

# Documentation for StackFood

## Introduction

Thank you for your interest in StackFood.

StackFood is a complete Multi Restaurant food delivery system developed using Laravel and Flutter Framework.

Laravel (PHP Framework for Web Artisans) is a web application framework with an expressive, elegant syntax. We've already laid the foundation — freeing you to create without sweating the small things.

Flutter framework created by Google is open-source mobile application development. It is used to develop applications for Android and iOS, as well as being the primary method of creating applications.

StackFood **comprises** of 3 Mobile Applications, 2 Web Panel and 1 landing page –

- Customer application for both Android and iOS developed using Flutter.
- Delivery Boy application for Android developed using Flutter (Not Included).
- Restaurant application for both Android and iOS developed using Flutter (Coming soon).
- Super Admin web panel developed using Laravel (admin can provide role base access).
- Restaurant web panel developed using Laravel (owner can provide role base access).
- Landing page developed using Laravel blade.

**IMPORTANT NOTE:** The delivery boy app is developed using flutter so it will run both in Android and iOS but as the delivery boy app fully depends on location we suggest don't use it for iOS.

This documentation covers mostly all the aspects of setting up Multi Restaurant Food Delivery Application - StackFood.

For any queries pre-sales or tech support, please drop a mail to [support@6amtech.com](mailto:support@6amtech.com)

## Delivery Man App setup

Note: This is not an independent app. This app is available only for StackFood Multi Restaurant delivery app. You should be pre-installed the StackFood in your server.

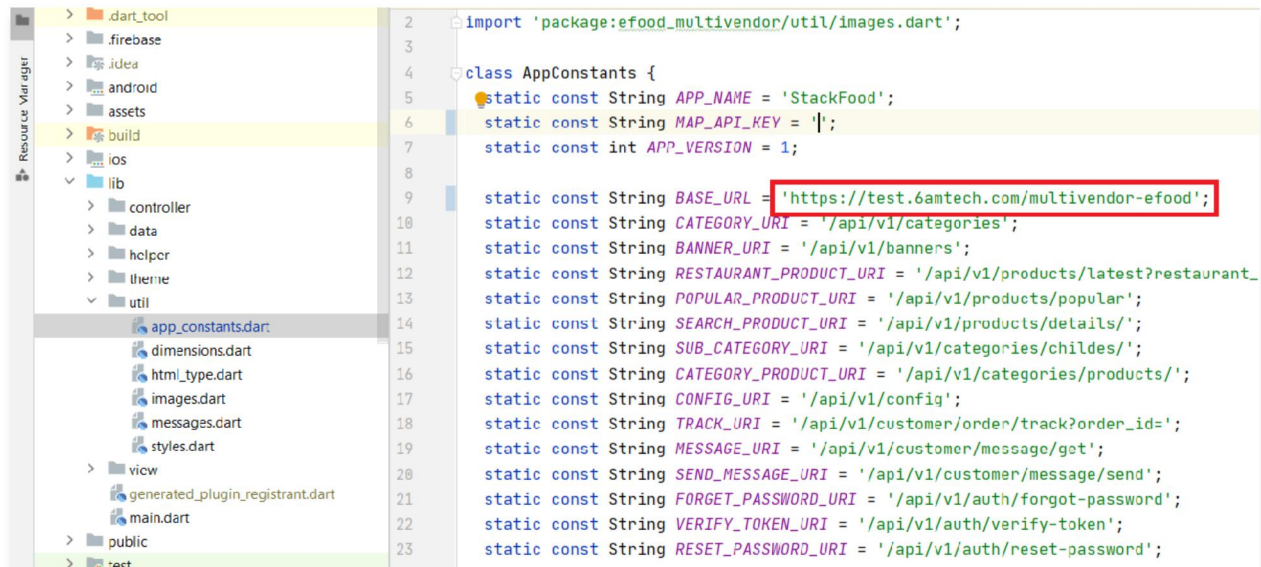
**It is extremely easy and convenient to rebrand the app and customize the theme of the app as per your needs.**

You can launch an app with your own branding. You can easily change the app color theme, logos, and icons with minimal effort.

The app has intuitive and user-friendly animations built-in within it. These animations intend to provide a smooth app usage experience to the end clients. The animations include – Hero Animations, Parallax Animations, Sliding & Swiping animations.

