How to activate an app?
- 5.** How to change the package name?
- 6.** How to build the App for testing (build and apk) ?
- 7.** How to run IOS on a simulator?
- 8.** How to configure the Stripe payment gateway?
- 9.** How to Update for Android?
- 10.** How to configure social login?
- 11.** How to configure push notification?
- 12.** How to configure google maps ?
- 13.** How to configure the default language for mobile apps?
- 14.** How to configure multiple languages for mobile apps?
- 15.** How to configure the deeplink?
- 16.** How to generate play store uploadable files for release?
- 17.** How to generate app store uploadable files?
- 18.** How to add frequently bought products?
- 19.** 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 3.32.8 • channel stable\(Dart 3.8.1\)](#)
- b. [Android-studio-2024.1.2.13 or Visual Studio Code \(September 2024 \(version 1.94\)\)](#)
- c. [Java version OpenJDK Runtime Environment \(build 21.0.5--12932927\)](#)



The screenshot shows the Oracle Java Download page. At the top, there are four dropdown menus: 'Java Version' set to '21', 'Operating System' set to 'Windows', 'Architecture' set to 'x86 64-bit', and 'Java Package' set to 'JDK'. Below these is a 'Reset' button. The main area is a table with columns: JAVA VERSION, OPERATING SYSTEM, ARCHITECTURE, JAVA PACKAGE, and DOWNLOAD. The table lists several Java versions for Windows x86 64-bit. A red arrow points from the 'Java Version' dropdown down to the table. A red box highlights the rows for Java 21.0.5+11, which are the correct versions for the Flutter project.

JAVA VERSION	OPERATING SYSTEM	ARCHITECTURE	JAVA PACKAGE	DOWNLOAD
21.0.7+6	Windows	x86 64-bit	JDK	.msi
21.0.7+6	Windows	x86 64-bit	JDK	.zip
21.0.6+7	Windows	x86 64-bit	JDK	.msi
21.0.6+7	Windows	x86 64-bit	JDK	.zip
21.0.5+11	Windows	x86 64-bit	JDK	.msi
21.0.5+11	Windows	x86 64-bit	JDK	.zip
21.0.4+7	Windows	x86 64-bit	JDK	.msi
21.0.4+7	Windows	x86 64-bit	JDK	.zip
21.0.3+9	Windows	x86 64-bit	JDK	.msi
21.0.3+9	Windows	x86 64-bit	JDK	.zip
21.0.3+1	Windows	x86 64-bit	JDK	.msi
21.0.3+1	Windows	x86 64-bit	JDK	.zip

- 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.

```
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Active IT Zone>flutter doctor -v
[✓] Flutter (Channel stable, 3.32.8, on Microsoft Windows [Version 10.0.19045.6093], locale en-US) [909ms]
  • Flutter version 3.32.8 on channel stable at C:\Apps Development\flutter
  • Upstream repository https://github.com/flutter/flutter.git
  • Framework revision edada7c56e (3 days ago), 2025-07-25 14:08:03 +0000
  • Engine revision ef0cd00001
  • Dart version 3.8.1
  • DevTools version 2.45.1

[✓] Windows Version (10 Pro 64-bit, 22H2, 2009) [4.2s]

[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.0) [5.0s]
  • Android SDK at C:\Users\Active IT Zone\AppData\Local\Android\sdk
  • Platform android-35-ext14, build-tools 35.0.0
  • Java binary at: C:\Apps Development\java21\bin\java
  • This JDK is specified in your Flutter configuration.
    To change the current JDK, run: `flutter config --jdk-dir="path/to/jdk"`.
  • Java version OpenJDK Runtime Environment OpenLogic-OpenJDK (build 21.0.5+11-adhoc Administrator.jdk21u)
  • All Android licenses accepted.

[✓] Chrome - develop for the web [327ms]
  • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.13.1) [325ms]
  • Visual Studio at C:\Program Files\Microsoft Visual Studio\2022\Community
```

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

**Answer:**

Extract the **sOURCE\_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, search\_bar\_text, 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>.

The screenshot shows a web-based activation form for an Active It Zone addon. The form is titled "Active It Zone Addon Activation". It includes several input fields: "Your Name" (text), "Your Email" (text), "CodeCanyon Purchase Code" (text), "Where Is My Purchase Code?" (text), "CodeCanyon User Name" (text), "Domain Name (example: www.abcd.com)" (text), "Main Item" (dropdown menu), and "Select Item" (dropdown menu). At the bottom of the form is a large blue "Activate" button. To the left of the main form, there is a sidebar with the title "Instruction for activating purchase code" and two bullet points: "1. Make sure you have valid purchase code." and "2. Make sure your CodeCanyon username is correct."

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

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

```

active_ecommerce_flutter D:\StudioProjects\active_ecommerce_flutter
> .dart_tool
> .idea
> android
> assets
> build
> dummy_assets
> ios
> lib
>   custom
>   data_model
>   dummy_data
>   helpers
>   l10n
>   middlewares
>   presenter
>   providers
>   repositories
>   screens
>   services
>   ui_elements
>   ui_sections
>     app_config.dart
>     lang_config.dart
>     main.dart
>     my_theme.dart
>     other_config.dart
>     social_config.dart
> test
>   flutter-plugins
>   flutter-plugins-dependencies
>   gitattributes
>   ignore
>   metadata
>   FlutterCommerceAPI.postman_collection.json
>   how_to_access_from_android_device_as_emulator.txt
>   l10n.yaml
>   pubspec.lock
>   pubspec.yaml
>   README.md
1  var this_year = DateTime.now().year.toString();
2
3  class AppConfig {
4    static String copyright_text =
5      "@ ActiveITZone " + this_year; //this shows in the splash screen
6    static String app_name = "Active eCommerce"; //this shows in the splash screen
7
8    static String purchase_code =
9      "bkash"; //enter your purchase code for the app from codecanyon
10   static String system_key =
11      "\$2 \$1 \$0"; //enter your purchase
12
13   //Default language config
14   static String default_language = "en";
15   static String mobile_app_code = "en";
16   static bool app_language_rtl = false;
17
18   //configure this
19   static const bool HTTPS = false;
20
21   static const DOMAIN_PATH = "domain.com"; //localhost
22
23   //do not configure these below
24   static const String API_ENDPOINT = "api/v2";
25   static const String PROTOCOL = HTTPS ? "https://" : "http://";
26   static const String RAW_BASE_URL = "${PROTOCOL}${DOMAIN_PATH}";
27   static const String BASE_URL = "${RAW_BASE_URL}${API_ENDPOINT}";
28
29   @override
30   String toString() {
31     // TODO: implement toString
32     return super.toString();
33   }
34
35 }

```

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

```

APP_NAME="Active eCommerce CMS"
APP_ENV=local
APP_KEY=base64:WrvqXJ+ilWOLItqI7C2N2R2dQkqFAHFYzTrQBBJqtso=
APP_DEBUG=true
APP_URL="http://localhost/e-commerce"
APP_TIMEZONE="UTC"
SYSTEM_KEY=""

DEMO_MODE="Off"

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST="localhost"
DB_PORT="3306"
DB_DATABASE="db_ecommerce"
DB_USERNAME="root"
DB_PASSWORD=""

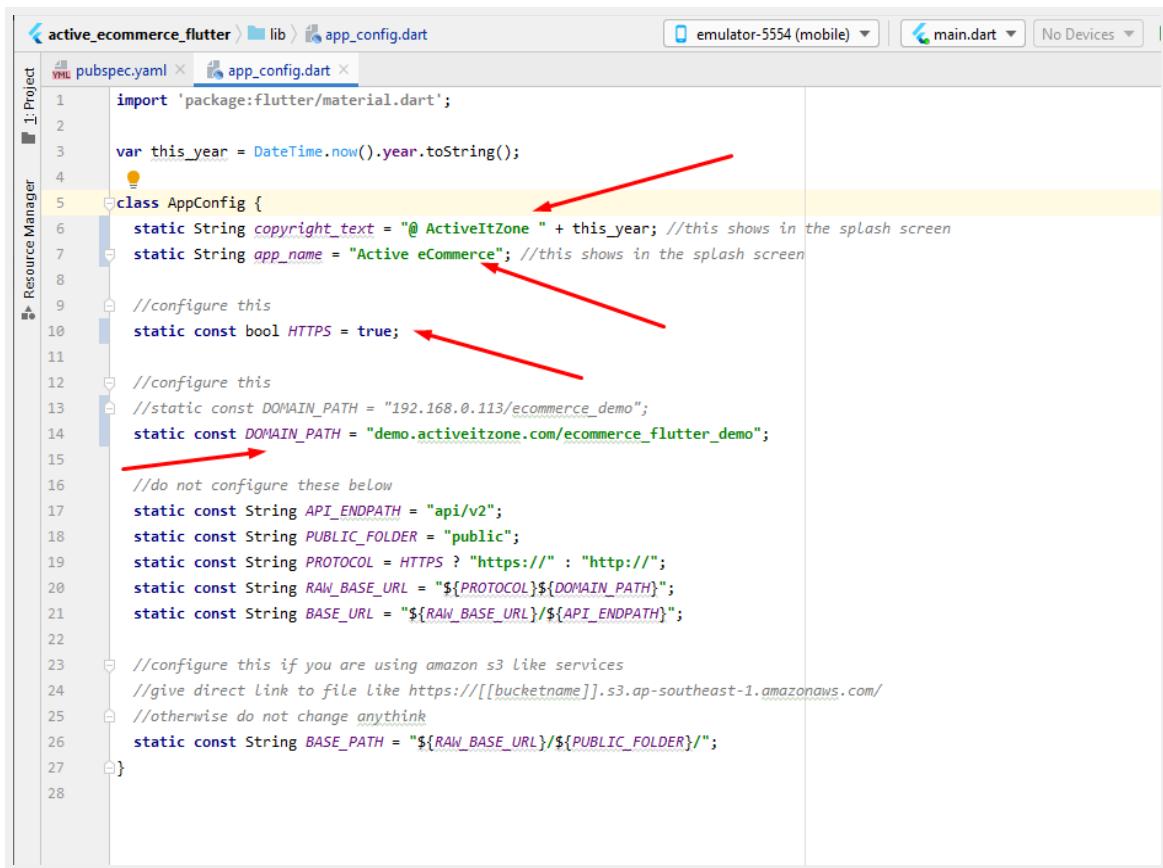
```

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 \*\***



```
active_ecommerce_flutter lib app_config.dart
pubspec.yaml main.dart No Devices

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 screen
  static String app_name = "Active eCommerce"; //this shows in the splash screen

  //configure this
  static const bool HTTPS = true;

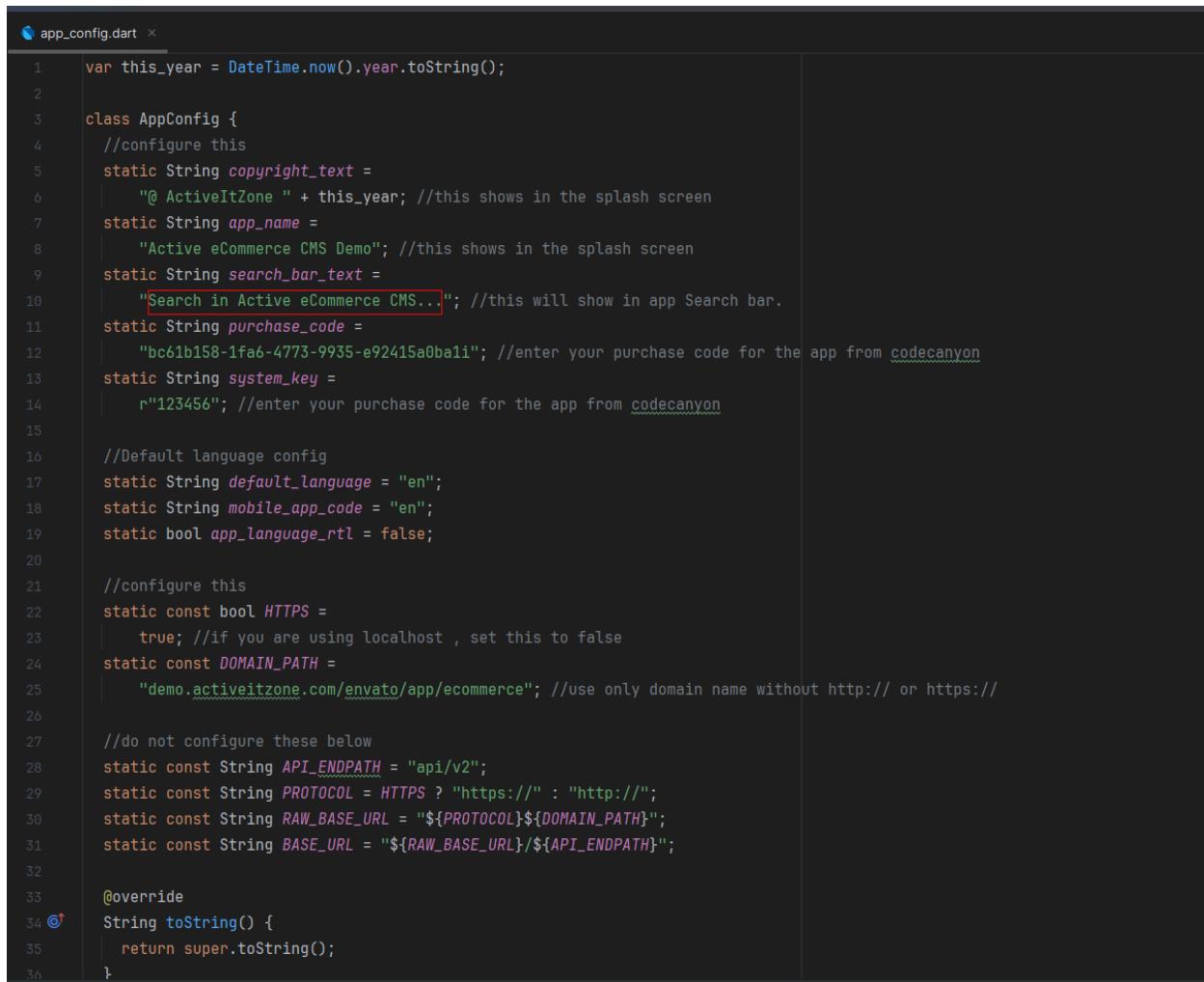
  //configure this
  //static const DOMAIN_PATH = "192.168.0.113/ecommerce_demo";
  static const DOMAIN_PATH = "demo.activeitzone.com/ecommerce_flutter_demo";

  //do not configure these below
  static const String API_ENDPOINT = "api/v2";
  static const String PUBLIC_FOLDER = "public";
  static const String PROTOCOL = HTTPS ? "https://" : "http://";
  static const String RAW_BASE_URL = "${PROTOCOL}${DOMAIN_PATH}";
  static const String BASE_URL = "${RAW_BASE_URL}/${API_ENDPOINT}";

  //configure this if you are using amazon s3 like services
  //give direct link to file like https://[[bucketname]].s3.ap-southeast-1.amazonaws.com/
  //otherwise do not change anything
  static const String BASE_PATH = "${RAW_BASE_URL}/${PUBLIC_FOLDER}/";
}


