

Introduction To MaanStore WooCommerce Flutter App

Version: 4.1

MaanStore WooCommerce Flutter app is as the name says is dominating and powerful set of Flutter apps containing complete app templates, complete API integration with WooCommerce, ready-to-use widgets, and code snippets i.e. readymade code integration and stunning screens covering many different use cases. The app comes with the latest material design. Maan Store UI templates are developed with the highest quality, ease of reusing widgets, fast, and completely user-friendly interface. You can easily integrate these into your WooCommerce store. Maan Store WooCommerce app makes your job easy to achieve the modern look and feel of the mobile application. It saves your hustle and time from developing an app for your e-commerce website made with WooCommerce.

Prerequisite:

- 1 Android Studio: 4.1.2
- 2 Flutter: 2.10.0

For 24/7 support email us here: maantheme@gmail.com

Note: No extra plugin is required. Just WooCommerce jwt authplugin.

Tools & Setup

Prerequisite

- Flutter & Dart [SDK](#)
- Any IDE Android Studio (Recommended), [Visual Studio Code](#) or [IntelliJ IDEA](#)
- To edit this project you must have Flutter and Dart installed and configured successfully on your computer.
- Set up your editor - Install the [Flutter and Dart plugins](#).
- If you have got Android SDK installed and configured, to install Flutter you only need to:
 - ◊ Download Flutter SDK from official website and extract it.
 - ◊ Add path to previously extracted SDK to your PATH variable
 - ◊ Run flutter doctor tool to check if everything is configured correctly.
 - ◊ All above steps are mentioned here: <https://flutter.dev/docs/get-started/install/>

Android Studio – Windows

- ♦ Download Android Studio - <https://developer.android.com/studio/Get>
- ♦ the Flutter SDK - <https://flutter.dev/docs/get-started/install>
- ♦ Learn more about Android Studio - <https://developer.android.com/studio/intro/>

Step 1 : Get the Flutter SDK

1 Download the following installation bundle to get the latest stable release of the Flutter SDK:

2 Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\src\flutter; do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges).

Step 2 : 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 values.
- ♦ If the entry doesn't exist, create a new user variable named Path with the full path to **flutter\bin** as its value.

Important

Note that you have to close and reopen any existing console windows for these changes to take effect. **You are now ready to run Flutter commands in the Flutter Console!**

Step 3 : Run flutter doctor

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:

```
c:\src\flutter>flutter doctor
```

Android Studio – macOS

- Download Android Studio - <https://developer.android.com/studio/>
- Download Xcode - <https://apps.apple.com/us/app/xcode/id497799835?mt=12> Get
- the Flutter SDK - <https://flutter.dev/docs/get-started/install>
- Learn more about Android Studio - <https://developer.android.com/studio/intro/>

Step 1 : Get the Flutter SDK

- Download the following installation bundle to get the latest stable release of the FlutterSDK:
- Download SDK and extract downloaded file, just double click on that. and just copy extracted folder and paste it to your desired location (for example, Documents\flutter).

Step 2 : Update your path

Important

Path variable needs to be updated to access “flutter” command from terminal. you can just update path variable for current terminal window only. and if you want to access flutter commands from anywhere in terminal, we need to update SDK path permanently.

To update PATH variable, we need to open terminal.

To update PATH variable for current terminal window only, then enter this command "**export PATH="\$PATH:`pwd`/flutter/bin"**" and hit enter key.

To update PATH variable permanently, then Open or create **.bash_profile** file. to open or create that file, then enter "**sudo open -e \$HOME/.bash_profile**" and hit enter key. Append below line to bash_profile file at bottom of all other content. "**export PATH="\$PATH:[PATH_TO_FLUTTER_GIT_DIRECTORY]/flutter/bin"**" as [PATH_TO_FLUTTER_GIT_DIRECTORY] is actual path of SDK folder.

Run this command on terminal "**source \$HOME/.bash_profile**" to refresh PATH variables. Then

check whether our SDK is successfully installed or not.

You are now ready to run Flutter commands in the Flutter Console!

Run "**flutter doctor**" into terminal, If you are getting check list of flutter sdk requirements, it means SDK is successfully installed on your machine. and you can start building flutter apps on your machine.

