

Happy Desire - Flutter Ecommerce App Template | Documentation

Welcome

Thank you for purchasing my app. If you have any questions that are beyond the scope of this help file, please feel free to email via my user page contact form here :

`royhayek27@gmail.com`

Android Studio

1) Download the latest stable release of the Flutter SDK from here: [r](https://flutter.dev/docs/get-started/install)

```
https://flutter.dev/docs/get-started/install
```

2) Extract the zip file in a desired install location (for example, C:\src\flutter ; do not install Flutter in a directory like C:\Program Files because it may require administrator privileges).

Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

- From the Start search bar, enter 'env' and select **Edit environment variables for your account**.
- Under **User variables** check if there is an entry called **Path**:
 - If the entry exists, append the full path to flutter\bin using ; as a separator from existing

Now from a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:content_copy (sometimes the pc needs to be restarted in order for changes to take effect)

```
flutter doctor
```

This command checks your environment and displays a report of the status of your Flutter installation.

Android Studio Setup

Note: Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step will discuss that

Install Android Studio

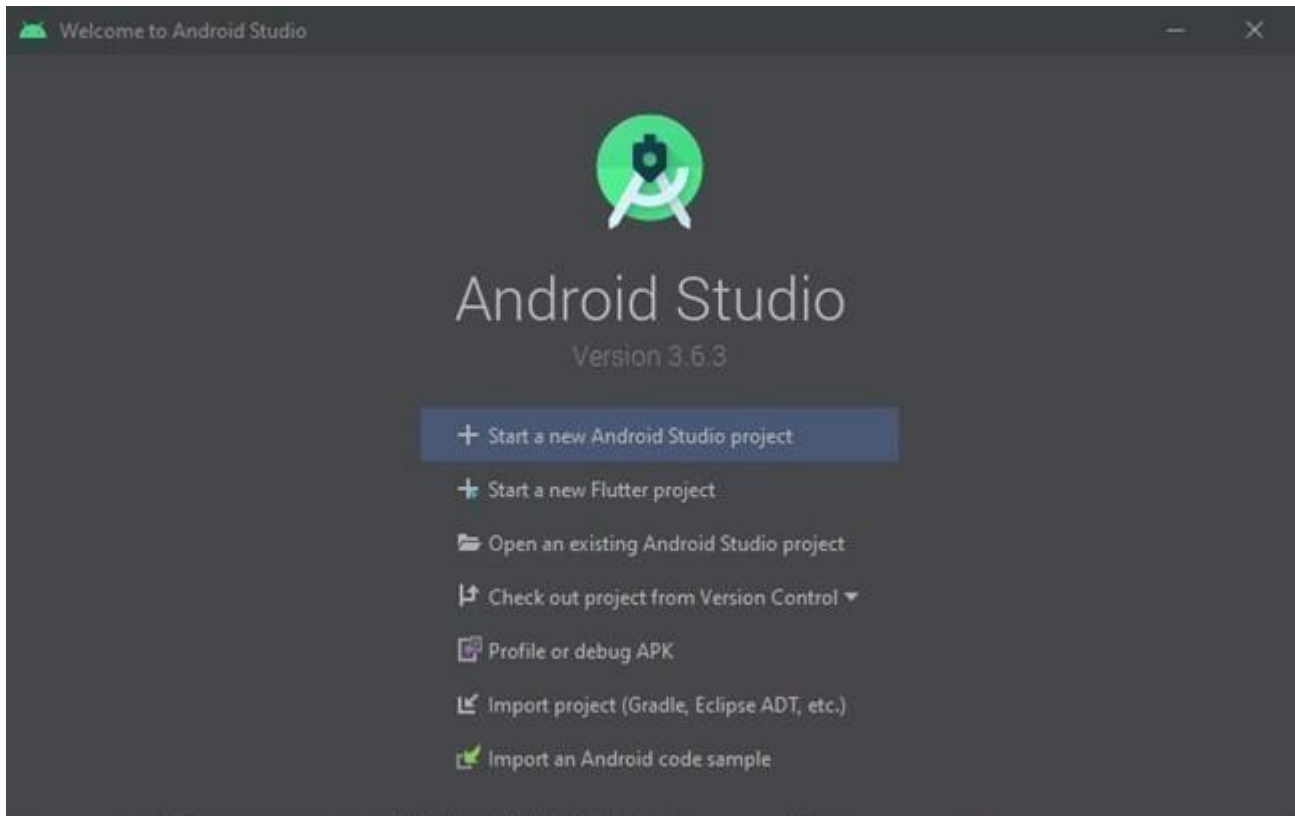
1. Download and install the latest version of Android Studio from here: (Keep everything as default in the installation just change the installation location if you want).

```
https://developer.android.com/studio/
```