```

\*Change this for Search bar:



```
1 var this_year = DateTime.now().year.toString();
2
3 class AppConfig {
4     //configure this
5     static String copyright_text =
6         "@ ActiveITZone " + this_year; //this shows in the splash screen
7     static String app_name =
8         "Active eCommerce CMS Demo"; //this shows in the splash screen
9     static String search_bar_text =
10        "Search in Active eCommerce CMS..."; //this will show in app Search bar.
11     static String purchase_code =
12        "bc61b158-1fa6-4773-9935-e92415a0ba11"; //enter your purchase code for the app from codecanyon
13     static String system_key =
14        "123456"; //enter your purchase code for the app from codecanyon
15
16     //Default language config
17     static String default_language = "en";
18     static String mobile_app_code = "en";
19     static bool app_language_rtl = false;
20
21     //configure this
22     static const bool HTTPS =
23         true; //if you are using localhost , set this to false
24     static const DOMAIN_PATH =
25         "demo.activeitzone.com/envato/app/eCommerce"; //use only domain name without http:// or https://
26
27     //do not configure these below
28     static const String API_ENDPATH = "api/v2";
29     static const String PROTOCOL = HTTPS ? "https://" : "http//";
30     static const String RAW_BASE_URL = "${PROTOCOL}${DOMAIN_PATH}";
31     static const String BASE_URL = "${RAW_BASE_URL}/${API_ENDPATH}";
32
33     @override
34     String toString() {
35         return super.toString();
36     }

```

Change this marked text to change your Searchbar text

#### d. Theme Config:

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

Open lib/my\_theme.dart

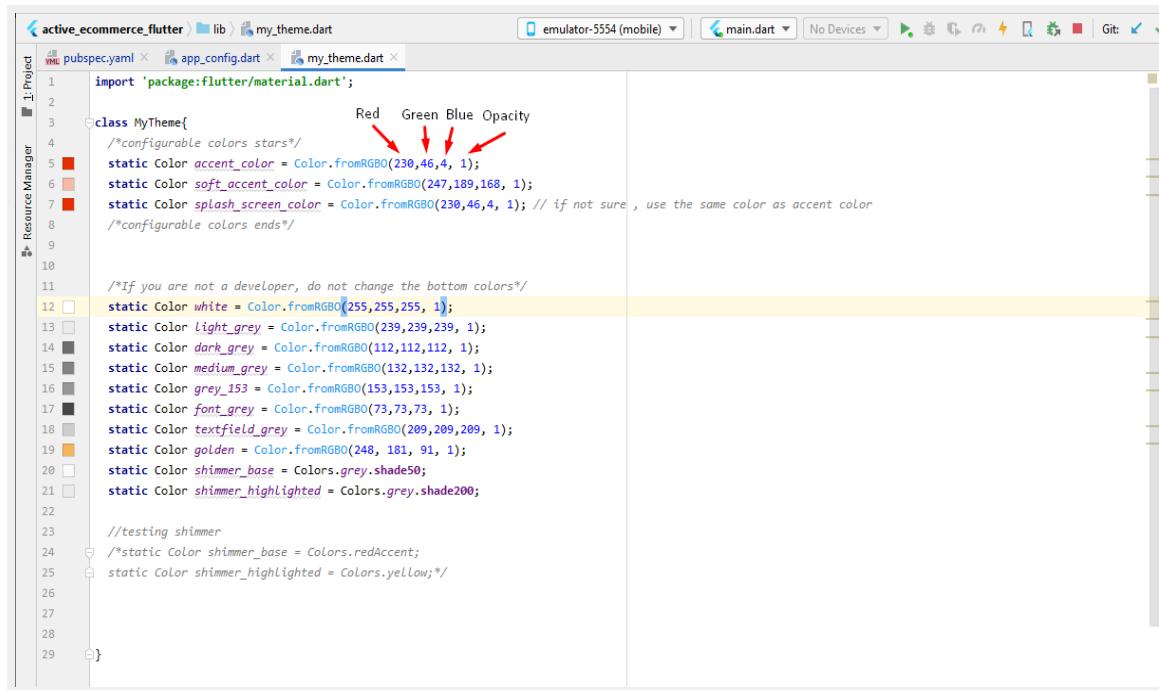
You can change the `accent_color`, `soft_accent_color`, `splash_screen_color` variable.

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:



```
active ecommerce_flutter lib my_theme.dart
pubspec.yaml app_config.dart my_theme.dart
Project
1 import 'package:flutter/material.dart';
2
3 class MyTheme{ Red Green Blue Opacity
4   /*configurable colors starts*/
5   static Color accent_color = Color.fromRGBO(230,46,4, 1);
6   static Color soft_accent_color = Color.fromRGBO(247,189,168, 1);
7   static Color splash_screen_color = Color.fromRGBO(230,46,4, 1); // if not sure , use the same color as accent color
8   /*configurable colors ends*/
9
10  /*If you are not a developer, do not change the bottom colors*/
11  static Color white = Color.fromRGBO(255,255,255, 1);
12  static Color light_grey = Color.fromRGBO(239,239,239, 1);
13  static Color dark_grey = Color.fromRGBO(112,112,112, 1);
14  static Color medium_grey = Color.fromRGBO(132,132,132, 1);
15  static Color grey_153 = Color.fromRGBO(153,153,153, 1);
16  static Color font_grey = Color.fromRGBO(73,73,73, 1);
17  static Color textfield_grey = Color.fromRGBO(209,209,209, 1);
18  static Color golden = Color.fromRGBO(248, 181, 91, 1);
19  static Color shimmer_base = Colors.grey.shade50;
20  static Color shimmer_highlighted = Colors.grey.shade200;
21
22  //testing shimmer
23  /*static Color shimmer_base = Colors.redAccent;
24  static Color shimmer_highlighted = Colors.yellow;*/}
25
26
27
28
29 }
```

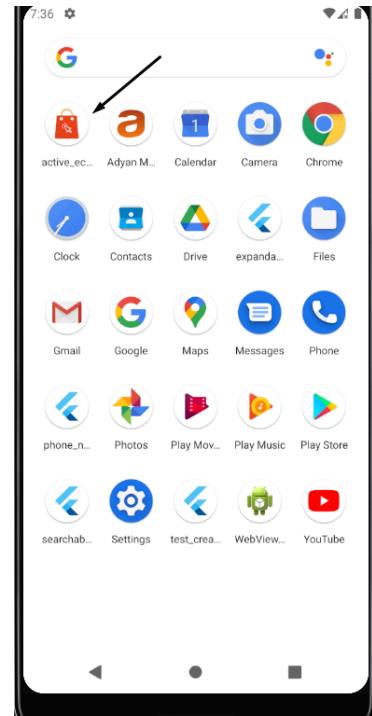
#### e. Configure the launcher icon:

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`

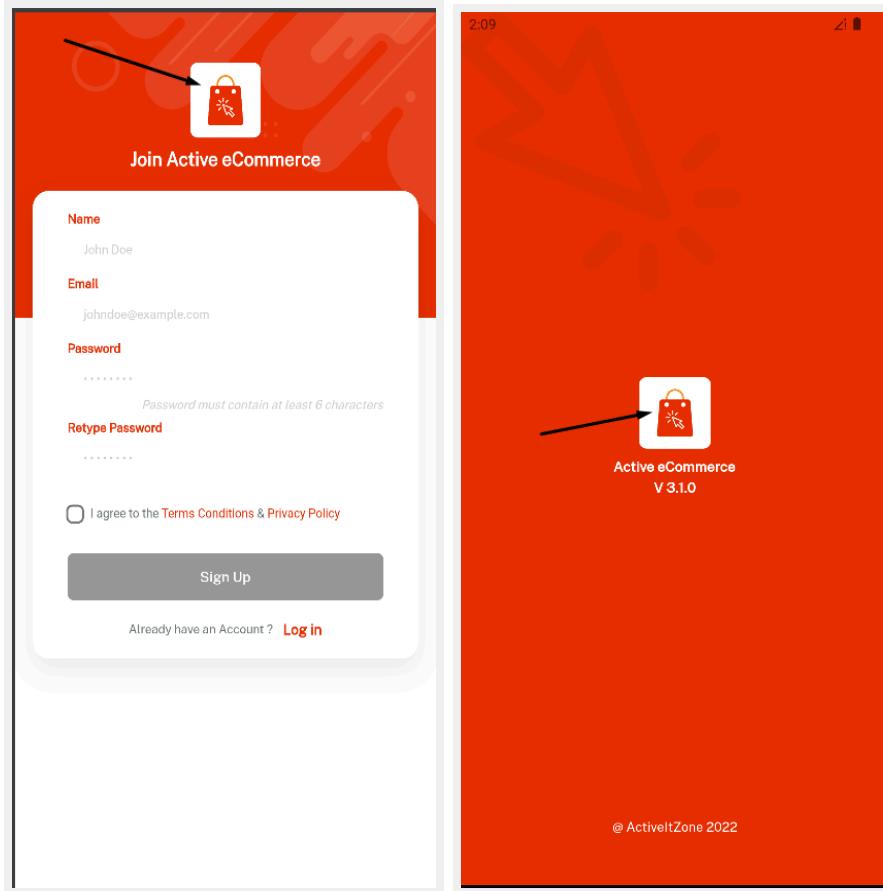
Then run:

`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 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.

#### **5. 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,

**For IOS Only :**

Run this command inside your flutter project root.

Run the command in Android studio terminal:

```
flutter pub run rename setAppName --targets=ios,android --value="Network Upp"
```

```
flutter pub run rename setBundleId --value=com.onatcipli.networkUpp
```

**Manually: 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.

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.

**For Android Only :**

Run the command in Android studio terminal:

```
flutter pub run rename setAppName --targets=ios,android --value="Network Upp"
```

```
flutter pub run rename setBundleId --value=com.onatcipli.networkUpp
```

for package name just change in build build.gradle.kts only ([android/app/build.gradle.kts](#))

You can search ctrl+f namespace

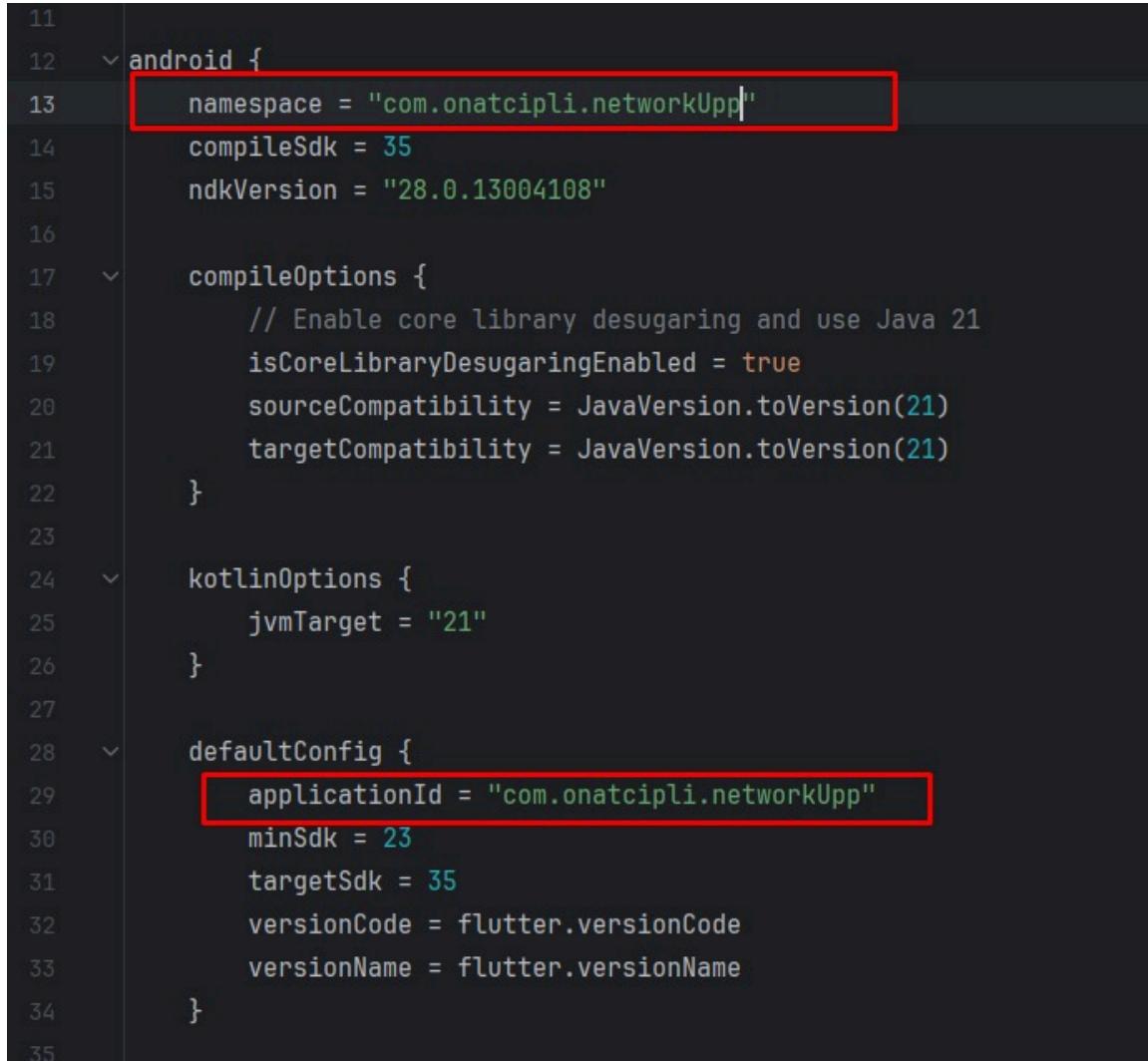
```
namespace = "com.onatcipli.networkUpp"
```

```
defaultConfig {
```

```
    applicationId = "com.onatcipli.networkUpp" .....
```

```
}
```

For better understand , please check the screenshot



```
11
12     android {
13         namespace = "com.onatcipli.networkUpp"
14         compileSdk = 35
15         ndkVersion = "28.0.13004108"
16
17         compileOptions {
18             // Enable core library desugaring and use Java 21
19             isCoreLibraryDesugaringEnabled = true
20             sourceCompatibility = JavaVersion.toVersion(21)
21             targetCompatibility = JavaVersion.toVersion(21)
22         }
23
24         kotlinOptions {
25             jvmTarget = "21"
26         }
27
28         defaultConfig {
29             applicationId = "com.onatcipli.networkUpp"
30             minSdk = 23
31             targetSdk = 35
32             versionCode = flutter.versionCode
33             versionName = flutter.versionName
34         }
35     }
```