Android Studio – Linux

- ♦ Download Android Studio - <https://developer.android.com/studio> Get
- ♦ the Flutter SDK - <https://flutter.dev/docs/get-started/install/linux>
- ♦ Learn more about Android Studio - <https://developer.android.com/studio/intro/>

Step 1 : Get the Flutter SDK

- ♦ Download the following installation bundle to get the latest stable release of the FlutterSDK:
- ♦ Download SDK and extract downloaded file, just double click on that. and just copy extracted folder and paste it to your desired location (for example, Documents\flutter).

Step 2 : Update your path

Important

Path variable needs to be updated to access “flutter” command from terminal. you can just update path variable for current terminal window only. and if you want to access flutter commands from anywhere in terminal, we need to update SDK path permanently.

You'll `probably` want to update this variable permanently, so you can run flutter commands in any terminal session. To update PATH variable, we need to open terminal.

1. Run `source $HOME/.flutter/bin` to refresh the current window, or open a new terminal window to automatically source the file.
2. Verify that the `flutter/bin` directory is now in your PATH by running:

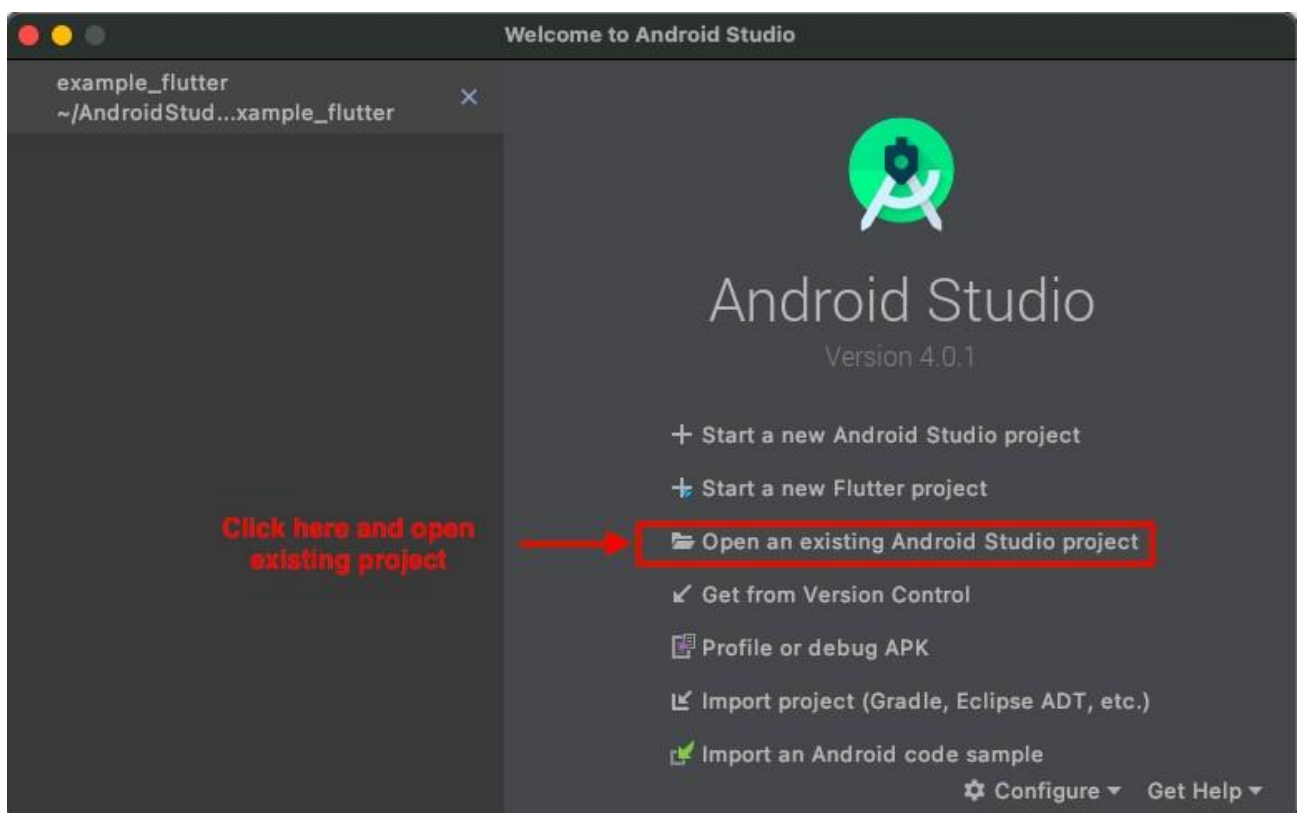
GETTING STARTED (BUILD & RUN)