## Change base URL

Open `/lib/util/app_constraints.dart` and replace `BASE_URL` variable value with your own URL

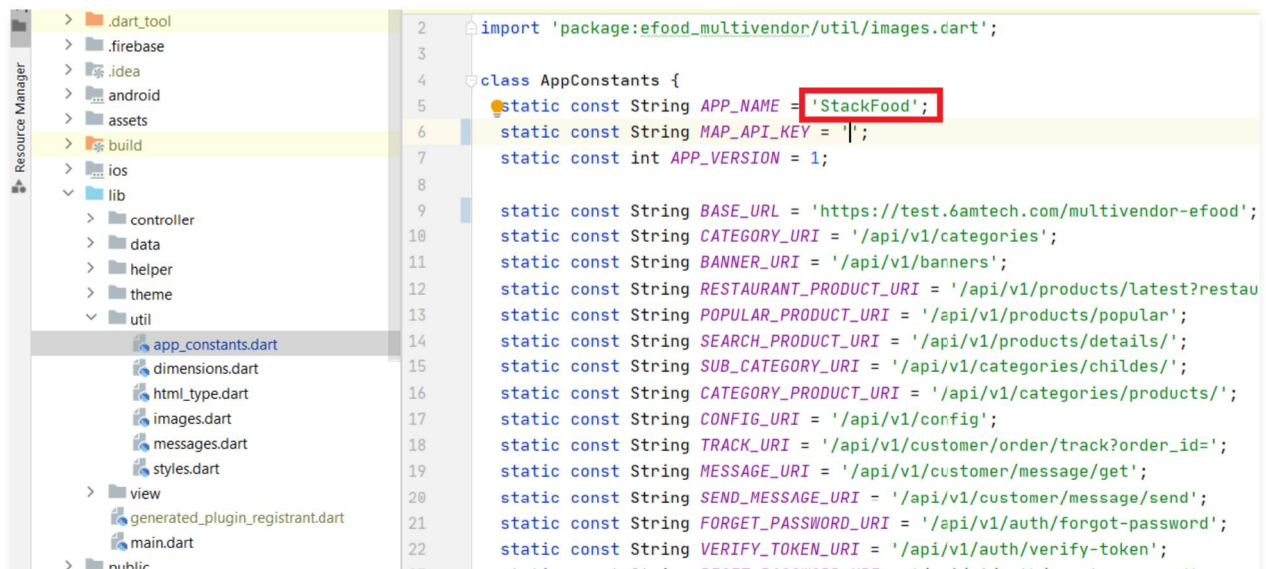


## Change app name

To change the app name for whole app follow the below steps-

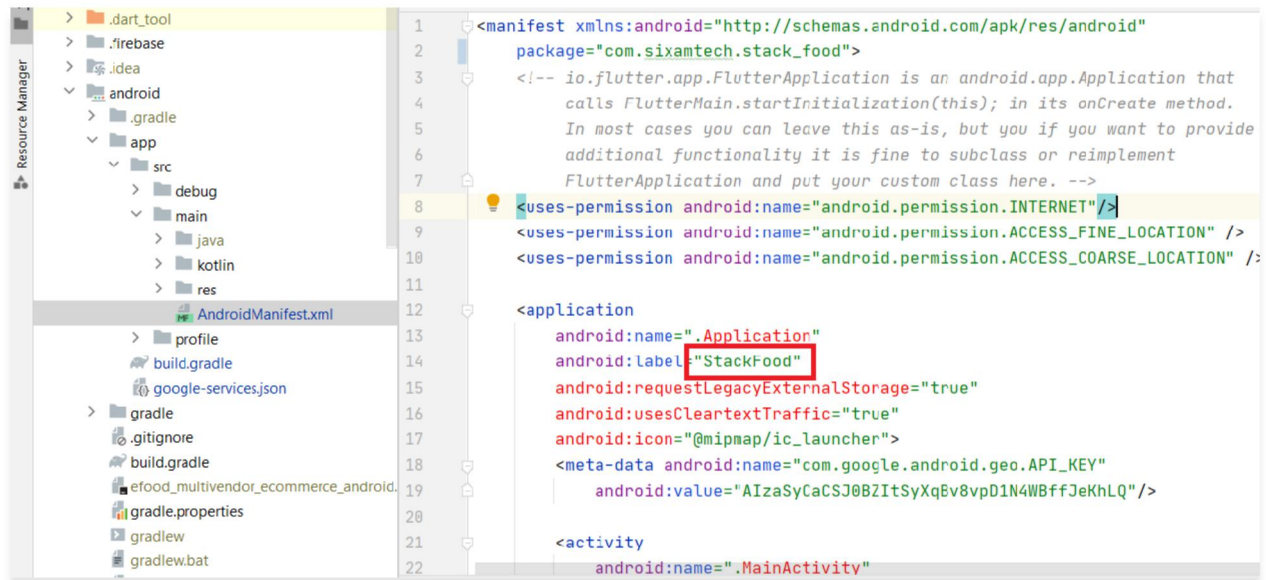
### Step 1

Open `/lib/util/app_constraints.dart` and replace `APP_NAME` variable value with your Name