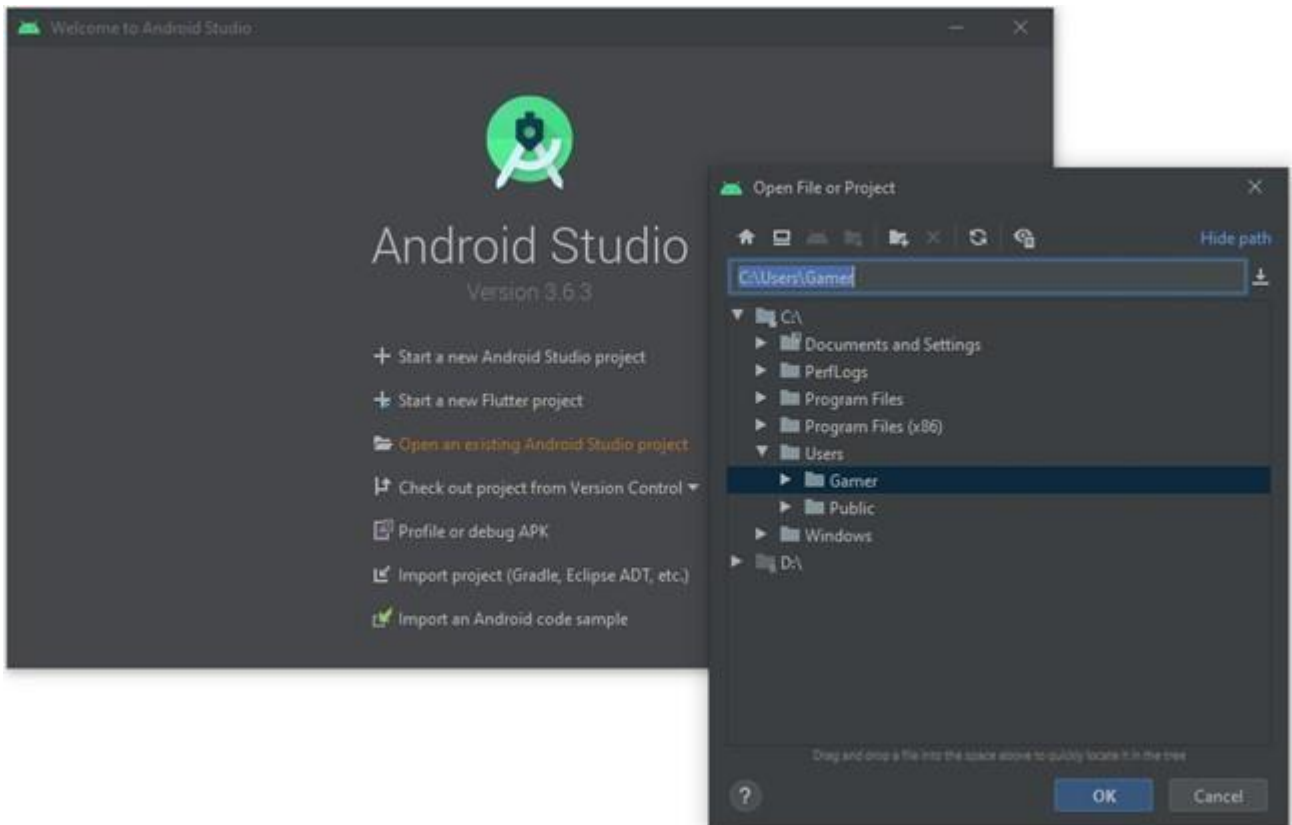
Import Project (Part 1)