Important

All below steps are must be followed to build and run application

Download Project

Download and find the your project folder, use your preferred IDE (**Android Studio** / **Visual Studio Code** / **IntelliJ IDEA**) to run the project.



logo

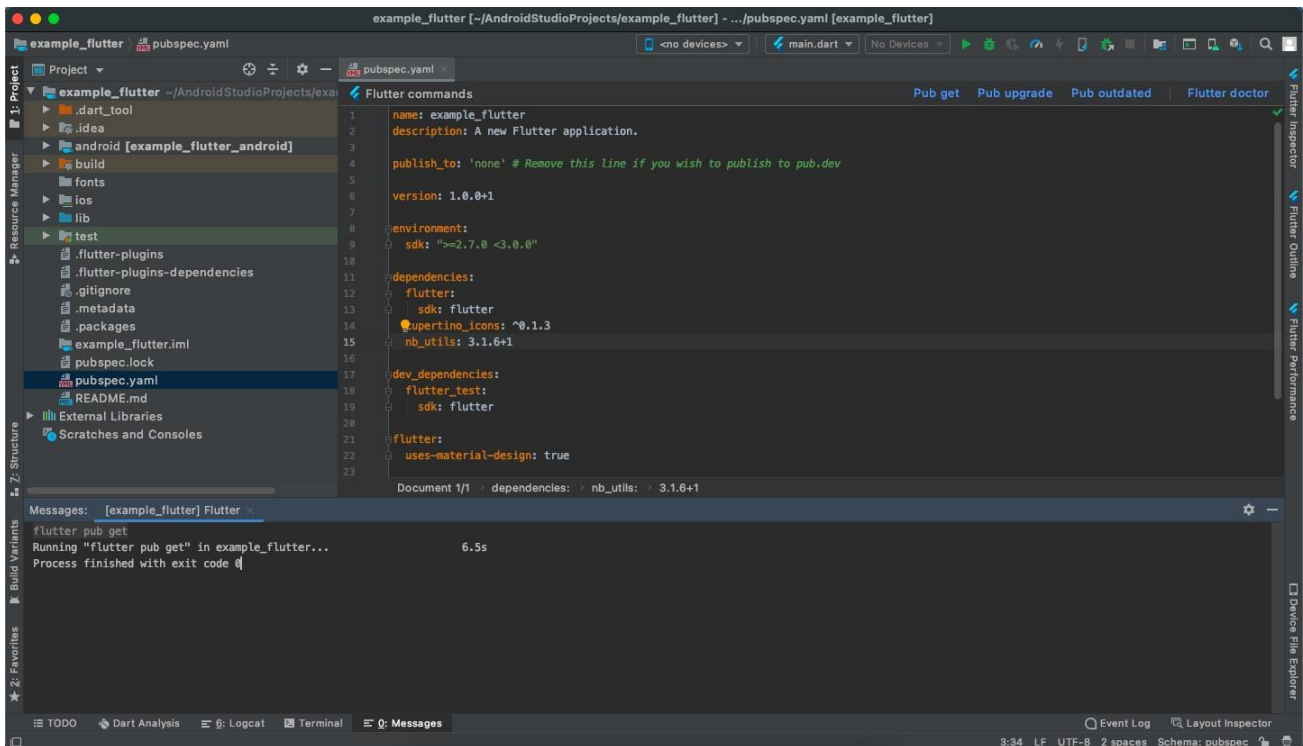
Get Dependencies

After you loaded project successfully, run the following command in the terminal to install all the dependencies listed in the `pubspec.yaml` file in the project's root directory or just click on Pub get in pubspec.yaml file if you don't want to use command.


```
flutter pub get
```