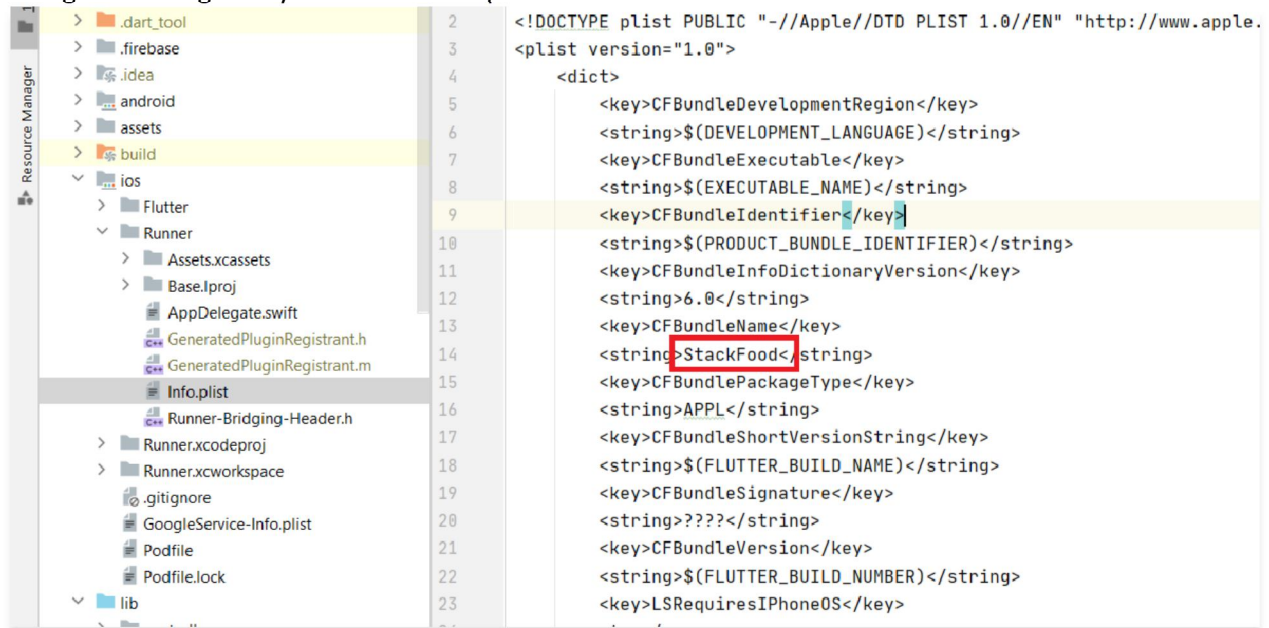
## Step 2

Change the label with you name in AndroidManifest.xml file



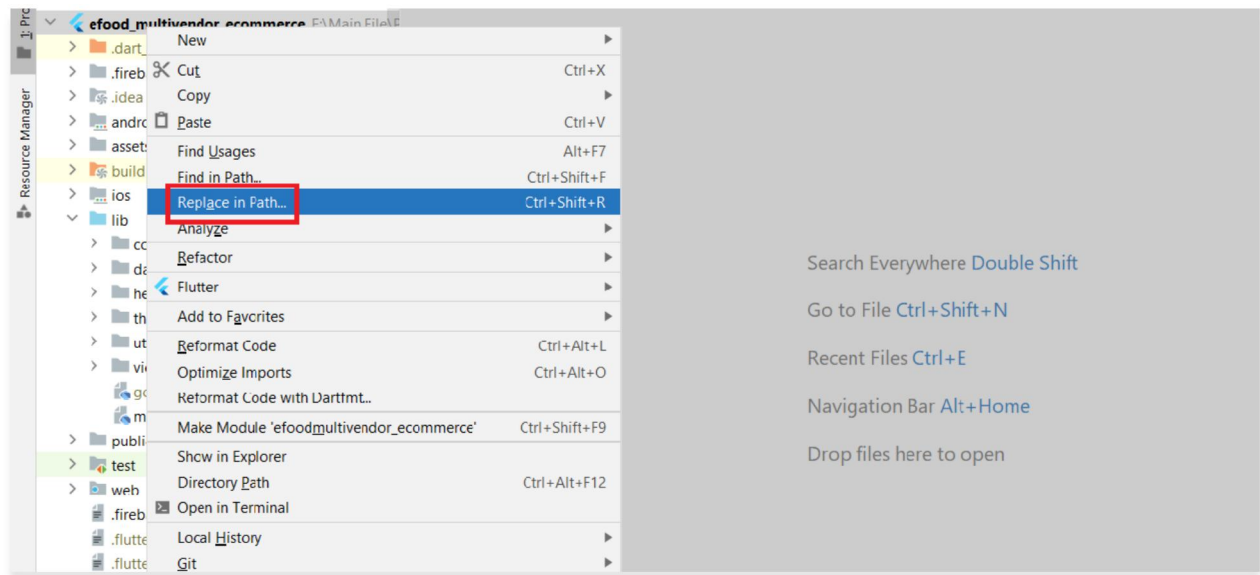
## Step 3

Change the string with you name in Info.plist file

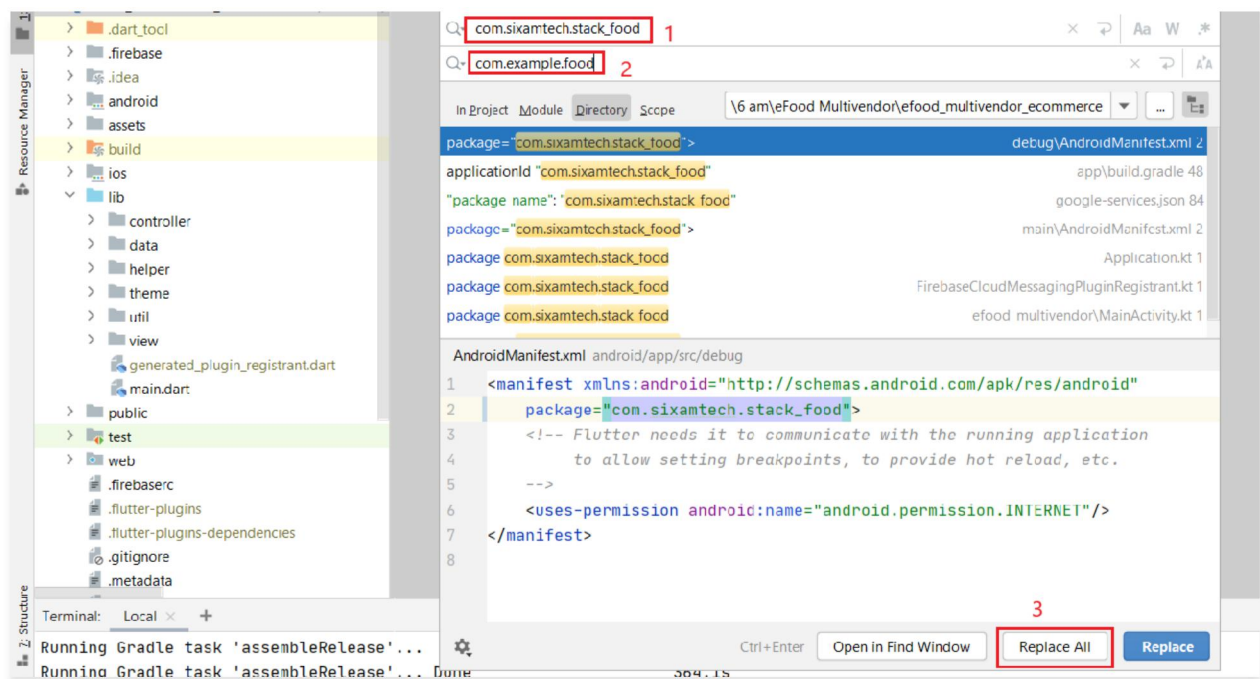


## Change app package

Right click on project folder and click on replace in path



Write current Package name in 1st marked box and write your new package name 2<sup>nd</sup> marked box and then click on Replace All button. **Please change this carefully we are not responsible if you make something wrong here.**



## Add firebase for push notification

### Step 1

Create your own firebase project from <https://console.firebase.google.com> and also add an android app there with your own package name and app name.

#### Step 2

Click register app and download google-services.json file from there.