\*\*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

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
    "project_info": {
        "project_number": "835571763337",
        "project_id": "active-commerce-flutter-app",
        "storage_bucket": "active-commerce-flutter-app.appspot.com"
    },
    "client": [
        {
            "client_info": {
                "mobilesdk_app_id": "1:835571763337:android:f6a0e8a9517a55ff8f5e58",
                "android_client_info": {
                    "package_name": "com.activeitzone.active_ecommerce_flutter_app"
                }
            },
            "oauth_client": [
                {
                    "client_id": "835571763337-6e717rb6gc6p2vlr35lv1tmer4dek4.apps.googleusercontent.com",
                    "client_type": 1,
                    "android_info": {
                        "package_name": "com.activeitzone.active_ecommerce_flutter_app",
                        "certificate_hash": "b13a53cff89a07171f0b6e148e24697cc03d52f"
                    }
                },
                {
                    "client_id": "835571763337-fct94i4ip6h7det4hb8cs322hp3pffrq.apps.googleusercontent.com",
                    "client_type": 3
                }
            ]
        }
    ]
}

```

## 6. How to build the app for testing (build an apk)?

**Answer:**

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

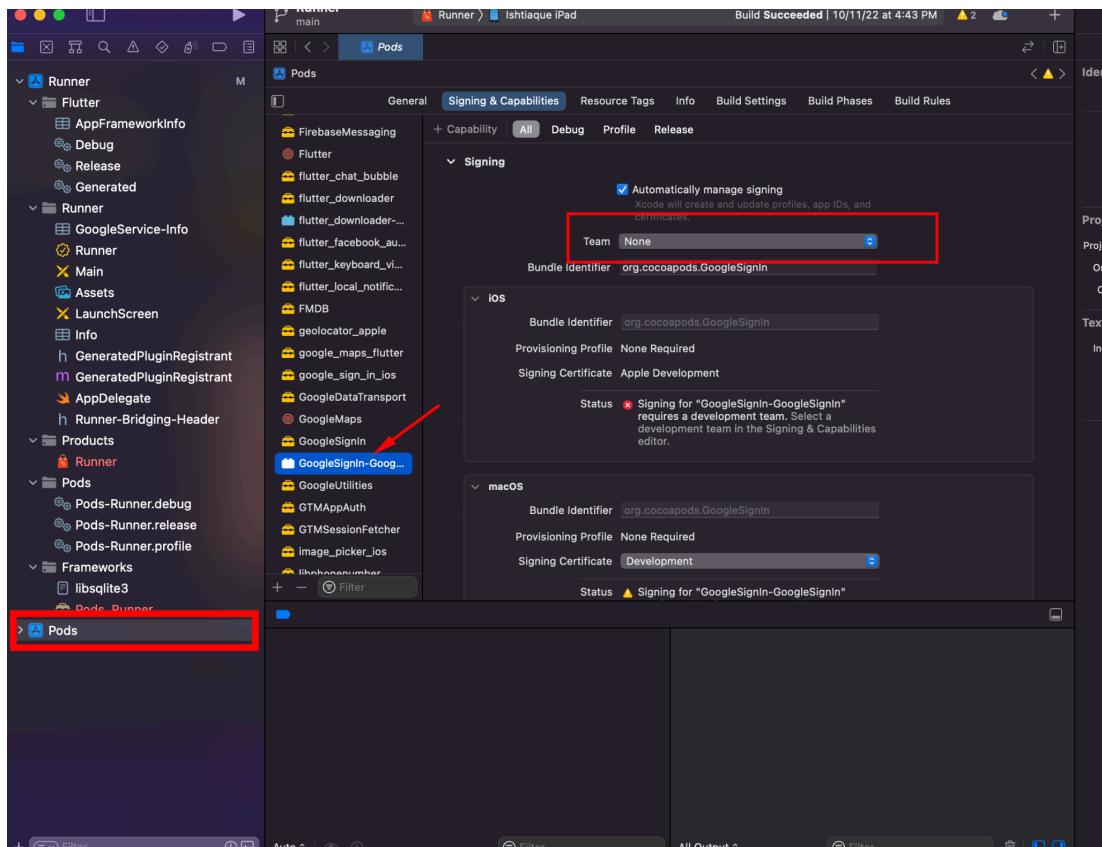
In terminal run: **flutter build apk**

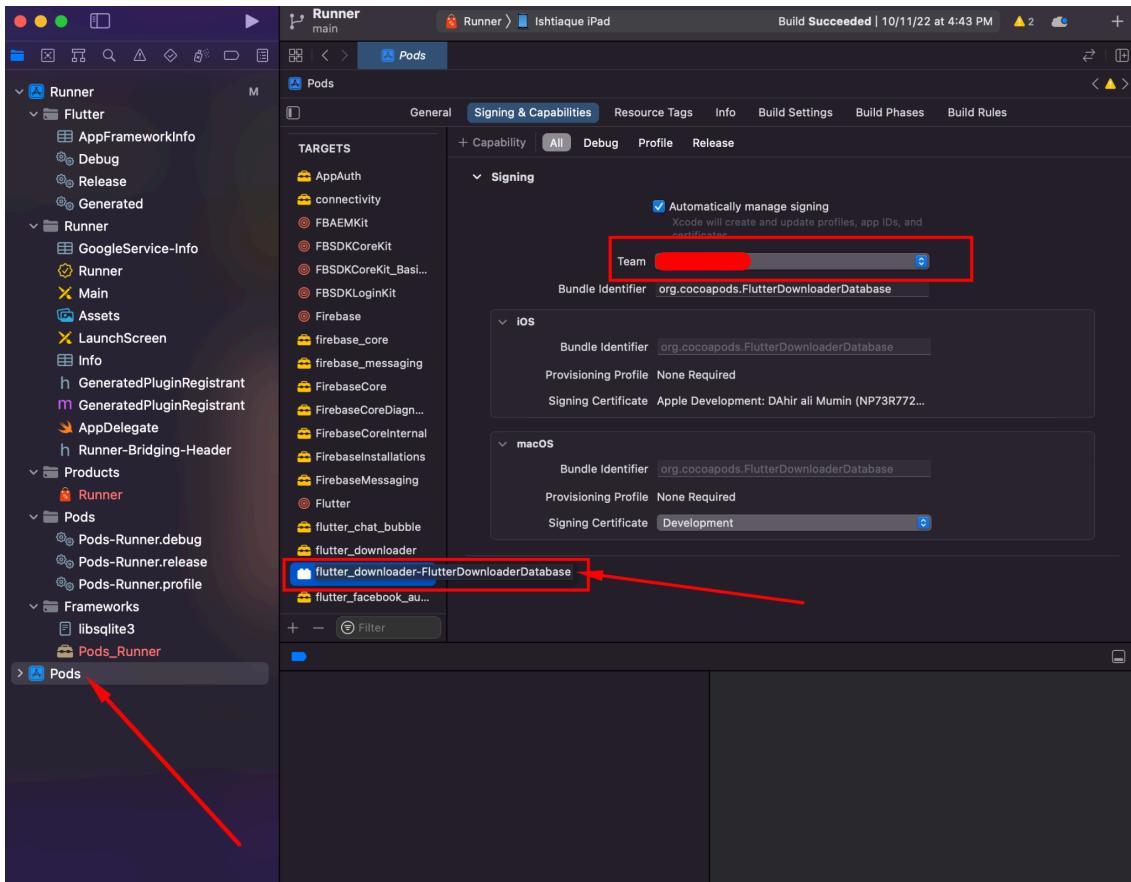
It will build an apk and show the folder. You can then install it on your phone to test, or share to multiple users for testing.

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

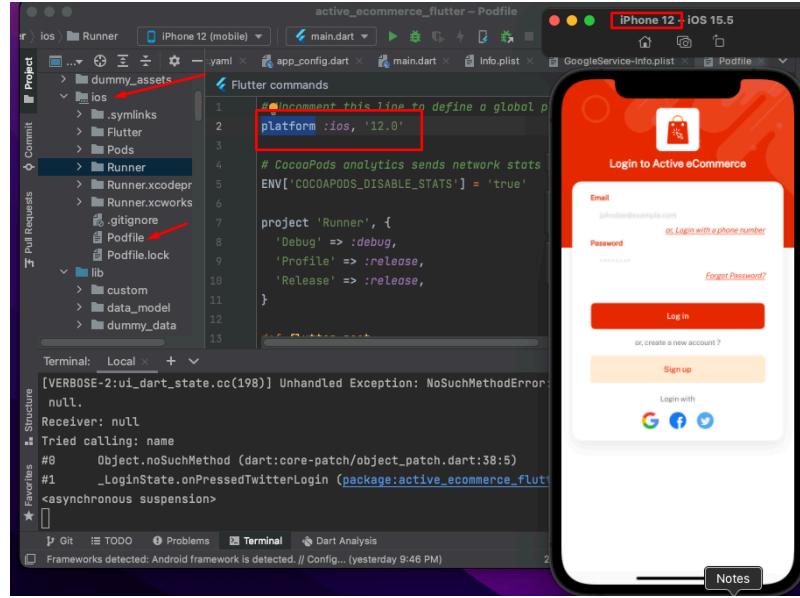
**Answer:**

- Install Xcode on your mac.
- Open your iOS folder in Xcode and select your developer team from Signin and Capabilities.
- 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 **flutter run**.

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

## 8. How to configure the Stripe payment gateway?

**Answer:**

Please follow the below procedure:

Just put www before your domain name

Also make sure your server is configured with www

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

**Answer:**

- 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.

## 10. 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.firebaseio.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.

```
C:\flutter_projects\active_ecommerce_flutter\android>gradlew signinReport
> Configure project :app
WARNING: The option setting 'android.enableR8=true' is deprecated.
It will be removed in version 7.0 of the Android Gradle plugin.
You will no longer be able to disable R8
WARNING: Please remove usages of 'jcenter()' Maven repository from your build scripts and migrate your build to other Maven repositories.
This repository is deprecated and it will be shut down in the future.
See http://developer.android.com/r/tools/jcenter-end-of-service for more information.
Currently detected usages in: root project 'android', project ':app', project ':connectivity', ...