Important

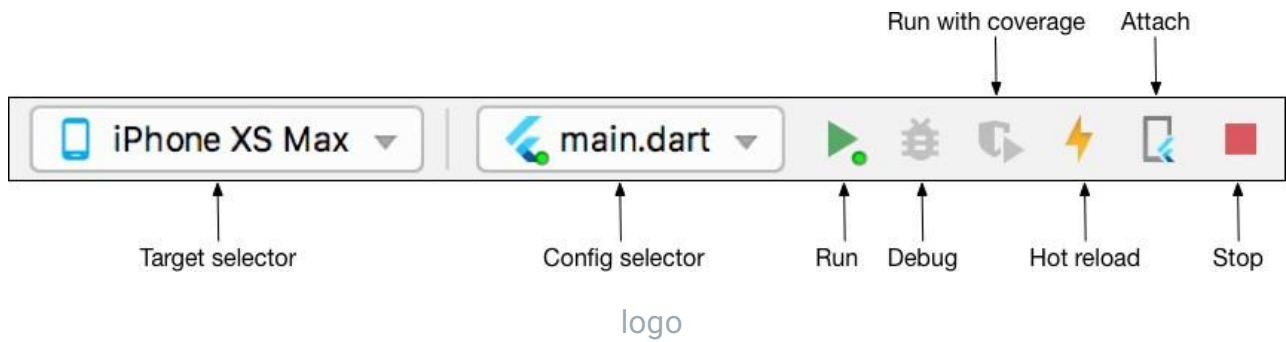
All below steps are must be followed to build and run application



logo

Build and Run App

1. Locate the main Android Studio toolbar.
2. In the target selector, select an Android device for running the app. If none are listed as available, select Tools > Android > AVD Manager and create one there. For details, see [Managing AVDs](#).
3. Click the run icon in the toolbar, or invoke the menu item Run > Run.



After the app build completes, you'll see the app on your device.

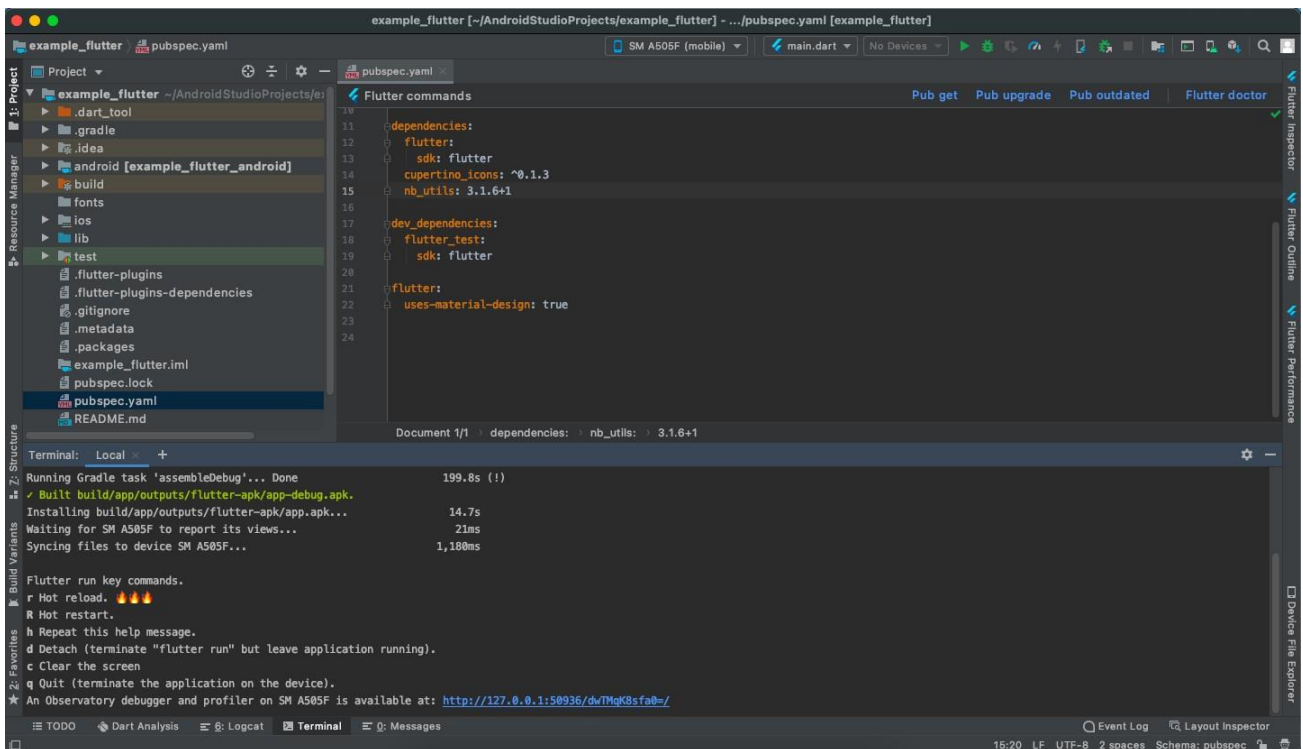
If you don't use Android Studio or IntelliJ you can use the command line to run your application using the following command