Extract **.zip** you download from Codecanyon, you will see some file and folder. the project folder is named "**HappyDesrieFlutterAppTemplate**" inside the "**android_studio**" folder extract it in a desired location.

1. Open Android Studio



1. Open the project by pressing on "**Open an existing Android Studio project**" (Navigate to where you extracted the **.zip** file, press on "**HappyDesrieFlutterAppTemplate**" then press "**OK**")



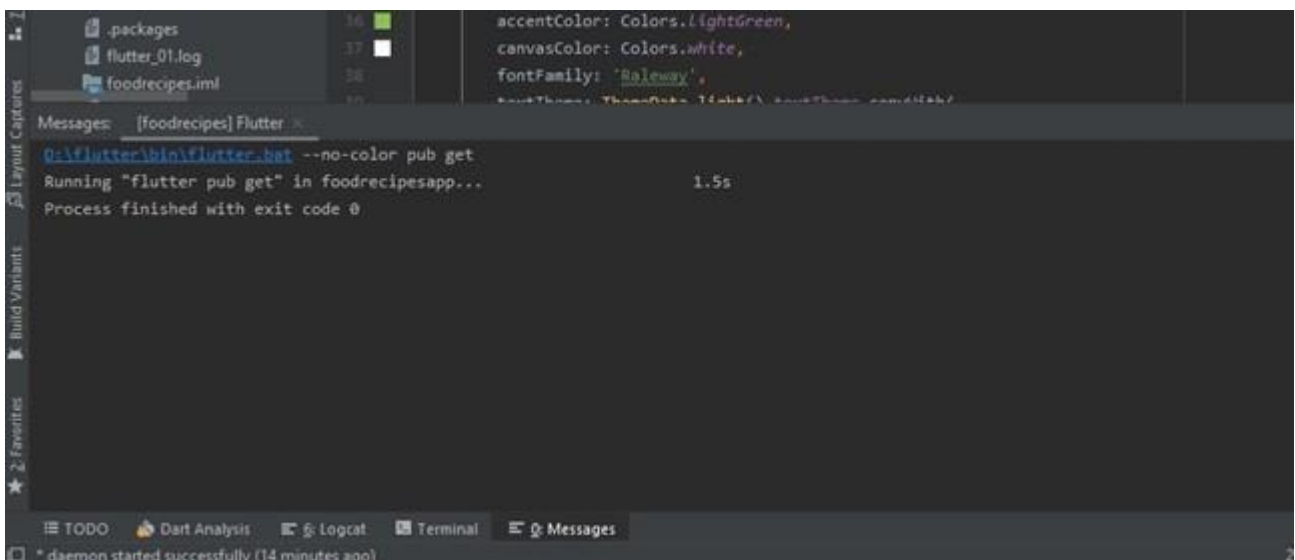
1. You will get a message at the top of the screen saying "**Packages get has not been run**" Click **get dependencies** to get flutter libraries. (if prompt)



If after pressing **get dependencies** you still have missing packages, go to **pubspec.yaml** and press **Pub get** at the top of the screen.

Import Project (Part 2)

4. Wait until the process is done down in the Messages, as shown in the picture below:



Configure Device

To prepare to run and test your Flutter app on an Android device, you'll need an Android device running Android 4.1 (API level 16) or higher.

1. Enable Developer options and USB debugging on your device. Detailed instructions are available in the Android documentation.
2. Windows-only: Install the Google USB Driver.
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID_HOME environment variable to that installation directory.