> Task :app:signingReport
Variant: debug
Config: debug
Store: C:/Users/ActiveITZone/.android/debug.keystore
Alias: AndroidDebugKey
MD5: F5:AA:B1:22:C1:21:B0:37:CC:DA:0E:44:77:00:42:4C
SHA1: 82:53:85:FC:C9:A8:1B:69:35:2E:F1:14:09:AE:43:6A:D3:E5:84
SHA-256: E1:81:46:48:7D:98:00:35:DA:43:48:9C:12:9A:AC:7A:3F:66:00:3F:FF:36:F5:DF:54:C2:B1:1F:8D:E9:0A:2B
Valid until: Monday, June 20, 2050
```

- 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.

The screenshot shows the Android Studio interface with the following details:

- Project Tree:** The project structure is displayed with red arrows pointing to the `android` and `app` folders.
- Terminal:** The terminal window shows the command `C:\Flutter_projects\active_ecommerce_flutter\android\keytool -list -v -keystore key.jks -alias key` being run, which results in an error message indicating that the file `key.jks` does not exist.
- Code Editor:** The `spp.config.dart` file is open, containing configuration code for domain paths and API endpoints.
- Status Bar:** The status bar at the bottom right says "Activate Windows Go to Settings to activate Windows."

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 uploading 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

Google Play Console

All apps

Dashboard

Inbox 2

Statistics

Publishing overview

Release

Releases overview

Production

Testing

Reach and devices

Overview

Device catalog

App bundle explorer

Setup

App integrity

Search Play Console

## App integrity

Protect your app and your users [Show more](#)

Integrity API responses off [Releases signed by Google Play](#)

Integrity API [App signing](#)

### App signing key certificate

This is the public certificate for the app signing key that Google uses to sign each of your releases. Use it to register your key with API providers. The app signing key itself is not accessible, and is kept on a secure Google server.

MD5 certificate fingerprint

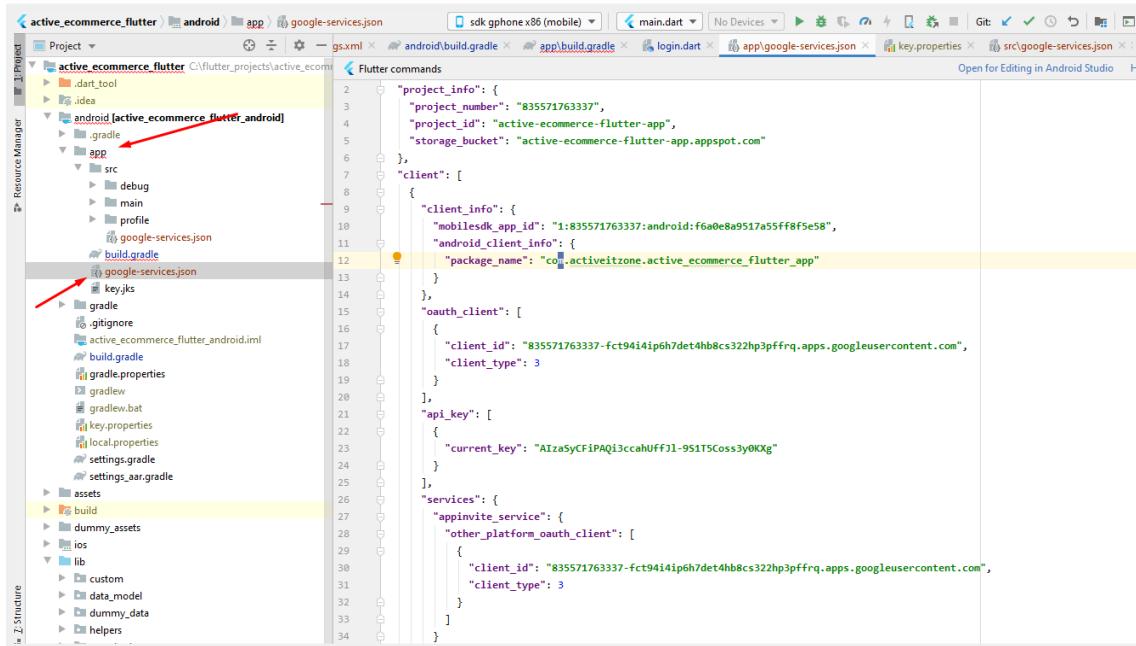
SHA-1 certificate fingerprint

SHA-256 certificate fingerprint

#### Upgrade your app signing key for new installs

You can upgrade your app signing key for new installs, for example if you want to move to a cryptographically stronger key. Google will use the new key to sign all new installs and their updates. Your legacy key will still be used to sign updates for users who already have your app.

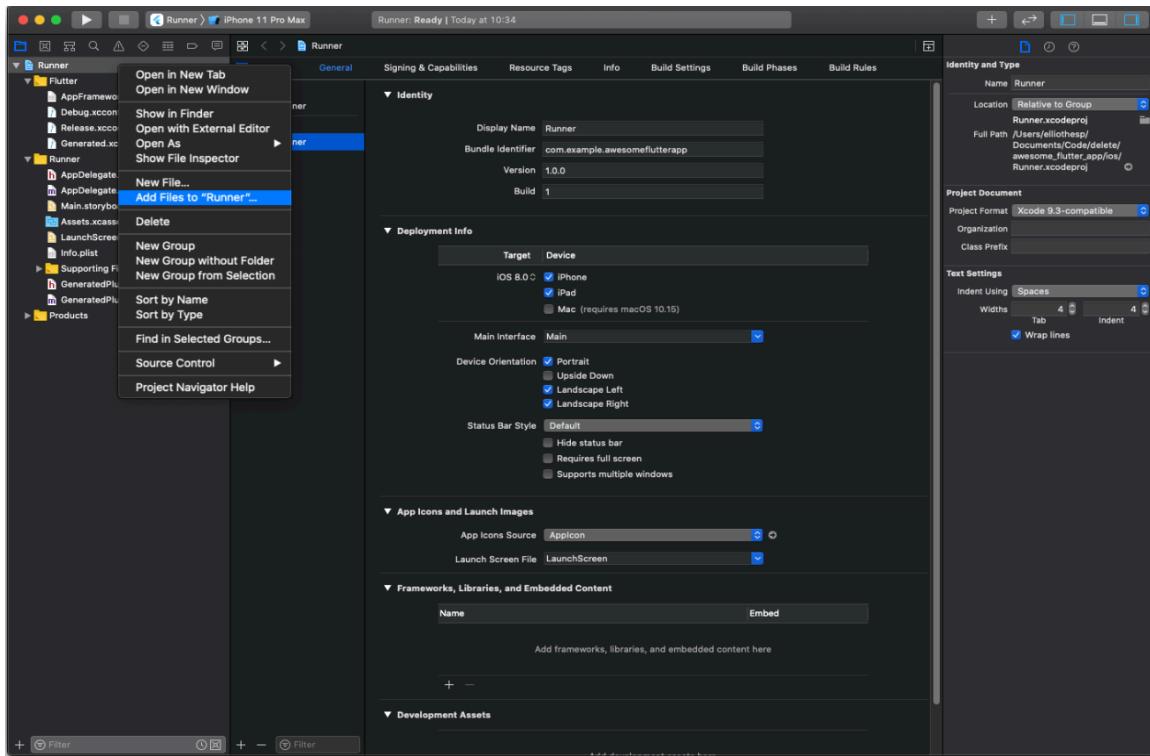
You must generate your own google-services.json. Do not use ours - it will not work for you.



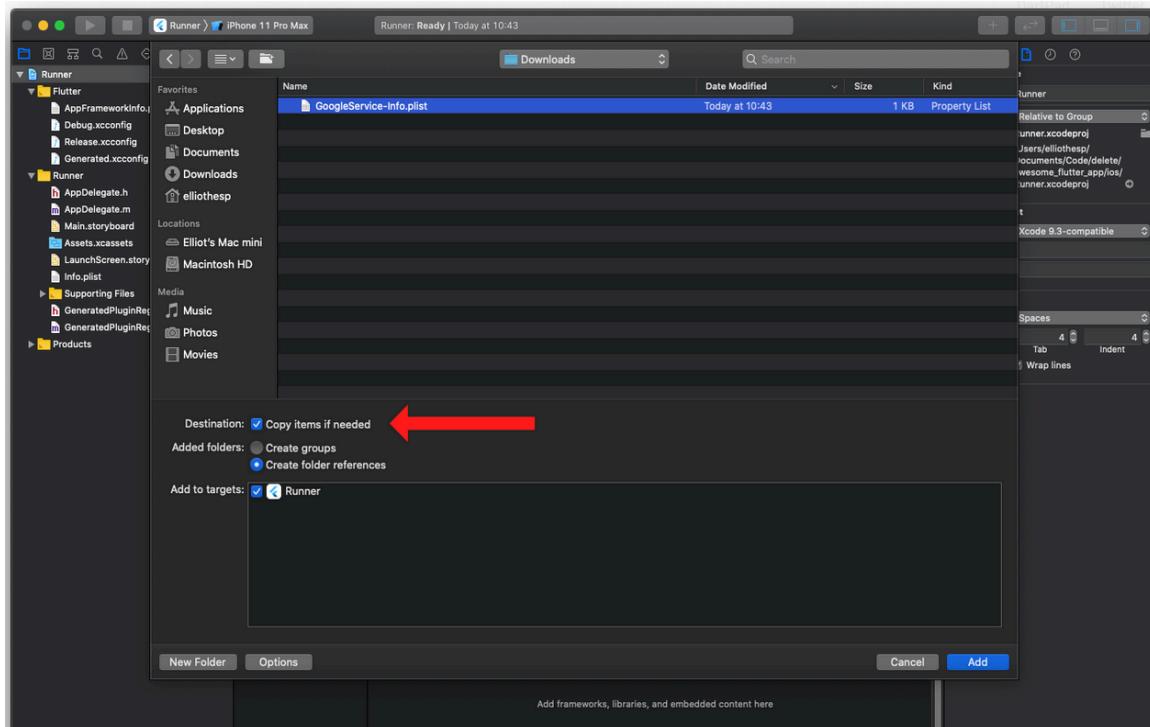
### Add the IOS app on your Firebase project

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.

The screenshot shows the Firebase console's 'Settings' page for an application named 'testapp'. The 'Basic' tab is selected in the sidebar. The main form contains the following data:

- App ID:** [REDACTED]
- App Secret:** [REDACTED] [Show](#)
- Display Name:** testapp
- Namespace:** [REDACTED]
- App Domains:** [REDACTED]
- Contact Email:** [REDACTED]
- Privacy Policy URL:** Privacy policy for Login dialog and App Details
- Terms of Service URL:** Terms of Service for Login dialog and App Details
- App Icon (1024 x 1024):** [Placeholder icon with a plus sign, 1024x1024]
- Category:** Choose a Category

At the bottom right, there is a 'Verification' section with a 'Start Verification' button, a link to 'Go to Settings', and 'Discard' and 'Save changes' buttons.

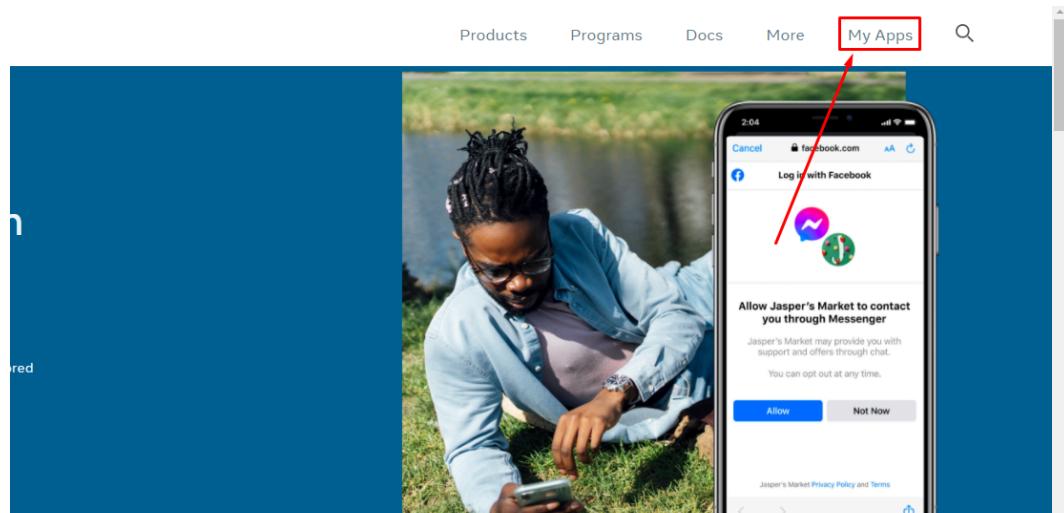
### I) Facebook: Package Used

[https://pub.dev/packages/flutter\\_facebook\\_auth](https://pub.dev/packages/flutter_facebook_auth)

See its documentation and steps

How to create a Facebook app?

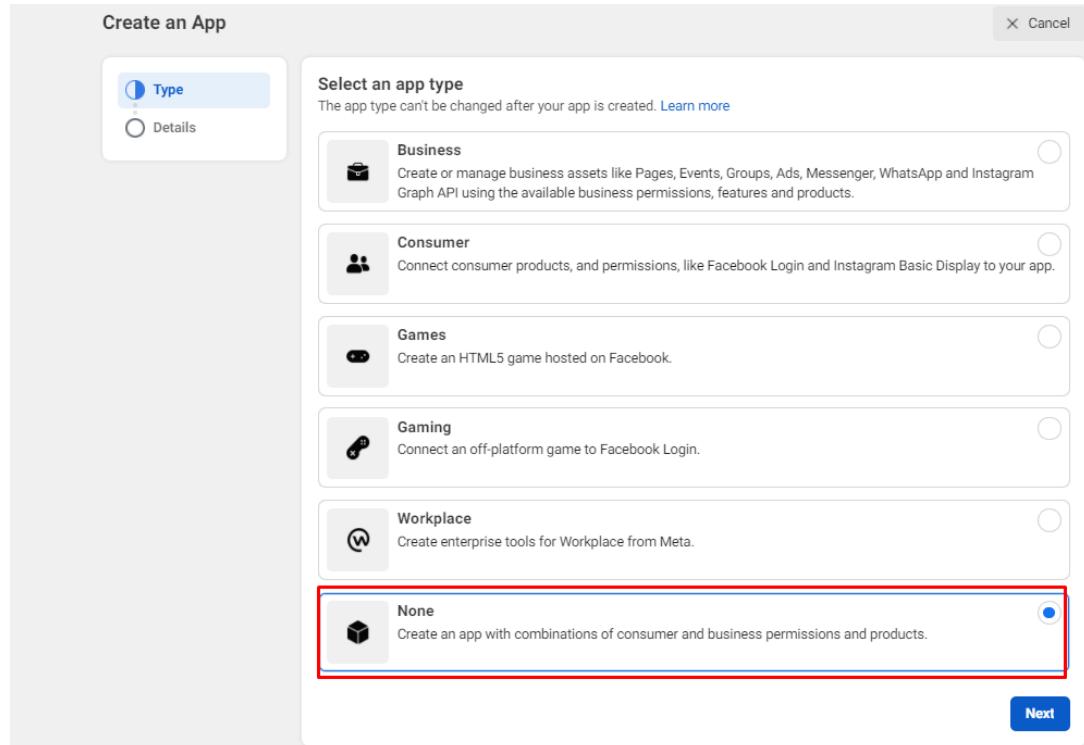
- A) First you have to log in to Facebook in your browser.
- B) Goto <https://developers.facebook.com/>
- C) Goto My Apps.



D) Create an app.

A screenshot of the Meta developer portal. The top navigation bar has tabs for 'Products', 'Programs', 'Docs', 'More', and 'My Apps'. A red box highlights the 'My Apps' tab. Below the navigation is a search bar and a 'Create App' button, which is also highlighted with a red box and an arrow. The main content area shows a grid of app icons. At the bottom, there are sections for 'Products', 'Programs', 'Support', 'News', and 'Terms and Policies'. A sidebar on the left shows 'Follow Us' with social media links and a 'Recently Used' dropdown.

Select App Type None.



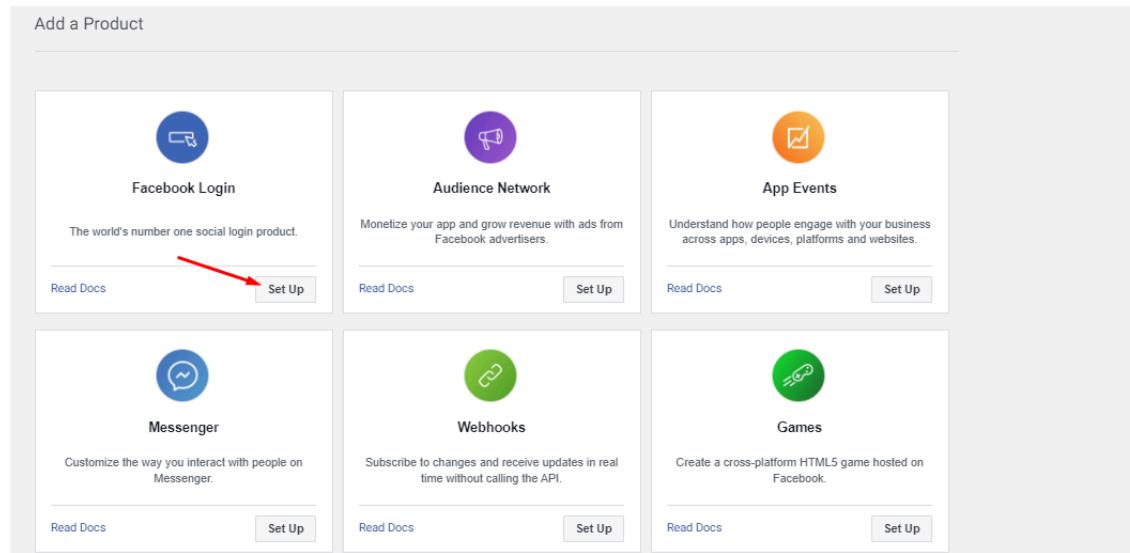
#### E) Add your app details

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

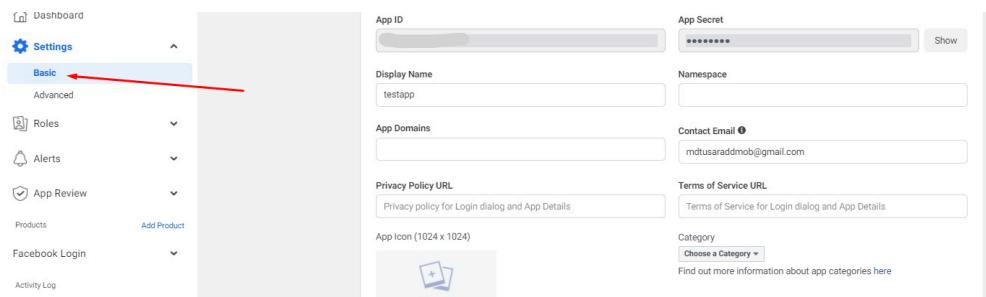
The screenshot shows the 'Create an App' interface in the 'Details' step. It includes fields for 'App Name' (with a red arrow pointing to the input field), 'App Contact Email' (with a red arrow pointing to the input field), and 'Business Account' (with a red arrow pointing to the dropdown menu which contains the text 'Don't change this option'). At the bottom, there's a note about agreeing to terms, a 'Previous' button, and a prominent green 'Create App' button.

#### F) Add product "Facebook Login".

- ❖ Click the "Set Up" button.



#### G) Click the “Basic” Option.



#### H) Add some information and save it.

- This app privacy policy URL.
- This app terms of service URL.
- This app icon (Icon size also 1024 X 1024 Or 512 X 512 ).
- This app category.

Display Name: testapp

Namespace:

App Domains:

Contact Email: mdtusaraddmob@gmail.com

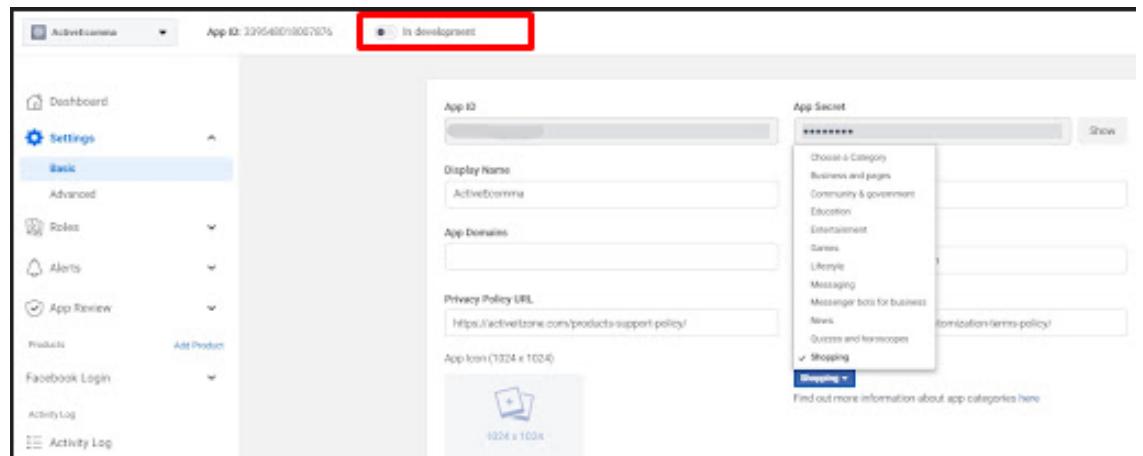
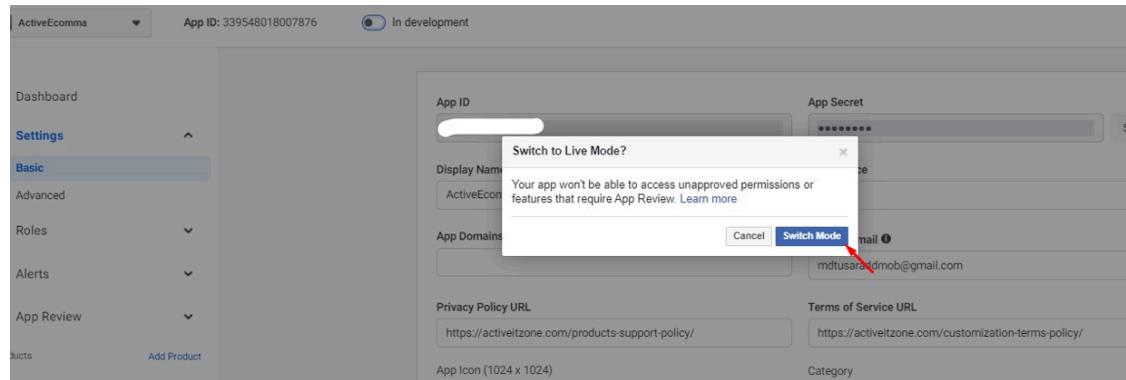
Privacy Policy URL: Privacy policy for Login dialog and App Details

Terms of Service URL: Terms of Service for Login dialog and App Details

App Icon (1024 x 1024):

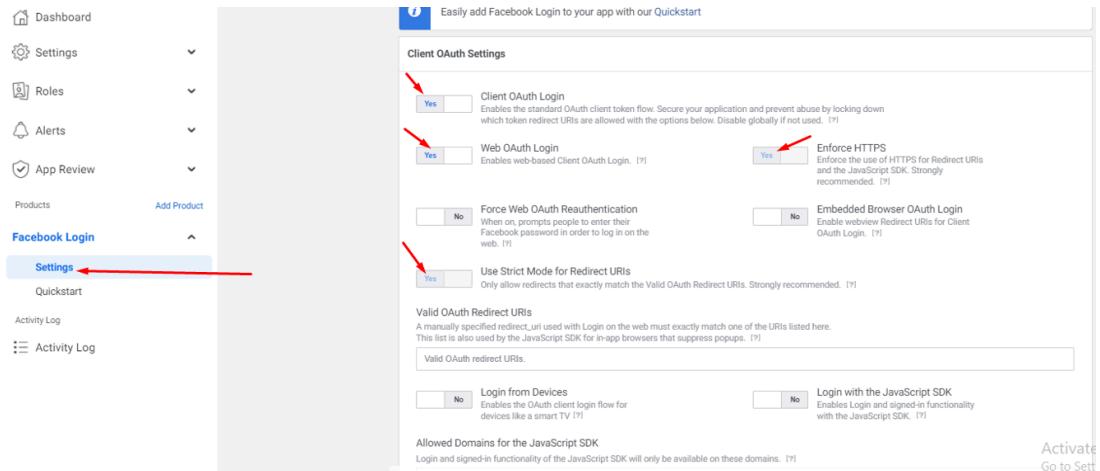
Category: Choose a Category

- I) Now activate your app.
- J) Click the "Switch Mode" Button.



K) 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



## Integration

### For Android:

L) 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.

The screenshot shows the Android project structure on the left and the `strings.xml` file content on the right. Red arrows point to several key locations:

- `src/main/java`
- `src/main/kotlin`
- `src/main/res`
- `src/main/values`
- `src/main/values/styles.xml`
- `src/main/values/styles.xml` (another instance)
- `src/main/AndroidManifest.xml`
- `strings.xml` (highlighted in blue)