Important

Below step requires flutter path to be set in your Environment variables. See <https://flutter.dev/docs/get-started/install/windows>

```
flutter run
```

You will see below like screen after you have build your app successfully.



logo

Try hot reload

Flutter offers a fast development cycle with Stateful Hot Reload, the ability to reload the code of a live running app without restarting or losing app state. Make a change to app source, tell your IDE or command-line tool that you want to hot reload, and see the change in your simulator, emulator, or device.

Important

Do not stop your app. let your app run.

PROJECT STRUCTURE & FEATURES

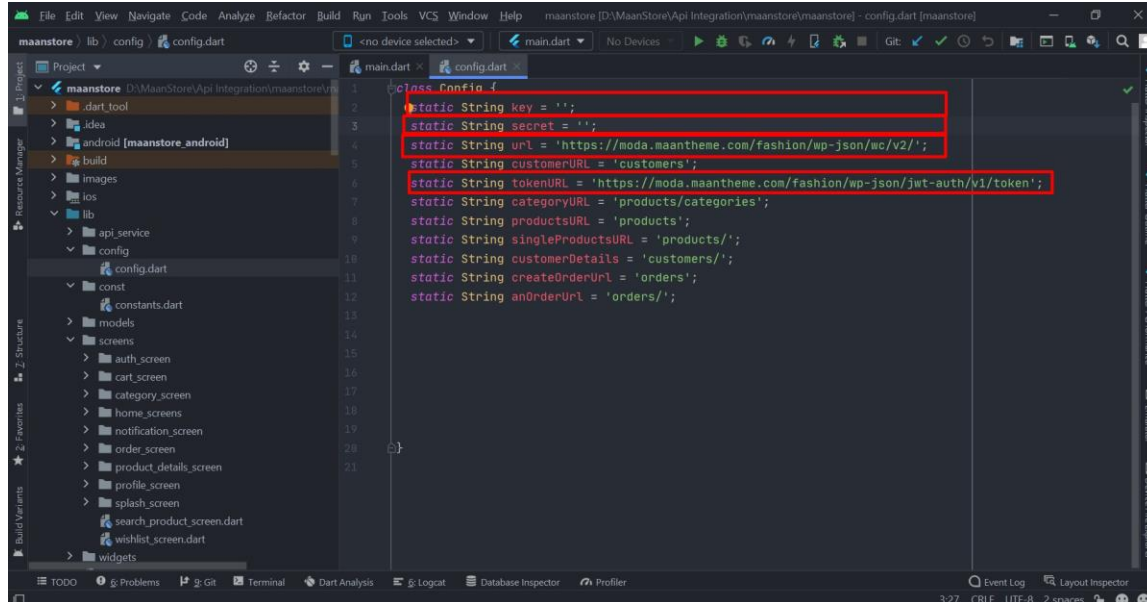
- **images:** apps/widgets/snippets images as well as applcon are stored here.
- **lib:** Application main file and folder are located here.
 - **Screens:** Full Apps UI Code can be found here.
 - **Components:** Re-useable widget can be found here
 - **constant:** All constant value are located here
 - **main:** All the routes are been declared here.
- **pubspec.yaml:** Application name and other project dependencies can be found here.

Application Features

- Clean Code and a well structured project
- Single code base for both Android & iOS
- 60 FPS Support for both Android & iOS
- Fully responsive UI
- Best UI & UX
 - Dark and light modes
- Great animation
- Easy to customize
- Free life time updates & stunning customer supportEasy
- to integrate in your project