Sometimes you have to specify the project SDK manually in order for your device to be detected in Android Studio you can do it by going to "File" ->

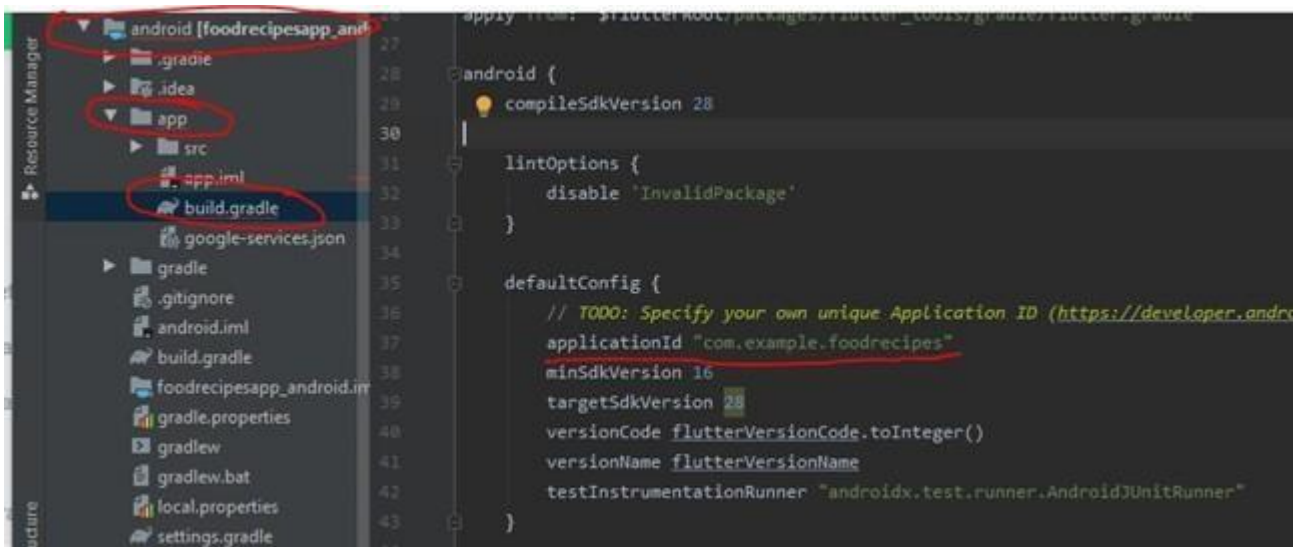
"Project Structure" and in the "Project SDK" press on **Android API Platform**.

Package Name

FOR ANDROID:

Step 1

1. Open **Gradle Scripts** → **build.gradle (Module: app)**

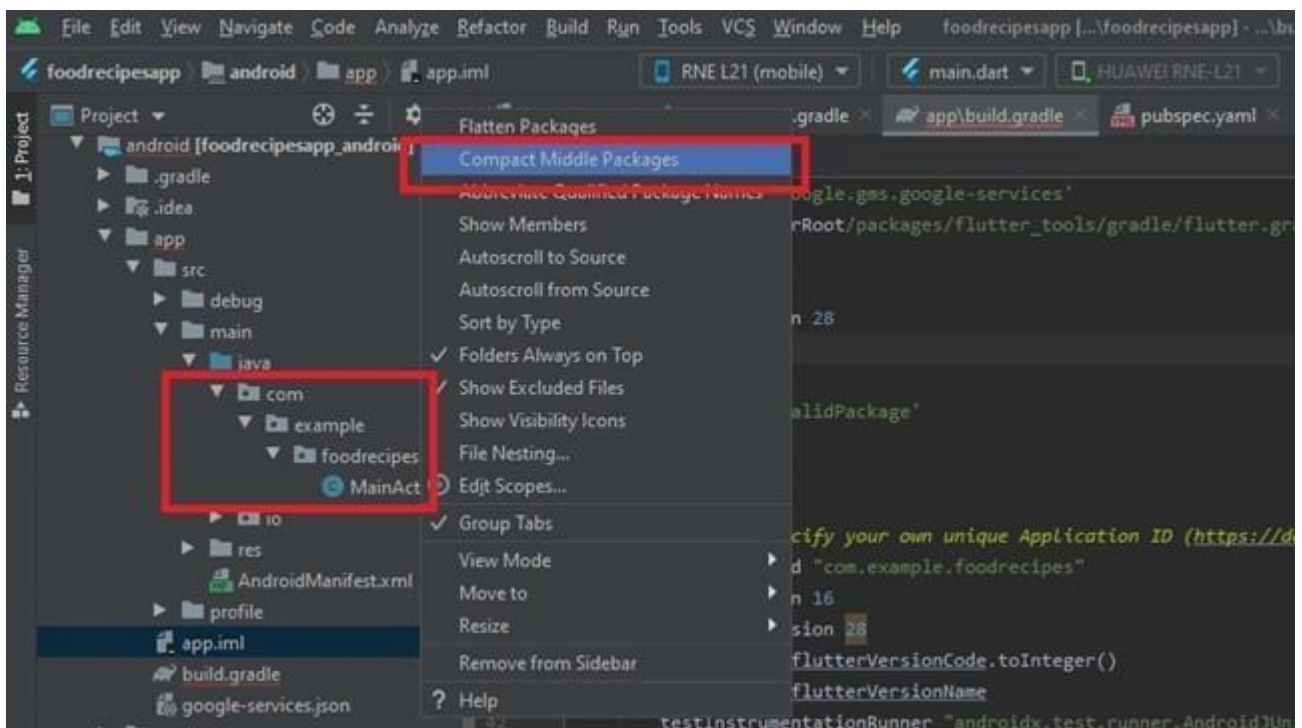


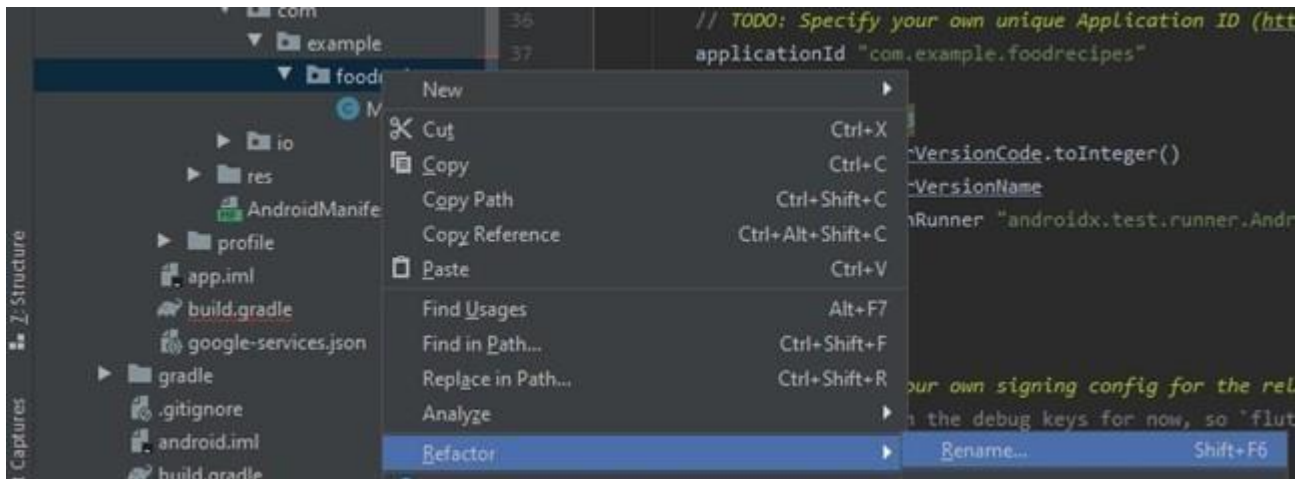
1. Change the application ID with your own id name
2. Standard naming procedure: **com.yourdomainname.yourappname**
3. Change the id, make it as unique as possible, because application id is very important if you want to publish the application in the store.