```

<resources>
    <string name="app_name">ActiveCommerce</string>

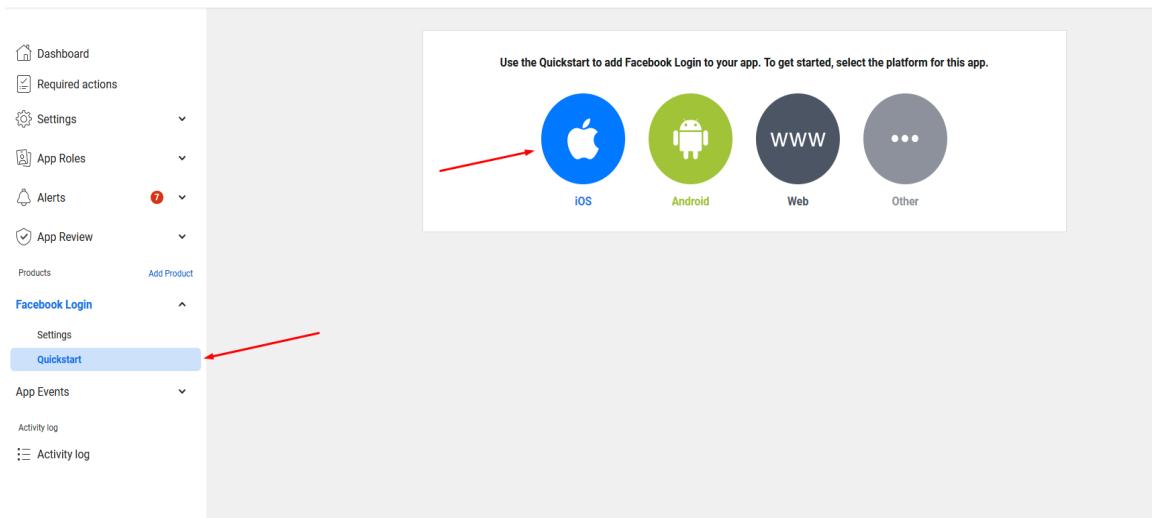
    <!-- Replace "000000000000" with your Facebook App ID here. -->
    <string name="facebook_app_id">708[REDACTED]30</string>

    <!-- Replace "000000000000" with your Facebook App ID here.
        **NOTE:** The scheme needs to start with 'fb' and then your ID.
    -->
    <string name="fb_login_protocol_scheme">fb708[REDACTED]30</string>
</resources>

```

### For iOS:

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



Then choose your environment SDK: Cocoapods

**1. Set up Your Development Environment**

**i** To make your app compatible with iOS 11, be sure to use the latest Facebook SDKs for iOS. If you link to the SDKs with CocoaPods, you must update your pods for the SDKs your app uses and recompile your app. You can also download the latest version of the Facebook iOS SDK, integrate it into your app, and recompile.

Set up your development environment before using Facebook Login for iOS.

SDK: Cocoapods ▾

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

**1. Set up Your Development Environment**

**2. Add your Bundle Identifier**

The bundle identifier (Bundle ID) should appear in the box below. If the box is empty, find your bundle identifier in your Xcode Project's iOS Application Target and paste it into the box below.

Bundle ID

You can change your bundle identifier in the future via the iOS section on the settings page.

com.facebook.samples.ios.quickstart

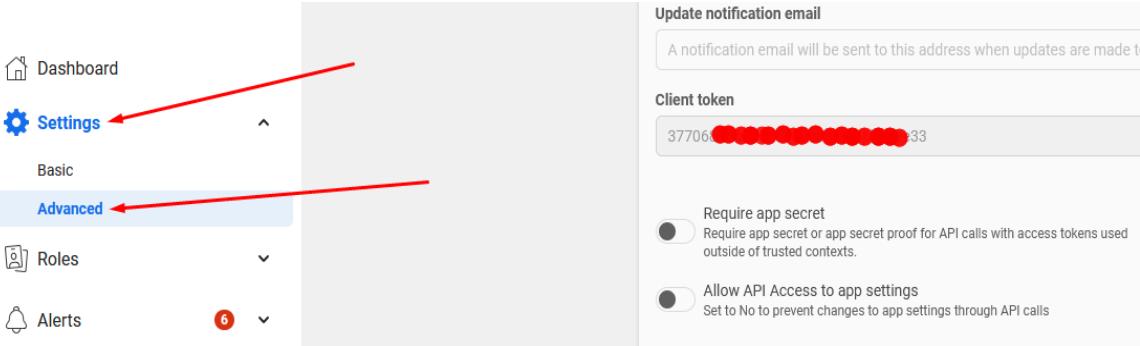
Save

Back Continue

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 find 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.

```

source_code [active_ecommerce_flutter] C:\Users\Use\Desktop\main\source_code
> .dart_tool
> .idea
> android
> assets
> build
> dummy_assets
> ios
> Flutter
> Assets.xcassets
> Base.lproj
> AppDelegate.swift
> GeneratedPluginRegistrant.h
> GeneratedPluginRegistrant.m
> Info.plist
> Runner-Bridging-Header.h
> Runner.xcodeproj
> Runner.xcworkspace
> .gitignore
> lib
> test
> .flutter-plugins
> .flutter-plugins-dependencies
> .gitattributes
> .gitignore
> .metadata
> .packages
> FlutterCommerceAPI.postman_collection.json
> how_to_access_from_android_device_as_emulator.txt
> 10n.yaml
> pubspec.lock
> pubspec.yaml
> README.md
> sha-1
> External Libraries
> Scratches and Consoles

Info.plist
<key>NSLocationAlwaysUsageDescription</key>
<string>User can set address using phone location</string>

<key>NSLocationWhenInUseUsageDescription</key>
<string>location when in use</string>

<key>NSBluetoothPeripheralUsageDescription</key>
<string>This app uses Bluetooth to connect with wearables.</string>

<key>NSBluetoothAlwaysUsageDescription</key>
<string>This app uses Bluetooth to connect with wearables.</string>

<!-- Facebook Login configuration -->
<key>CFBundleURLTypes</key>
<array>
    <dict>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>fbxxxxxxxx</string> App Id
        </array>
    </dict>
</array>
<key>FacebookAppID</key>
<string>xxxxxxxx</string> Client Token
<key>FacebookClientToken</key>
<string>xxxxxxxxxxxxxx</string>
<key>FacebookDisplayName</key>
<string>Your App Name</string> App Name

<key>LSApplicationQueriesSchemes</key>
<array>
    <string>fbapi</string>
    <string>fb-messenger-share-api</string>
    <string>fbauth2</string>
    <string>fbshareextension</string>
</array>
</dict>
</plist>

```

**Google:** Package Used

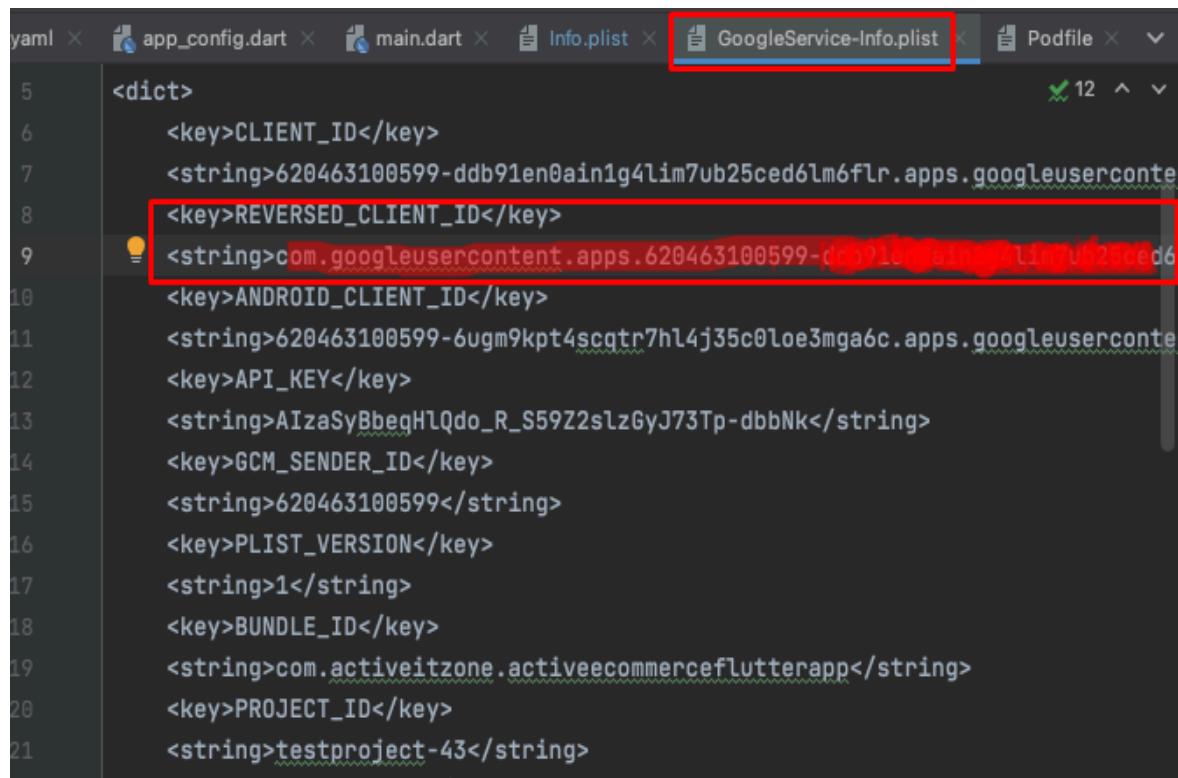
[https://pub.dev/packages/google\\_sign\\_in](https://pub.dev/packages/google_sign_in)

# Integration

For IOS:

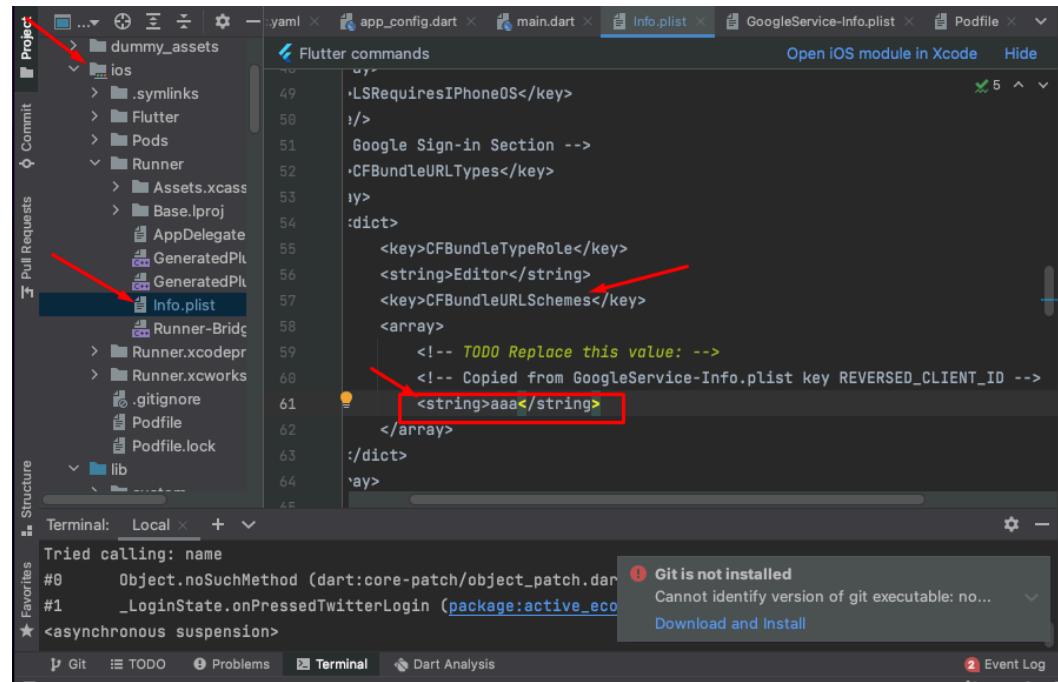
1. [First register your application.](#)
2. Make sure the file you download in step 1 is named GoogleService-Info.plist.
3. Move or copy GoogleService-Info.plist into the [my\_project]/ios/Runner directory.
4. Open Xcode, then right-click on Runner directory and select Add Files to "Runner".
5. Select 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**