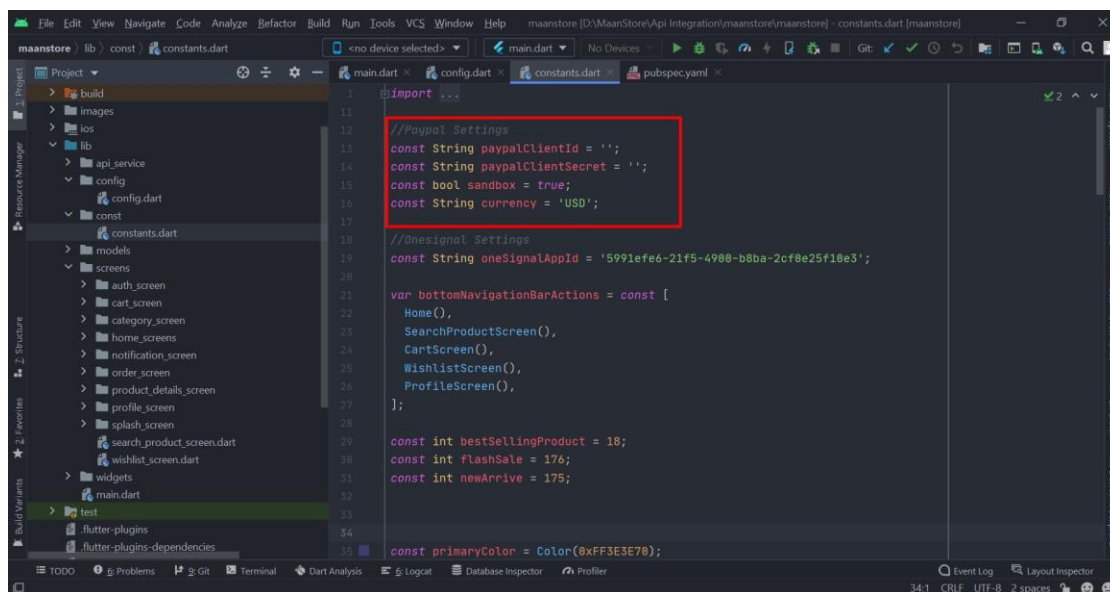
API Setu0:

Please go to the file lib > Config > config.dart and add your website link with /wp-json/wc/v3/ at the end.

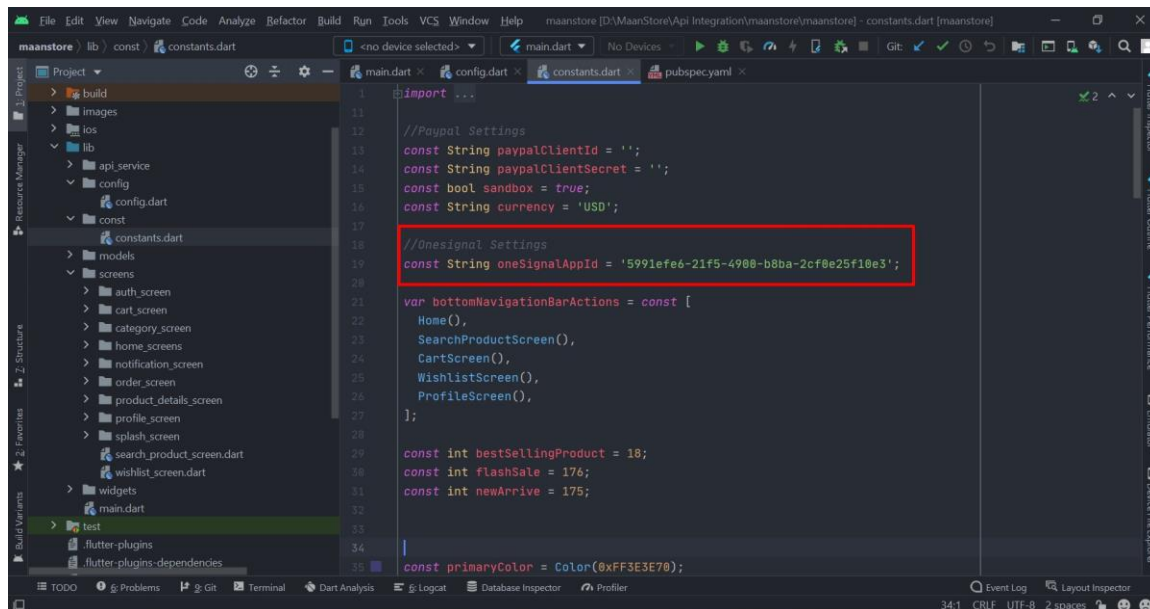


Pay0al Integration:

Add your PayPal clientId and secret at lib > const > constants.dart and set to sandbox true or false based on your use case. Also, change the currency and currency icon for payment. The currency icon will be changed in the whole app if you change it here.