#### Step 3

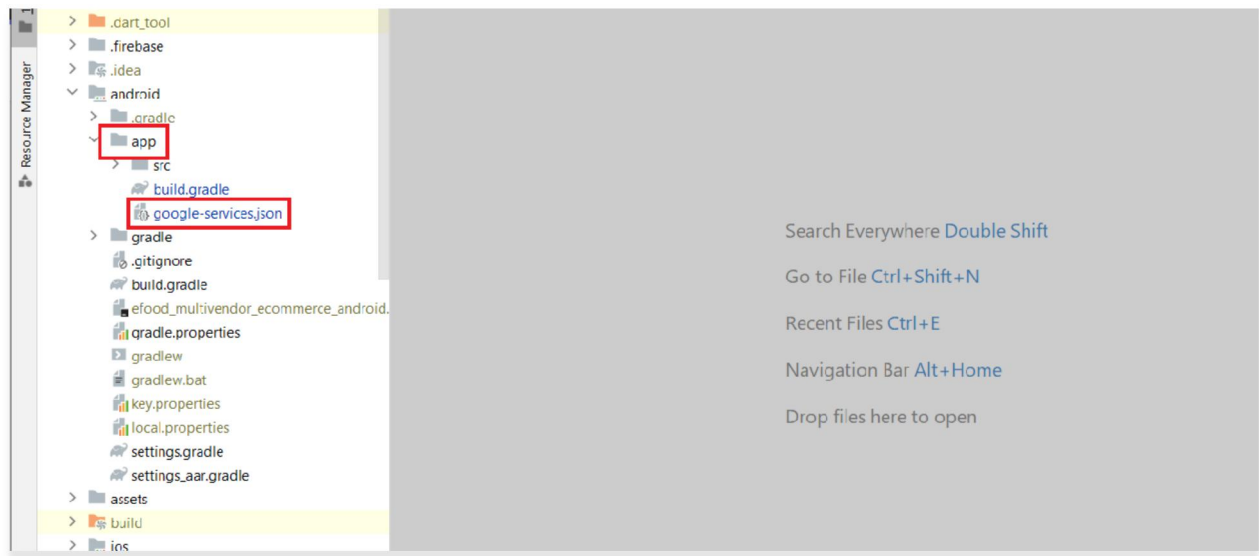
Copy that file and paste and replace it in /android/app

#### Step 4

After replace the file please restart your IDE.

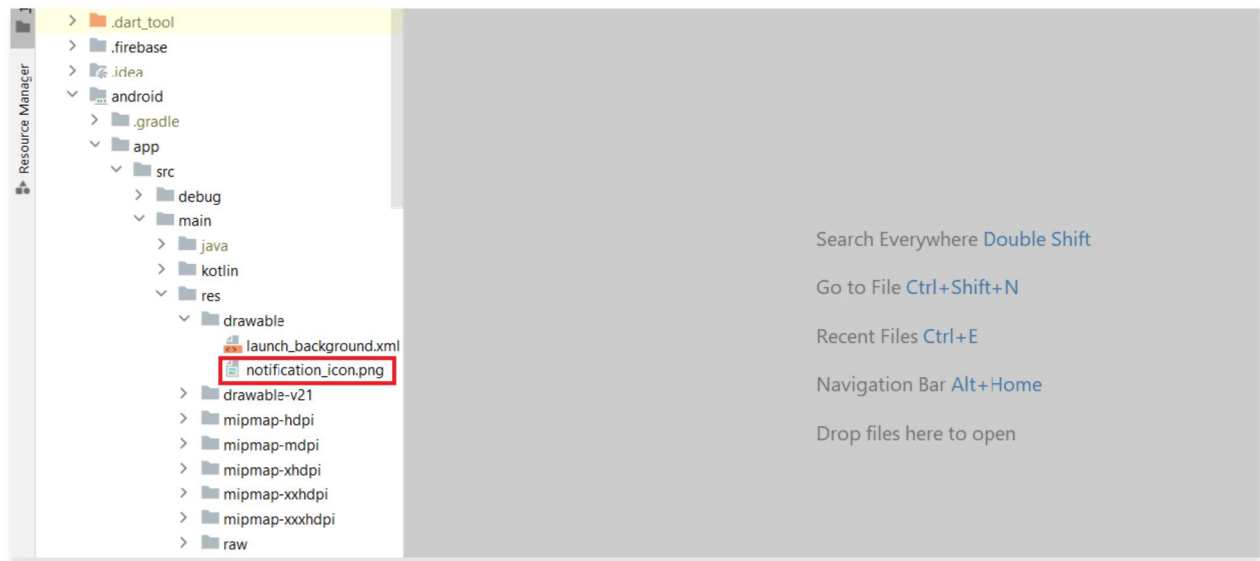
**IMPORTANT NOTE:** Must add server key in Admin panel's Notification Settings to get push notification.