```
yaml × app_config.dart × main.dart × Info.plist × GoogleService-Info.plist × Podfile × ✓ 12 ▲ ▼  
5 <dict>  
6     <key>CLIENT_ID</key>  
7     <string>620463100599-ddb91en0ain1g4lim7ub25ced6lm6flr.apps.googleusercontent.com</string>  
8     <key>REVERSED_CLIENT_ID</key>  
9     <string>com.googleusercontent.apps.620463100599-ddb91en0ain1g4lim7ub25ced6lm6flr</string>  
10    <key>ANDROID_CLIENT_ID</key>  
11    <string>620463100599-6ugm9kpt4scqtr7hl4j35c0loe3mga6c.apps.googleusercontent.com</string>  
12    <key>API_KEY</key>  
13    <string>AIzaSyBbegHlQdo_R_S59Z2s1zGyJ73Tp-dbbNk</string>  
14    <key>GCM_SENDER_ID</key>  
15    <string>620463100599</string>  
16    <key>PLIST_VERSION</key>  
17    <string>1</string>  
18    <key>BUNDLE_ID</key>  
19    <string>com.activeitzone.activeecommerceflutterapp</string>  
20    <key>PROJECT_ID</key>  
21    <string>testproject-43</string>
```

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](https://pub.dev/packages/google_sign_in)

**Twitter** : package used

[https://pub.dev/packages/twitter\\_login](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.

**Developer Portal**

Dashboard

Projects & Apps

Overview

Elevated

+ New Project

Products NEW

Account

Standalone Apps V1.1 ACCESS

NO PROJECTS HERE

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

1 Project name

Your Project helps you organize your work and monitor your usage with the Twitter API.

Your project name

2 Use case

This is how you intend to use the Twitter developer platform.

Building consumer tools (professional)

Select a use case

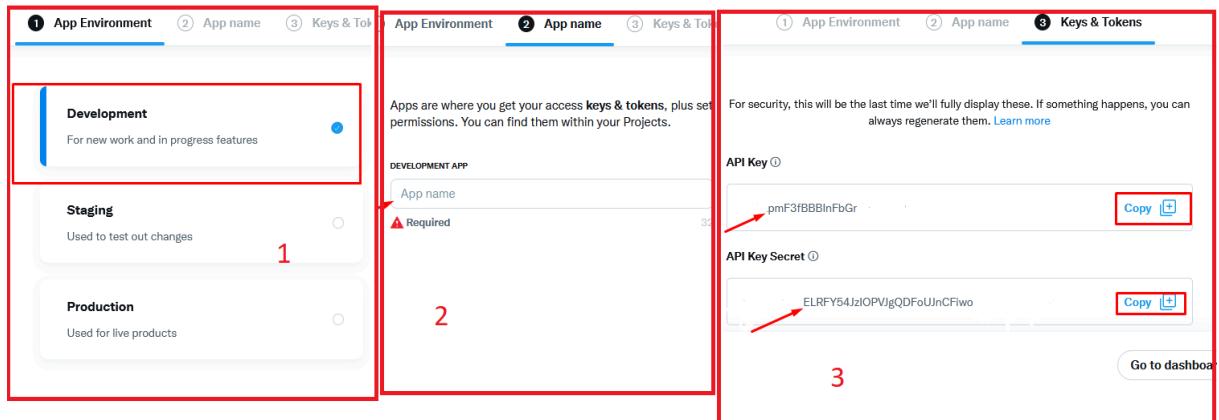
- Doing academic research
- Making a bot
- Build customized solutions in-house
- Embedding Tweets in a website
- Exploring the API
- Doing something else
- Publishing ads programmatically
- Student
- Teaching
- Building tools for businesses
- Building consumer tools (hobbyist)
- Building consumer tools (professional)

3 Project description

⚠ Required

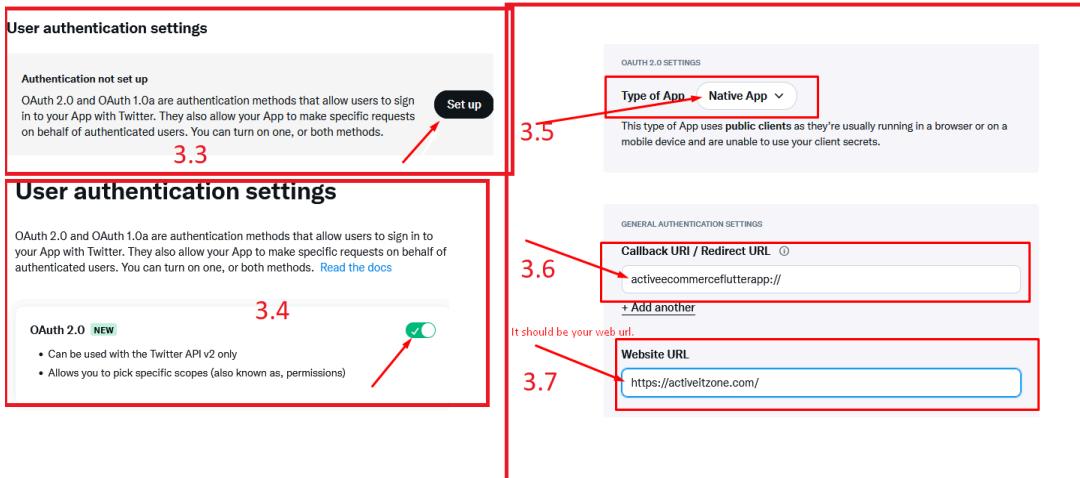
## 2. Set up your app

- 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.

```

active_ecommerce_flutter lib social_config.dart
1 class SocialConfig{
2     var twitter_consumer_secret = <your consumer key>;
3     var twitter_consumer_key = <your consumer secret>;
4 }
5

```

**Apple SignIn:** package used

[https://pub.dev/packages/sign\\_in\\_with\\_apple](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.

## 11. How to configure push notifications?

**Answer:**

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;

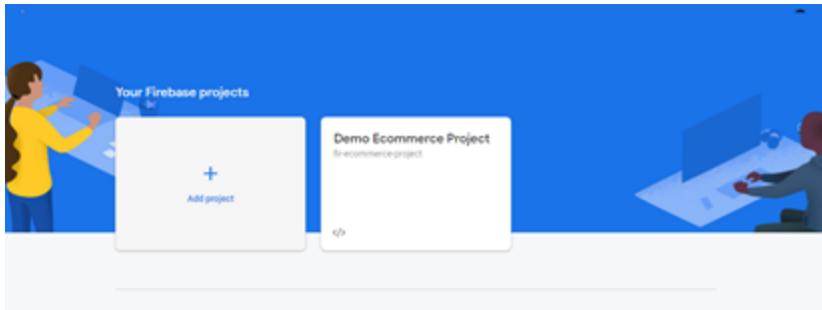
The screenshot shows the Android Studio interface with the project tree on the left and the code editor on the right. The project tree shows a folder structure: .dart\_tool, .idea, android, assets, build, dummy\_assets, ios, lib (which is highlighted with a red box), custom, data\_model, dummy\_data, helpers, I10n, middlewares, presenter, providers, repositories, screens, services, ui\_elements, ui\_sections, app\_config.dart, lang\_config.dart, main.dart, my\_theme.dart, other\_config.dart (which is highlighted with a red box), and social\_config.dart. The code editor displays the 'other\_config.dart' file with the following content:

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

A red arrow points from the 'lib' folder in the project tree to the 'other\_config.dart' file in the code editor.

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

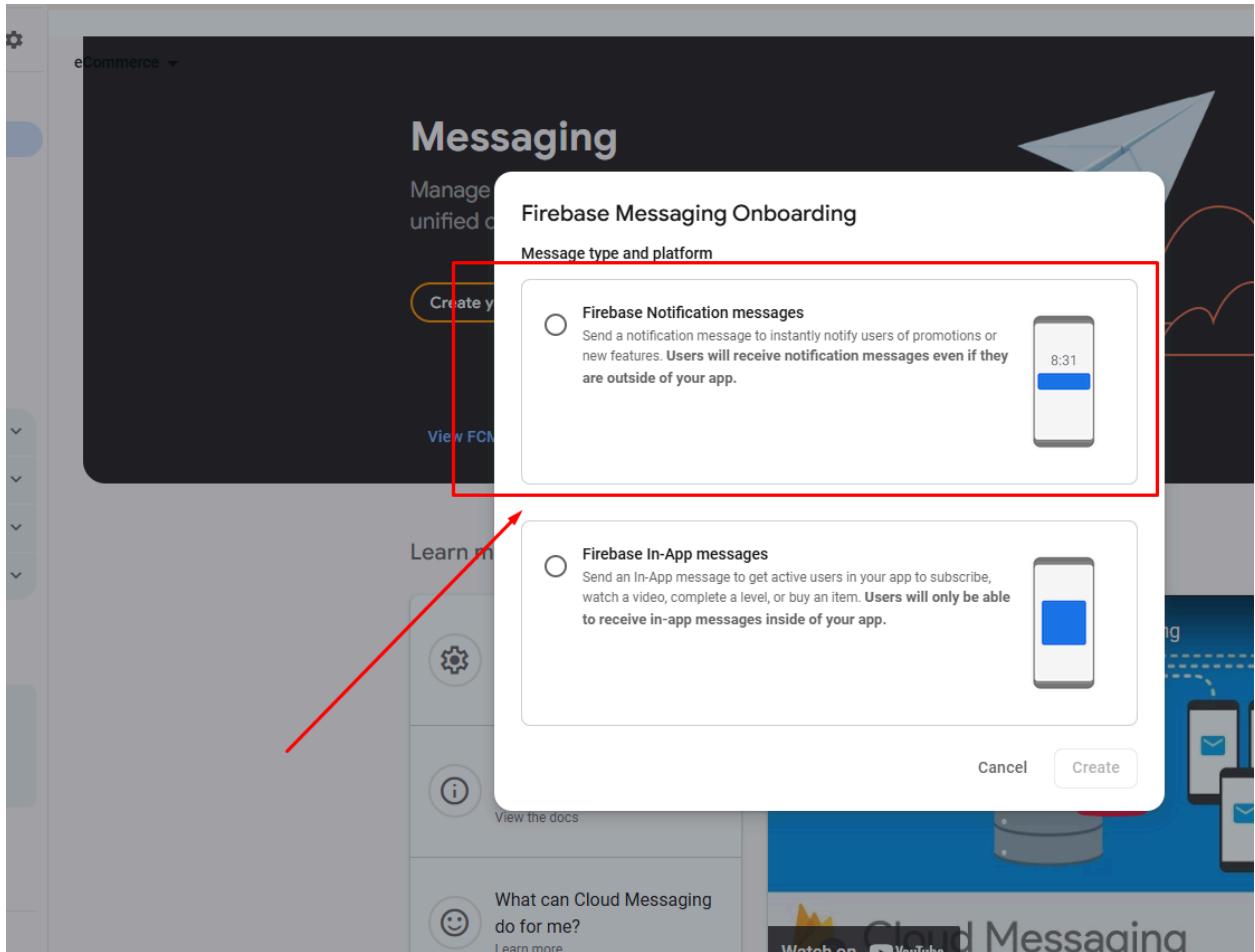
If you already have a project then continue with that.



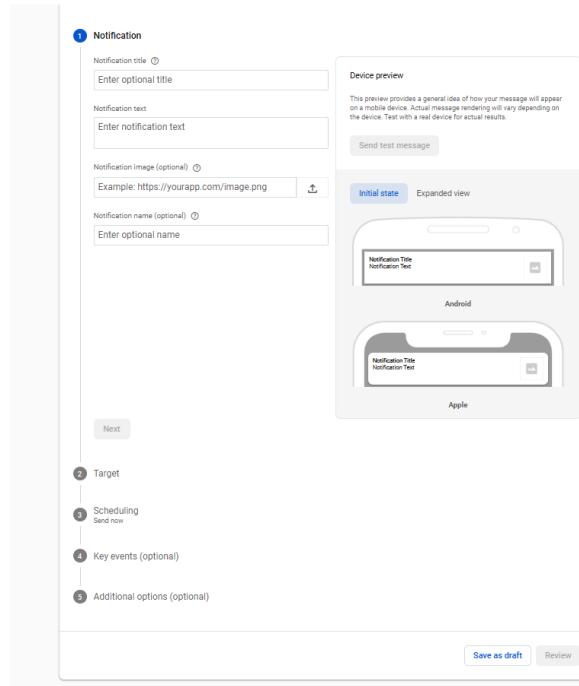
3. Go to project Overview and scroll down and find the Cloud Messaging

The screenshot shows the Firebase Project Overview page for a project named 'flutterFireNotes'. The left sidebar includes links for Project shortcuts, Firestore Database, Messaging, Genkit, Vertex AI, and other product categories like Build, Run, Analytics, and AI. A red arrow points from the 'Messaging' link in the sidebar to the 'Cloud Messaging' section in the main content area. The 'Cloud Messaging' section is highlighted with a red box and contains the following text: 'Cross-platform push messaging infrastructure between your servers and user devices'. To the right of the main content, there are sections for Crashlytics, Performance, and Remote Config, along with a 'See all Run features' link.

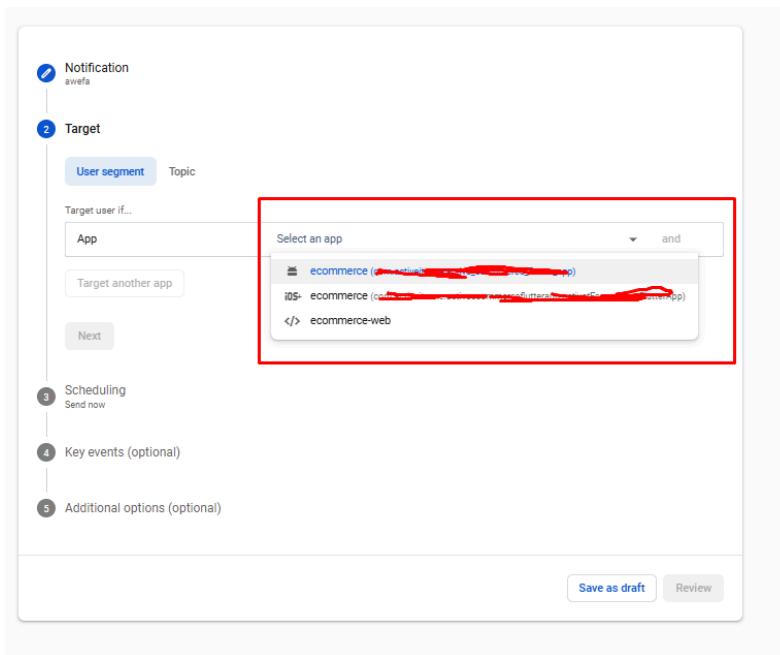
4. Click the Cloud Messaging and create your first campaign . then click Firebase Notification Messages and click the create button .



5. Write your title , Notification text and other optional.you will see your Next Button is now visible then click it



After Click the next button you have to select your app and follow the screenshot



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

The screenshot shows the Android Studio interface with the project 'active\_ecommerce\_flutter' open. The left sidebar displays the project structure, including the 'app' module which contains the 'google-services.json' file. A red arrow points to this file. The main editor area shows the JSON content of the 'google-services.json' file:

```

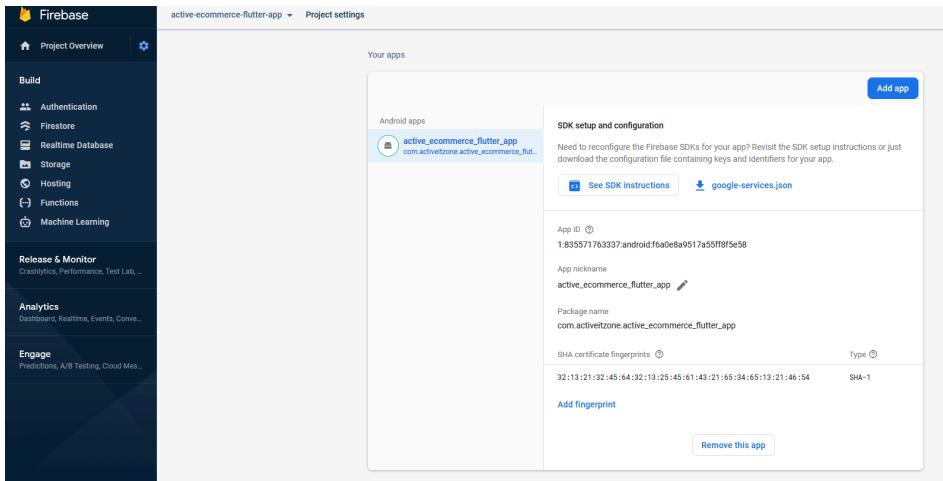
{
  "project_info": {
    "project_number": "835571763337",
    "project_id": "active-ecommerce-flutter-app",
    "storage_bucket": "active-ecommerce-flutter-app.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:835571763337:android:f6a0e8a9517a55ff8f5e58",
        "android_client_info": {
          "package_name": "co.activeitzone.active_ecommerce_flutter_app"
        }
      },
      "oauth_client": [
        {
          "client_id": "835571763337-fct9414ip6h7det4hb8cs322hp3pffrq.apps.googleusercontent.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key": "AIzaSyCFiPAQj3ccahUffJl-951T5Coss3y0KXg"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client": [
            {
              "client_id": "835571763337-fct9414ip6h7det4hb8cs322hp3pffrq.apps.googleusercontent.com",
              "client_type": 3
            }
          ]
        }
      }
    ],
    "configuration_version": "1"
  }
}

```

## Firebase console:

<https://console.firebaseio.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.

```
C:\Program Files\Android\Android Studio\jre\bin>keytool -list -v -keystore C:\flutter_projects\active_ecommerce_flutter\key.jks -alias key -storepass 123456 -keypass 123456
Alias name: key
Creation date: Apr 1, 2021
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Test, OU=Test, O=Test, L=Test, ST=Test, C=US
Issuer: CN=Test, OU=Test, O=Test, L=Test, ST=Test, C=US
Serial number: 656ab3f5
Valid from: Thu Apr 01 21:56:13 BDT 2021 until: Mon Aug 17 21:56:13 BDT 2048
Certificate fingerprints:
    MD5: B4:60:55:48:8F:D0:E1:1B:43:2D:76:3D:99:1A:D0:B8
    SHA1: B1:3A:53:CF:F8:9A:07:17:1F:9B:6E:14:8E:24:69:7C:EC:03:D5:2F
    SHA256: 92:09:F0:BF:56:F8:14:AB:AD:C8:C6:43:1D:79:FA:3F:66:2E:D8:2D:66:FD:5F:BE:00:10:88:06:FA:37:46:A1
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3
Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
```

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

## Apple Integration

Please Follow this URL's documentation

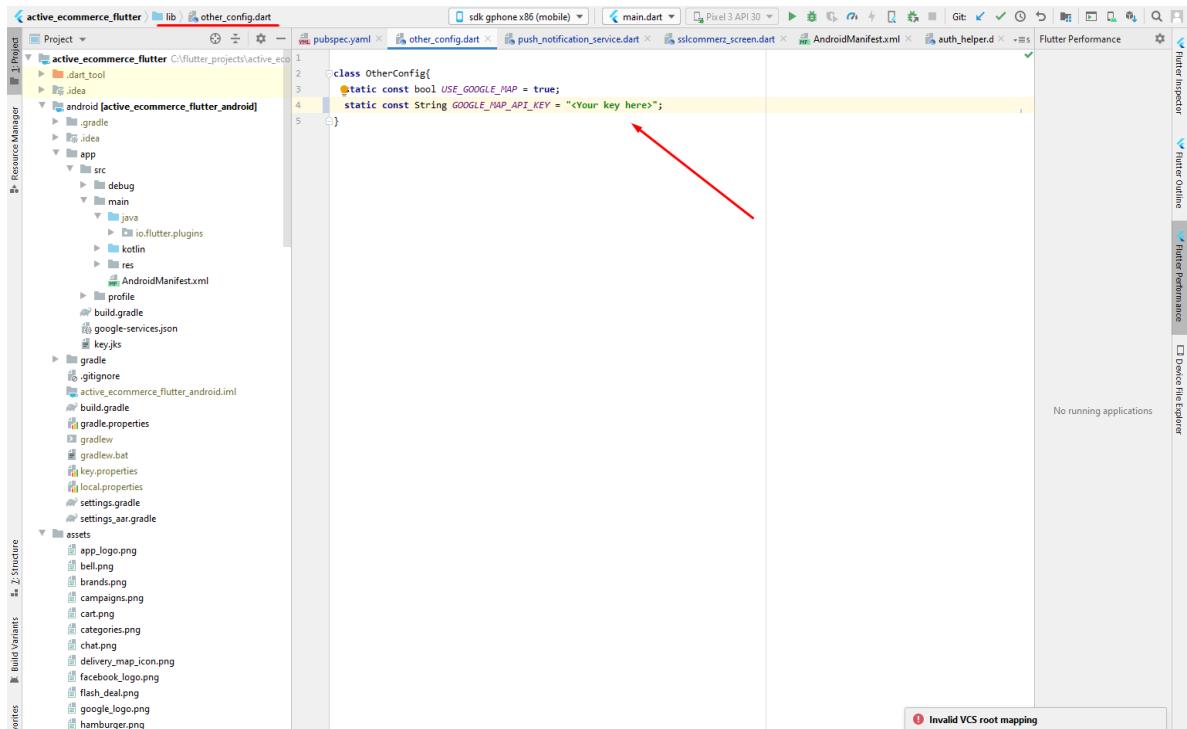
<https://firebase.flutter.dev/docs/messaging/apple-integration/>

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

Answer:

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

2. In lib/other\_config.dart make, use google map = true and put google map api key



The screenshot shows the Android Studio interface with the project 'active\_ecommerce\_flutter' open. The 'lib' directory contains the file 'other\_config.dart'. The code in this file is:

```
1 class OtherConfig{  
2     static const bool USE_GOOGLE_MAP = true;  
3     static const String GOOGLE_MAP_API_KEY = "<Your key here>";  
4 }  
5 }
```

A red arrow points from the text '<Your key here>' to the right, indicating where the user should enter their Google Map API key.

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

."/>

```

<manifest>
    ...
    <application>
        ...
        <activity android:name="io.flutter.embedding.android.FlutterActivity" />
        ...
    </application>
</manifest>

```

The screenshot shows the AndroidManifest.xml file in the Android Studio project structure. The manifest file contains code for Flutter embedding and a meta-data entry for Google Maps API key.

```

<manifest>
    ...
    <application>
        ...
        <activity android:name="io.flutter.embedding.android.FlutterActivity" />
        ...
    </application>
</manifest>

```

A red arrow points to the line:

```

<meta-data android:name="com.google.android.geo.API_KEY" android:value="Your map api key"/>

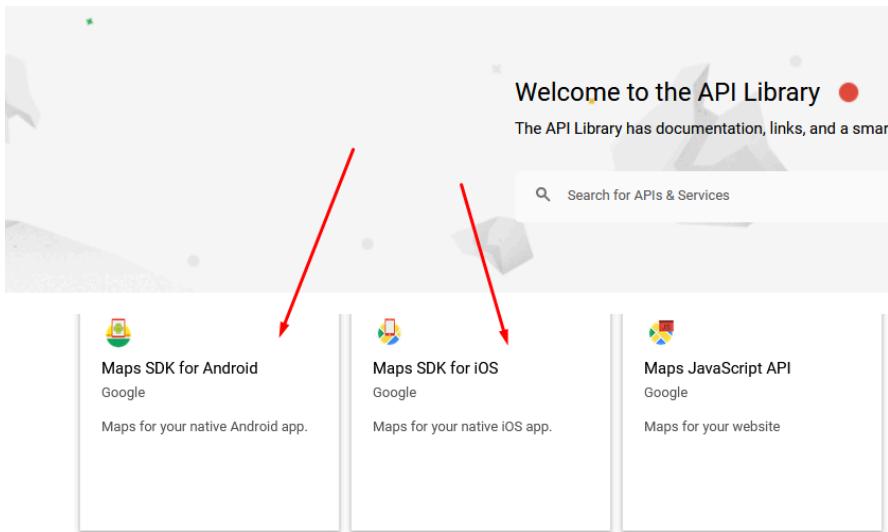
```

A tooltip message "Invalid VCS root mapping" is visible at the bottom right.

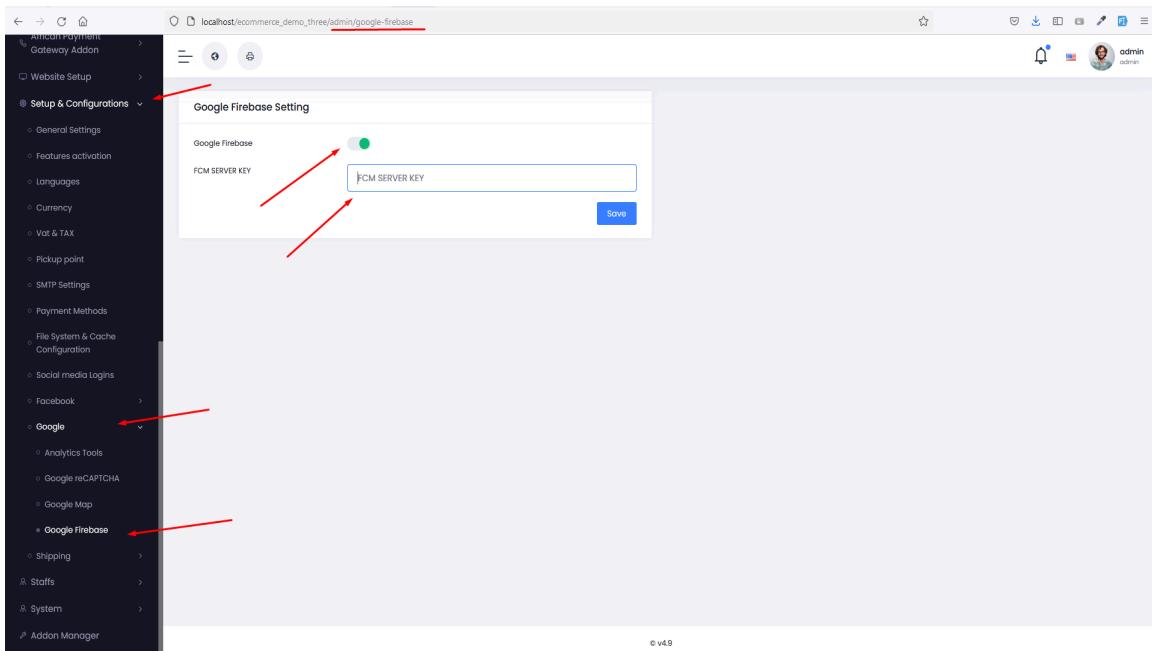
#### 4. For ios follow this

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

#### 5. Enable Android and ios api. These are free.



Machine learning



6. 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



### Geocoding API

Google Enterprise API

Convert between addresses and geographic coordinates.

[MANAGE](#)

API Enabled



### Places API

Google Enterprise API

Get detailed information about 100 million places

[MANAGE](#)

API Enabled

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

### Answer:

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;

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 codecanyon

    //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	default_language	mobile_app_code	app_language rtl	Options
1	English	en	en		<input type="checkbox"/>	<span>EN</span> <span>ES</span> <span>FR</span> <span>DE</span> <span>IT</span> <span>PT</span> <span>NL</span> <span>GR</span>
2	Bangla	bd		bn	<input type="checkbox"/>	<span>EN</span> <span>ES</span> <span>FR</span> <span>DE</span> <span>IT</span> <span>PT</span> <span>NL</span> <span>GR</span>
3	Arabic	sa		ar	<input checked="" type="checkbox"/>	<span>EN</span> <span>ES</span> <span>FR</span> <span>DE</span> <span>IT</span> <span>PT</span> <span>NL</span> <span>GR</span>

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

**Answer:**

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](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.

#	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"/>	
4	Spain	es		<input type="checkbox"/>	
5	Japan	jp		<input type="checkbox"/>	
6	France	fr	fr	<input type="checkbox"/>	
7	India	in		<input type="checkbox"/>	

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.

#	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"/>	
4	Spain	es		<input type="checkbox"/>	
5	Japan	jp		<input type="checkbox"/>	
6	France	fr	fr	<input type="checkbox"/>	
7	India	in		<input type="checkbox"/>	
8	Netherlands	nl		<input type="checkbox"/>	
9	Afghanistan	af		<input type="checkbox"/>	

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.

#	Name	Code	Flutter App Long 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"/>	
4	Spain	es		<input type="checkbox"/>	
5	Japan	jp		<input type="checkbox"/>	
6	France	fr	fr	<input type="checkbox"/>	
7	India	in		<input type="checkbox"/>	
8	Netherland	nl		<input type="checkbox"/>	
9	Afghanistan	af		<input type="checkbox"/>	
10	Egypt	eg		<input type="checkbox"/>	

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.

## 15. How to configure deeplink?

Go to your Androidmanifest.xml file which is in app>src>main>androidmanifest.xml

Set your **domain.com** name in **android:host**

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. Red arrows point from the top of the project tree to the 'app' folder, then to the 'src' folder, then to the 'main' folder, and finally to the 'AndroidManifest.xml' file. The code editor displays the XML configuration for the manifest. A red box highlights the line containing the 'host' attribute: '<data android:scheme="http" android:host="domain.com"/>'. This line is part of an intent filter definition.

```
45 <provider android:name="io.flutter.plugins.workManagerInitializer">
46     android:value="androidx.startup"
47     tools:node="remove" />
48 </provider>
49
50 <!-- declare customized Initializer -->
51 <provider android:name="vn.hunghd.flutterdownloader.FlutterDownloaderInitializer" android:authorities="${applicationId}.flutter-downloader-init" android:exported="false">
52     <!-- changes this number to configure the maximum number of concurrent tasks -->
53     <meta-data android:name="vn.hunghd.flutterdownloader.MAX_CONCURRENT_TASKS" android:value="5" />
54 </provider>
55 <!-- End FlutterDownloader customization -->
56
57 <activity android:name=".MainActivity" android:exported="true" android:launchMode="singleTask" android:theme="@style/LaunchTheme" android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|colorScheme" android:hardwareAccelerated="true" android:windowSoftInputMode="adjustResize">
58     <!-- Specifies an Android theme to apply to this Activity as soon as the Android process has started. This theme is visible to the user while the Flutter UI initializes. After that, this theme continues to determine the Window background behind the Flutter UI. -->
59
60     <meta-data android:name="flutter_deeplinking_enabled" android:value="true" />
61     <intent-filter android:autoVerify="true">
62         <action android:name="android.intent.action.VIEW" />
63         <category android:name="android.intent.category.DEFAULT" />
64         <category android:name="android.intent.category.BROWSABLE" />
65         <data android:scheme="http" android:host="domain.com"/>
66         <data android:scheme="https"/>
67     </intent-filter>
68 
```

**N.B:**

Deeplink will only work over these routes.