Onesignal Integration: Add your Onesignal appId at lib > const > constants.dart



The screenshot shows an IDE window with the file `constants.dart` open. The file is located at `lib > const > constants.dart` in the project structure. The code contains the following sections:

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

//Paypal Settings
const String paypalClientId = '';
const String paypalClientSecret = '';
const bool sandbox = true;
const String currency = 'USD';

//Onesignal Settings
const String oneSignalAppId = '5991efe6-21f5-4900-b8ba-2cf0e25f10e3';

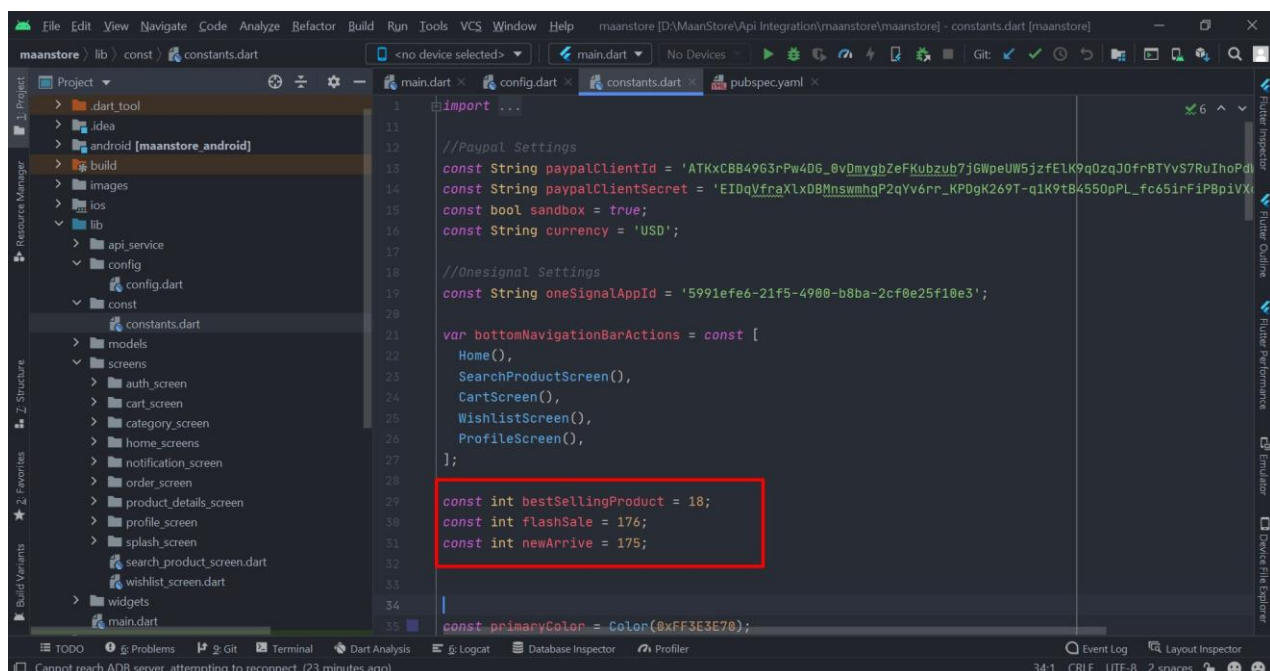
var bottomNavigationBarActions = const [
  Home(),
  SearchProductScreen(),
  CartScreen(),
  WishListScreen(),
  ProfileScreen(),
];

const int bestSellingProduct = 18;
const int flashSale = 176;
const int newArrive = 175;