Goto Project settings-> Cloud messaging-> Server key



#### Change the notification icon

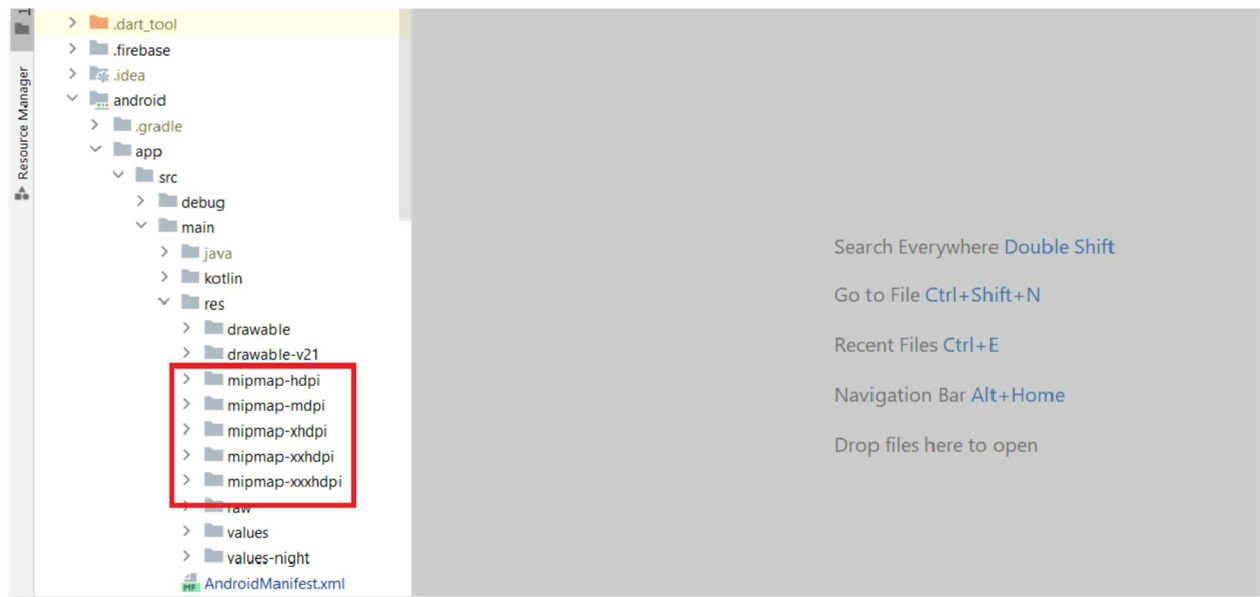
Must use a white icon for push notification



## Change App icon and Logo

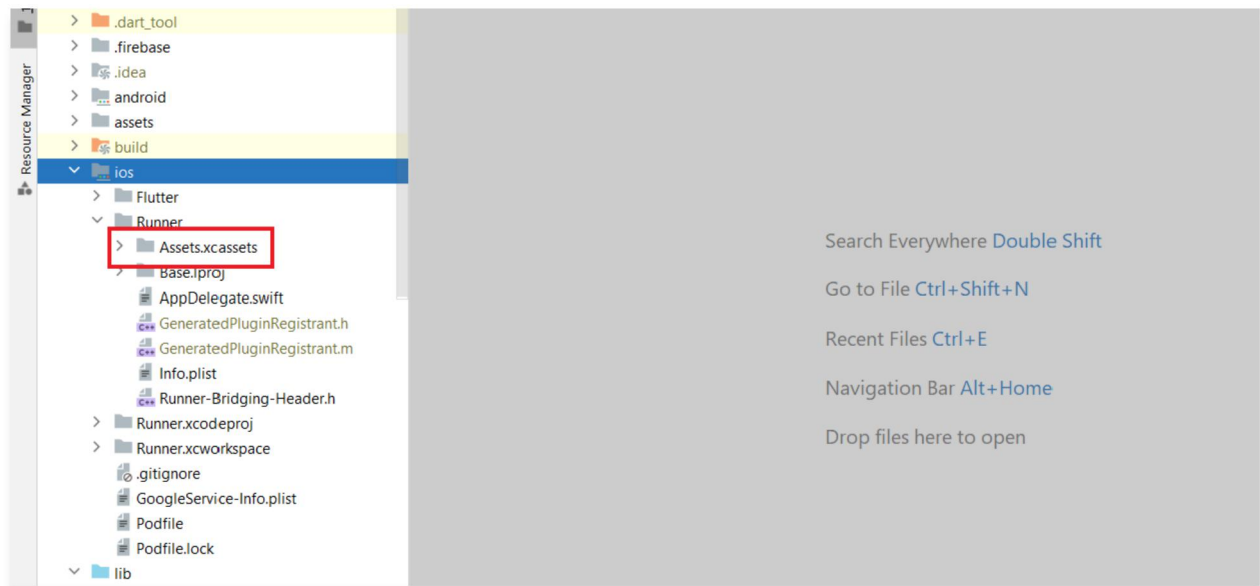
Generate icon from [appicon.co](https://appicon.co)