```
1  deeplink paths...
2  /,
3  customer_products,
4  customer-products,
5  customer-product/:slug,
6
7  product/:slug,
8
9  customer-packages,
10
11 auction_product_bids,
12 auction-products,
13 auction-product/:slug,
14 auction/purchase_history,
15
16 users/login,
17 users/registration,|
18 dashboard,
19
20 brand/:slug,
21 brands,
22 cart,
23 categories,
24 category/:slug,
25 flash-deals,
26
27 flash-deal/:slug,
28 followed-seller,
29
30 purchase_history,
31 purchase_history/details/:id,
32 sellers,
33 shop/:slug,
34 todays-deal,
35
36 coupons,
```

## 16. How to generate Play Store uploadable files for release?

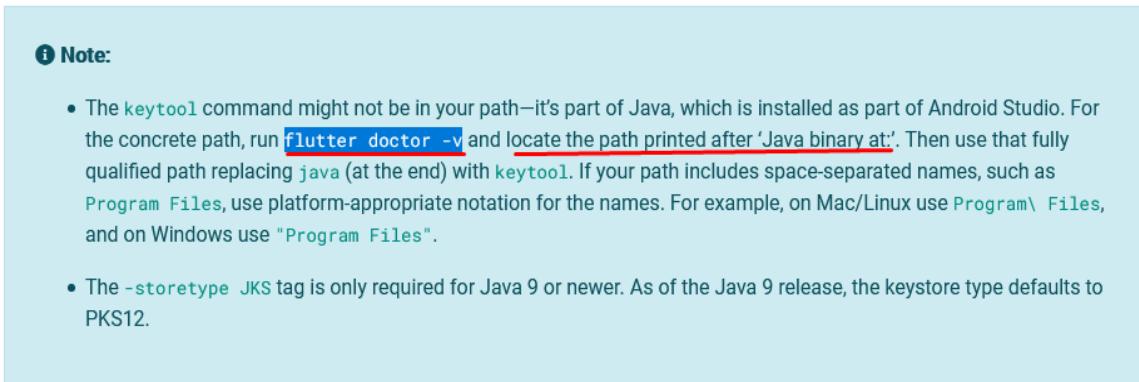
**Answer:**

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

Siging 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:



### Setup environment for keytool:

- Run this command in your system terminal: `flutter doctor -v`
- Find the Java binary at: copy this path except the "java" folder.
- 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**
- Run the keytool.exe file. Example: **C:\Program Files\Android\Android Studio\jre\bin>keytool.exe**

```
C:\Users\User>flutter doctor -v
[!] flutter (Channel stable, 3.0.2, on Microsoft Windows [Version 10.0.19044.1889], locale en-US)
  • Flutter version 3.0.2 at C:/flutter_sdk/flutter
  • Upstream repository https://github.com/flutter/flutter.git
  • Framework revision cd41fdd495 (3 months ago), 2022-06-08 09:52:13 -0700
  • Engine revision f15fb824b57
  • Dart version 2.17.3
  • DevTools version 2.12.2

[!] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
  • Android SDK at C:\Users\User\AppData\Local\Android\sdk
  • Platform android-32, build-tools 32.1.0-rc1
  • Java binary at: C:\Program Files\Android\Android Studio\jre\bin\java (Copy this path)
  • Java version OpenJDK Runtime Environment (build 11.0.12+7-b1504.28-7817840)
  ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses

[?] Chrome - develop for the web
  • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

[X] Visual Studio - develop for Windows
  X Visual Studio not installed; this is necessary for Windows development.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components

[?] Android Studio (version 2021.2)
  • Android Studio at C:\Program Files\Android\Android Studio
  • Flutter plugin can be installed from:
    https://plugins.jetbrains.com/plugin/9212-flutter
  • Dart plugin can be installed from:
    https://plugins.jetbrains.com/plugin/6351-dart
  • Java version OpenJDK Runtime Environment (build 11.0.12+7-b1504.28-7817840)

[?] VS Code (version 1.70.2)
  • VS Code at C:\Users\User\AppData\Local\Programs\Microsoft VS Code
  • Flutter extension can be installed from:
    https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter

[?] Connected device (4 available)
  • sdk gphone64 x86_64 (mobile)    • emulator-5554    • android-x64      • Android 12 (API 31) (emulator)
  • Windows (desktop)               • windows        • windows-x64     • Microsoft Windows [Version 10.0.19044.1889]
  • Chrome (web)                  • chrome         • web-javascript • Google Chrome 104.0.5112.102
  • Edge (web)                     • edge          • web-javascript • Microsoft Edge 104.0.1293.54

[?] HTTP Host Availability
  • All required HTTP hosts are available
```

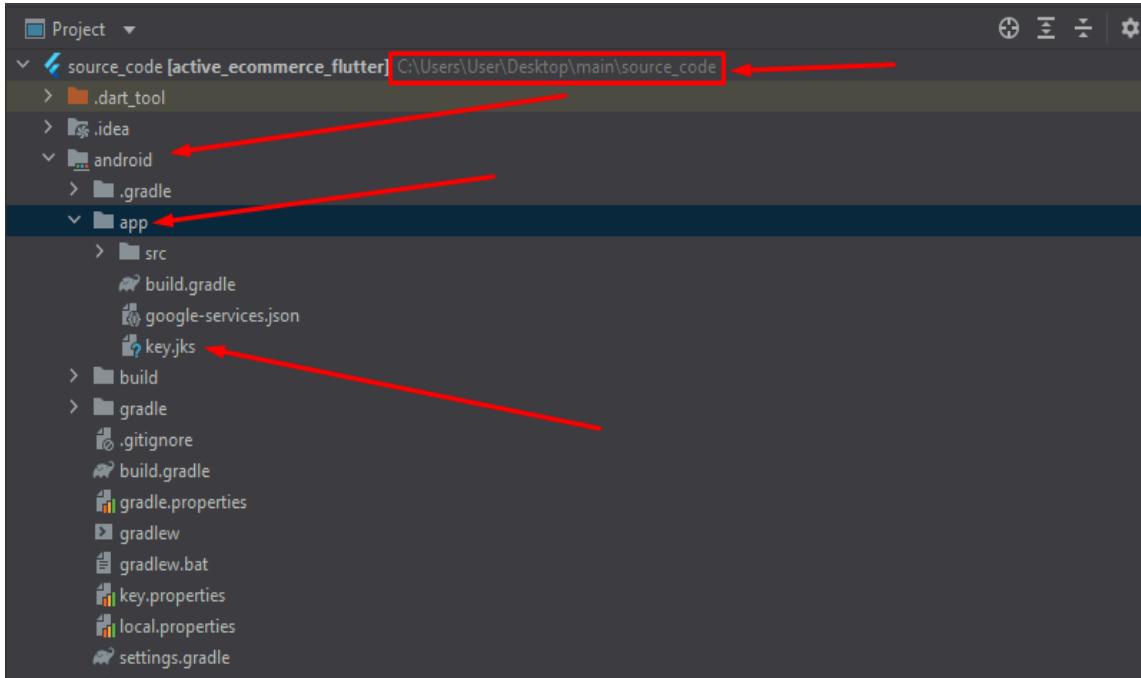
Create jks file for uploading app on playstore:

- 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 100000 -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

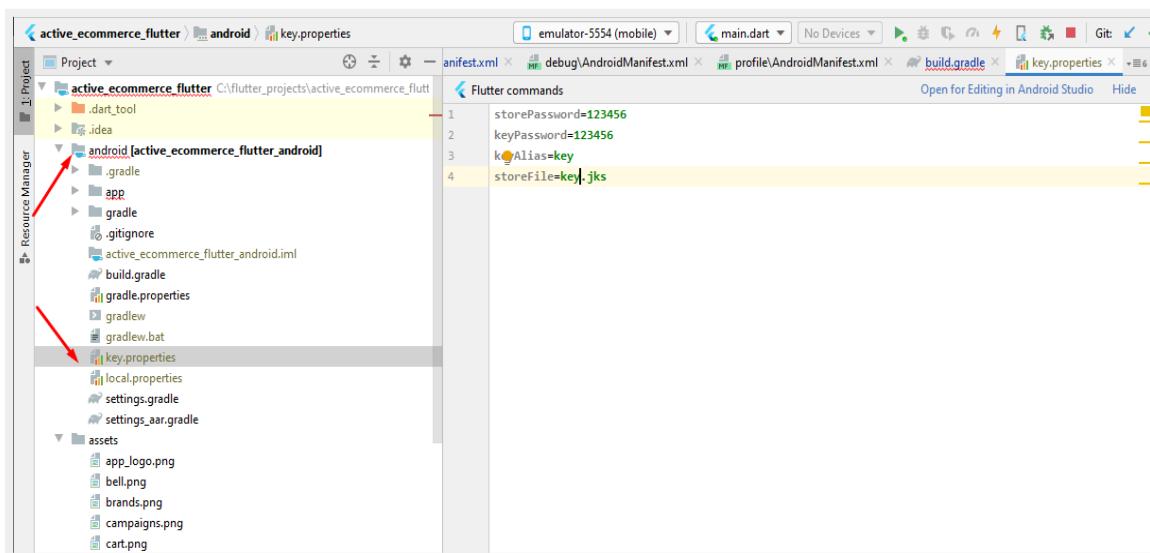
Create a file named `<your app dir>/android/key.properties` that contains a reference to your keystore:

```
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:

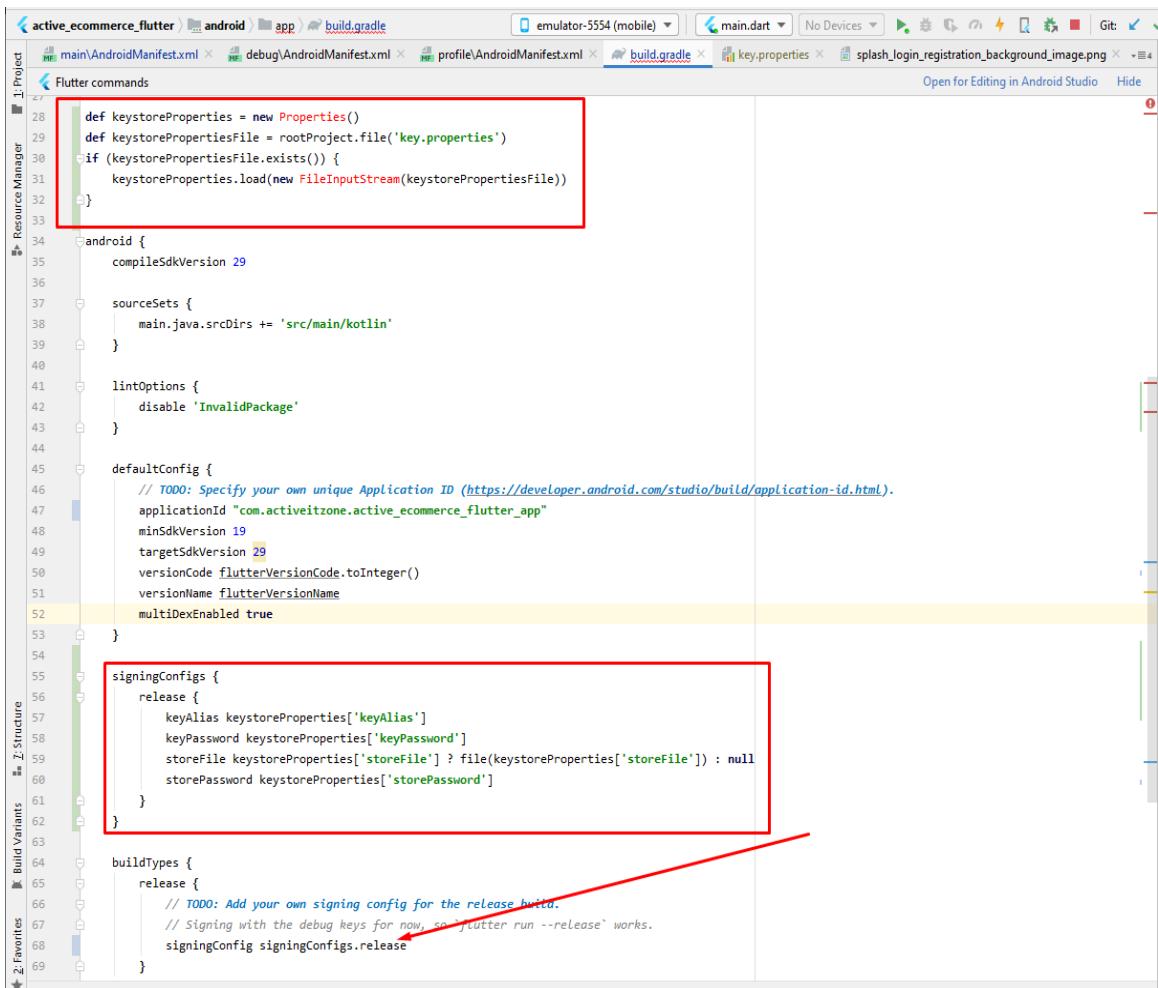
```
signingConfigs {  
    release {  
        keyAlias keystoreProperties['keyAlias']  
        keyPassword keystoreProperties['keyPassword']  
        storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null  
        storePassword keystoreProperties['storePassword']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



```
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.activezone.active_ecommerce_flutter_app"
        minSdkVersion 19
        targetSdkVersion 29
        versionCode flutterVersionCode.toInt()
        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

## 17. 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 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.

## 18. How to add frequently bought products?

**Answer:**

**Admin needs to choose this from the e-commerce web admin panel.**

The screenshot shows a user interface for editing a product. On the left, there's a sidebar with tabs: General, Files & Media, Price & Stock, SEO, Shipping, and Frequently Bought (which is currently selected). The main area has tabs for English and Bangla. Below these, under the 'Frequently Bought' tab, there are two radio buttons: 'Select Product' and 'Select Category'. A red arrow points to the 'Select Category' button, which is selected (indicated by a blue circle). Another red arrow points to the text input field where 'Women Clothing & Fashion' is typed. There's also a validation note 'Category \*' next to the input field.

There are two types of frequently bought product.

1. Product wise
2. Category wise.

For Category wise frequently bought product-> When select category for a specific product,

Main category is woman & clothing fashion only those product will show to that product

The screenshot shows the 'Edit Product' interface with the 'General' tab selected. In the 'Product Information' section, the 'Product Name' field contains 'Apple iPhone 14 Pro , 128GB, 512GB, Deep Purple-Unlocked'. The 'Brand' field is set to 'Apple'. Under 'Tags', there are three entries: 'iphone', 'mobile', and 'smart phone'. In the 'Product category' section, the 'women Clothing & Fashion' checkbox is checked and highlighted with a red box. A red arrow points from the 'Select Main' button to this checkbox. Other categories listed include 'Hot Categories', 'Wedding & events', 'Bottom', 'Tops & sets', 'Accessories', 'Women Watch', 'Baby Dresses', 'Men Clothing & Fashion', 'Computer & Accessories', 'Automobile & Motorcycle', and 'Kids & toy'.

## 19. How to remove cache data.

**Answer:**

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.