const primaryColor = Color(0xFF3E3E70);
```

The `//Onesignal Settings` section is highlighted with a red box.

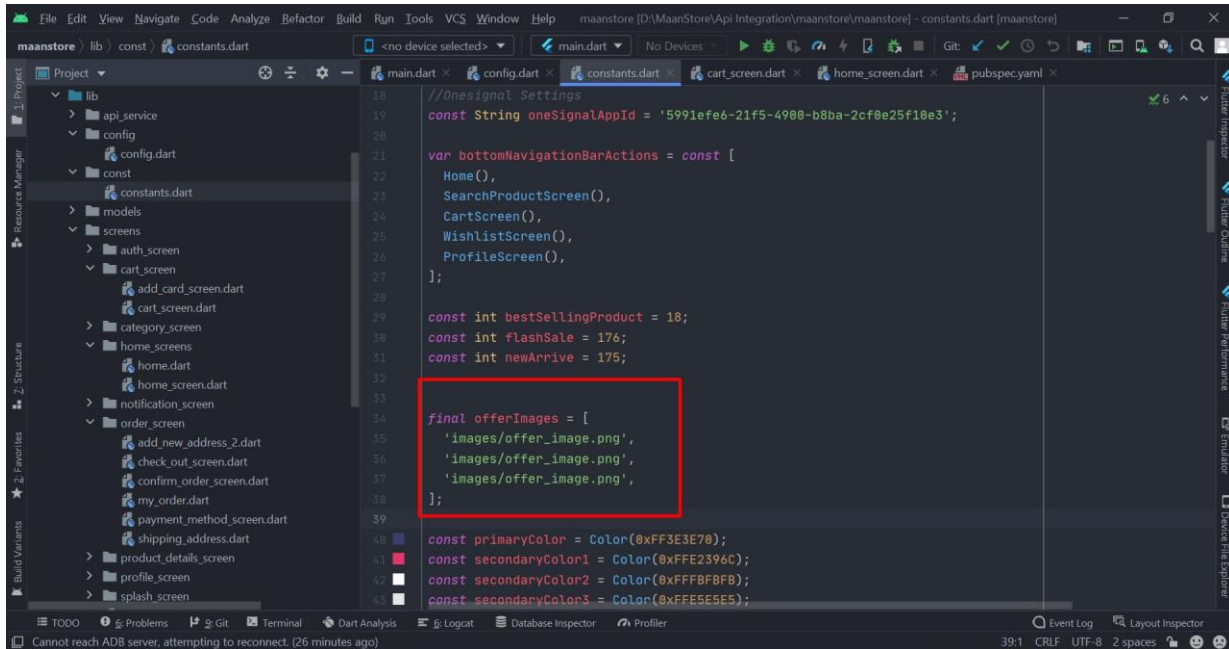
Home Screen Sections Configuration: Change the category ID which you want to show on your home screen at the best selling, flash sale and new arrival section. Add those Category Id at lib > const > constants.dart



The screenshot shows the same IDE window with the `constants.dart` file. The code is identical to the previous screenshot, but the `bestSellingProduct`, `flashSale`, and `newArrive` constants are highlighted with a red box:

```
const int bestSellingProduct = 18;
const int flashSale = 176;
const int newArrive = 175;
```

Change The banners: To change the banners on the home screen, add your banner images to the images folder. Add your image name one by one at lib > const > constants.dart



```
18 //OneSignal Settings
19 const String oneSignalAppId = '5991efe6-21f5-4908-b8ba-2cf0e25f10e3';
20
21 var bottomNavigationBarActions = const [
22   Home(),
23   SearchProductScreen(),
24   CartScreen(),
25   WishlistScreen(),
26   ProfileScreen(),
27 ];
28
29 const int bestSellingProduct = 18;
30 const int flashSale = 176;
31 const int newArrive = 175;
32
33 final offerImages = [
34   'images/offer_image.png',
35   'images/offer_image.png',
36   'images/offer_image.png',
37 ];
38
39 const primaryColor = Color(0xFF3E3E70);
40 const secondaryColor1 = Color(0xFFE2396C);
41 const secondaryColor2 = Color(0xFFFFB8F8);
42 const secondaryColor3 = Color(0xFFE5E5E5);
```

HELP & SUPPORT

Free support policy includes troubleshooting, technical assistance with the product only. It does not include any customization, additional features integration or concerns about third-party plugins compatibility. But, support is applied to plugin(s) we have developed and integrated ourselves. We appreciate your understanding!

If you need assistance and information on purchased product that is not covered in documentation, mail them on our support at

 maantheme@gmail.com

Skype: <https://join.skype.com/invite/kEPqImF1Vfqk>

You can expect answer within 24-48 hours, usually as soon as possible in the order they were received.

Additionally, if any email has no response from the item owner for more than 7 days, that support email will be archived by default. However, if you need further assistance, you can mail us again.