! **Important** : your applicationId or package name must be the same to the package name you've create in your firebase console when generating google-services.json

Step 2

1. Click once on your package and then click setting icon on Android Studio.
2. Close / Unselect **Compact Empty Middle Packages**
3. Right click your package and rename it.





FOR IOS:

In iOS the package name is the bundle identifier in "**ios/Runner/Info.plist**":

```
1 <key>CFBundleIdentifier</key> <
2 2 <string>$(PRODUCT_BUNDLE_IDENTIFIER)</string> >
```

Which is found in "**Runner.xcodeproj/project.pbxproj**":

```
PRODUCT_BUNDLE_IDENTIFIER = com.example.appname;
```

But there are 3 lines of **PRODUCT_BUNDLE_IDENTIFIER** in "**project.pbxproj**" which you have to change, locate it by pressing **CTRL + F** and looking for **PRODUCT_BUNDLE_IDENTIFIER**.

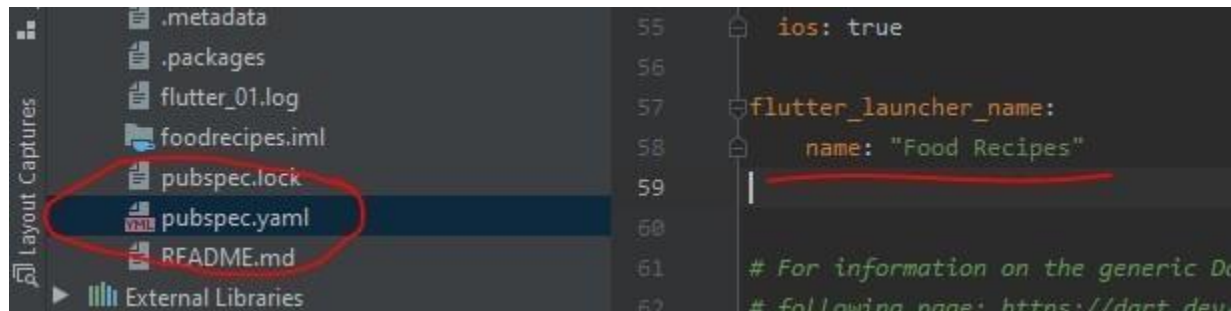
Last Step

In order for the rate option to work in the application you have to update your Google Play and your App Store identifiers. To do that go to "**lib/constants.dart**" and change the **GooglePlayIdentifier** value to your package name and the **AppStoreIdentifier** value to your App store identifier.

App Name

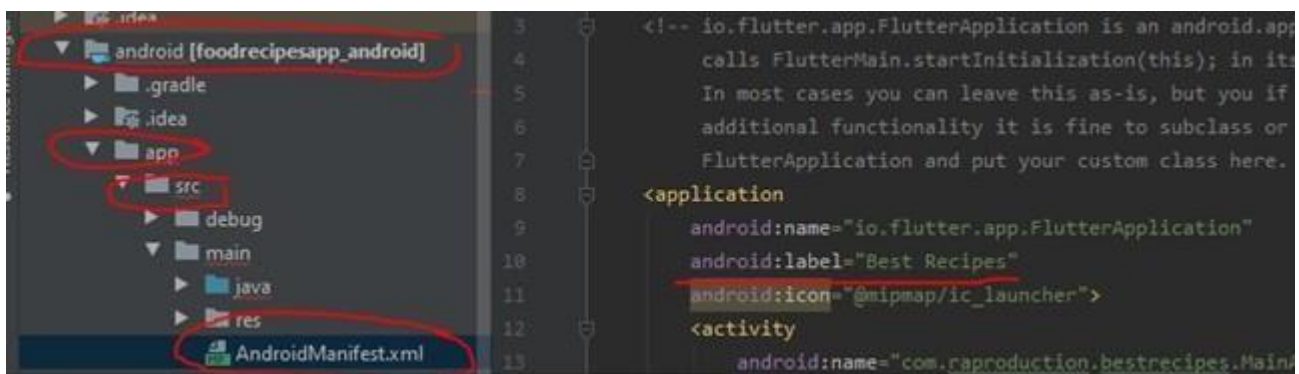
To change your app name follow the below steps: **Step 1**