Go to `/android/app/src/main/res` and replace all mipmap folders with your generated icon android folder.

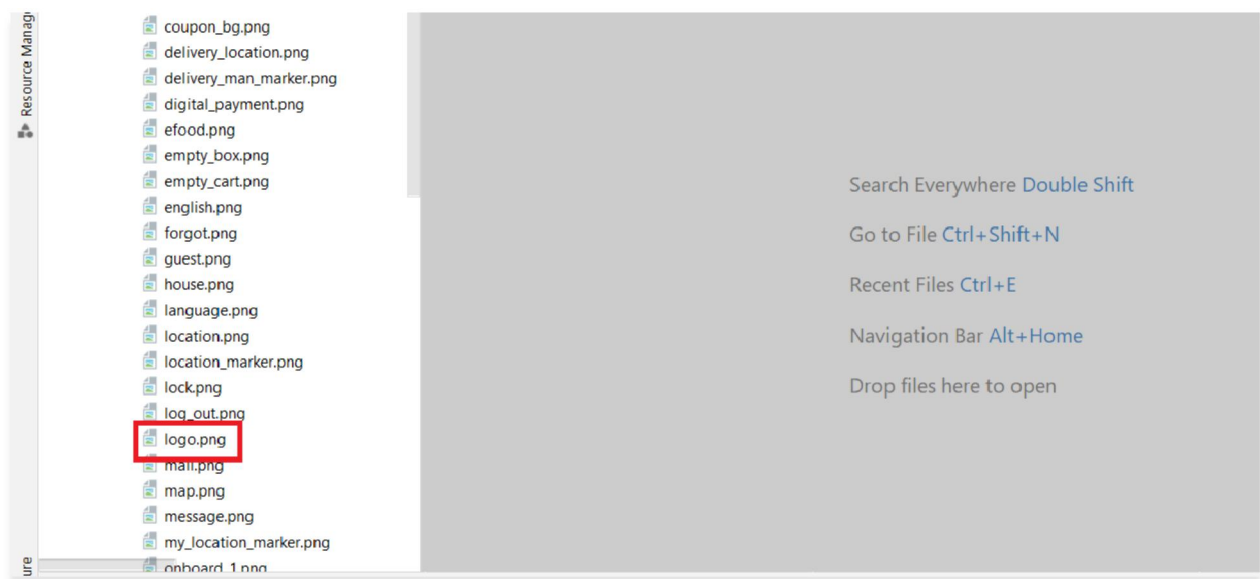


Go to `/ios/Runner` and replace Assets.xcassets with your generated Assets.xcassets folder.