go to "**pubspec.yaml**" located in your project root folder then change the flutter_launcher_name to your desired app name.



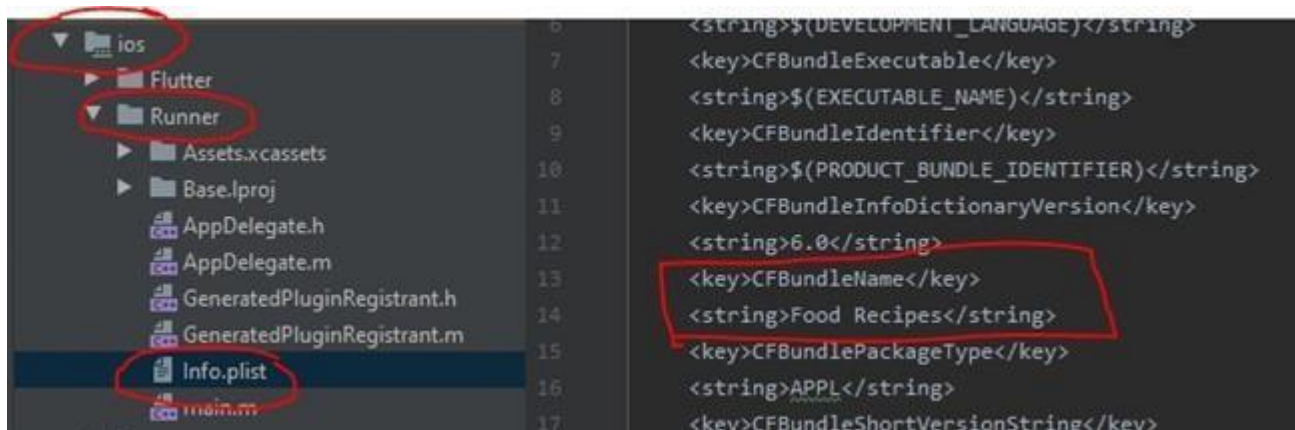
Step 2

go-to "**android/app/src/AndroidManifest.xml**" located in your project root folder then change the label to your desired app name.



Step 3

go to "**ios/Runner/Info.plist**" located in your project root folder then change the CFBundleName value to your desired app name.

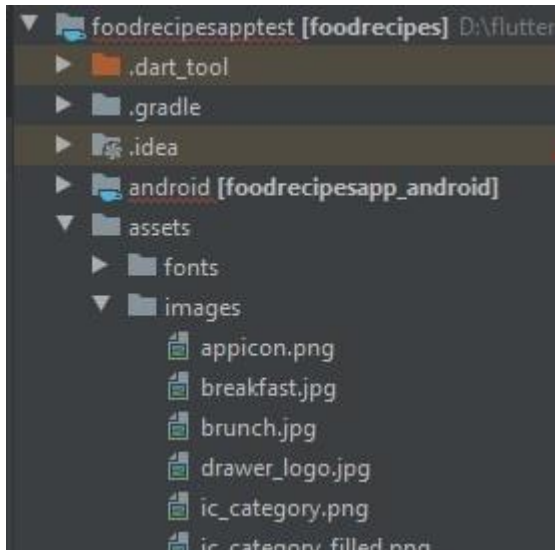


App Icon

To change your app name follow the below steps:

Step 1

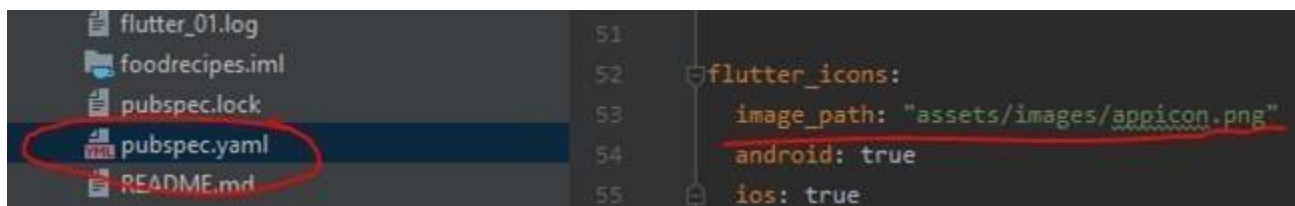
Add your logo in "assets/images"



Step 2

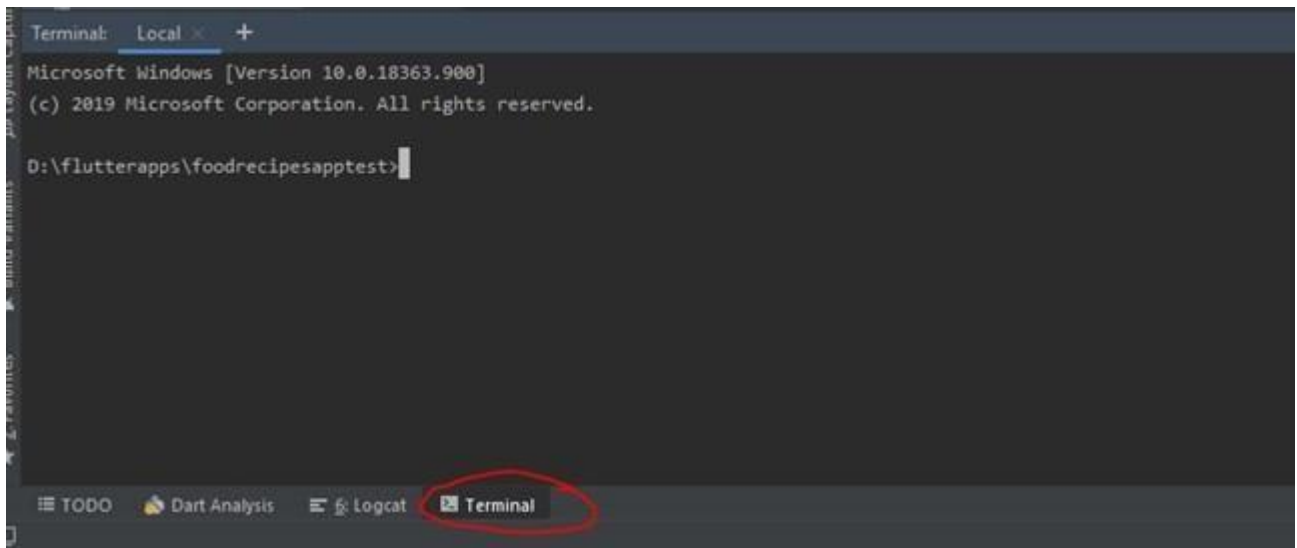
Go to "pubspec.yaml" and change the edit the image_path in this way

"assets/images/your_icon_name.png" (or jpg depending on your icon)



Step 3

Open the Terminal from the bottom of the screen



And execute the below commands in the terminal

```
flutter pub get
```

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

After that, your icon will be updated

Before Running or Building

Before running or building the application you need to switch your flutter channel to the **master** channel if it is not, by running the following command:

```
flutter channel master
```

Otherwise, you will get errors and some dependencies may be missing.



You can check your current channel by running "**flutter channel**"