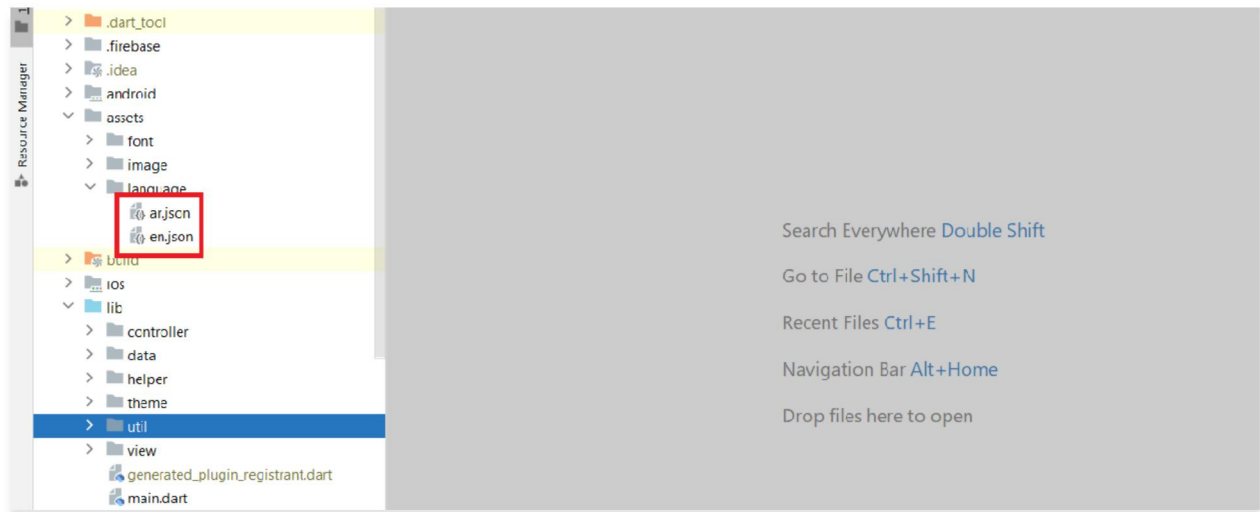


Go to <project>/assents/image folder and replace logo.png with your own one



### Add new local language

Go to /assets/language and press right button on language folder and create new file and name it with your language code (.json)



Copy all data from en.json and paste it in your created file.

Translate all English text placed here after colon(:) to your local language.

Copy your country picture and paste it on /assets/image.

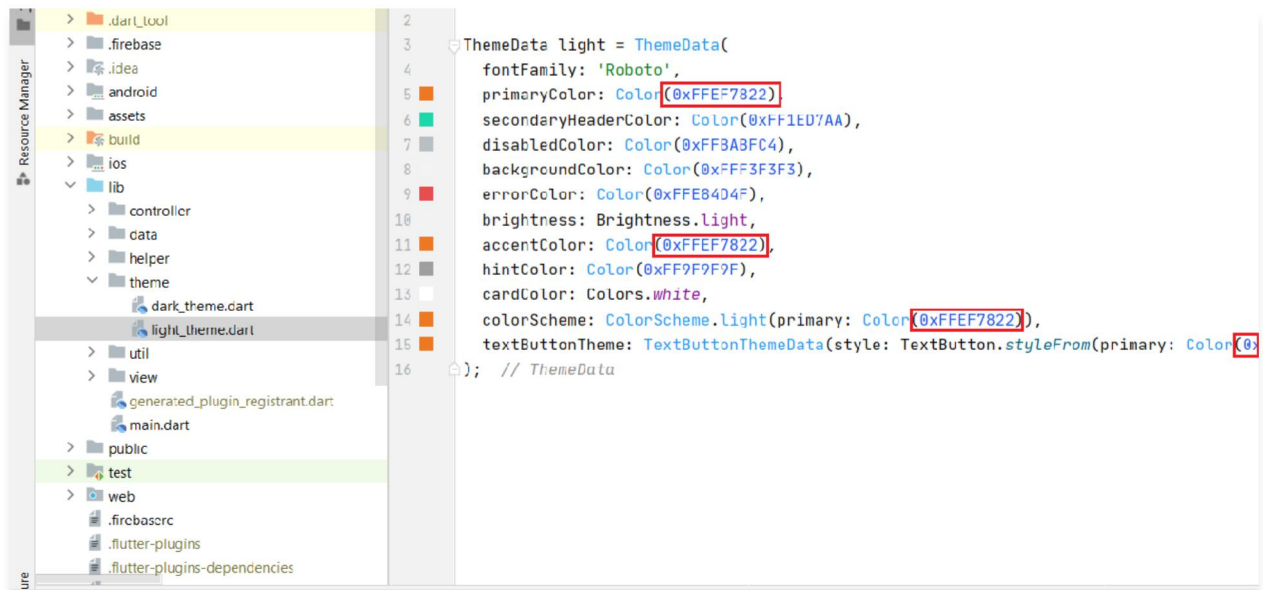
Open /lib/util/app\_constants, scroll down to bottom and add LanguageModel with imageUrl, languageName, countryCode and languageCode.



## Change app color

Goto /lib/theme/ and change your Primary, Accent etc. Colors.





### Build apk:

Release APK:

*flutter build apk --split-per-abi*

Release Bundle:

*flutter build appbundle*

Command for normal apk build:

*flutter build apk --target-platform android-arm,android-arm64,android-x64 --split-per-abi*

IMPORTANT NOTE: Please do your customization carefully we are not responsible if make any mistake.

Thank you & Best of Luck