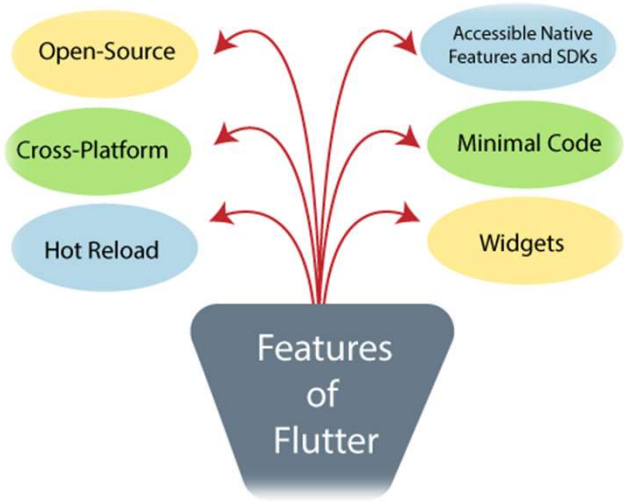
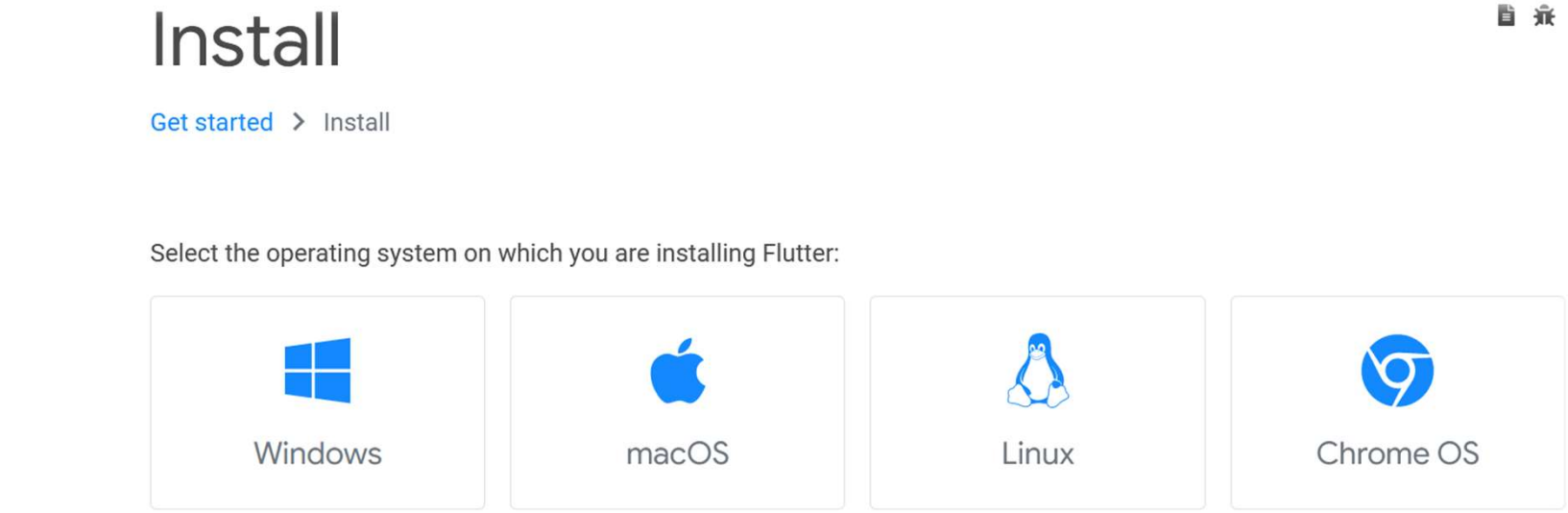
Flutter

By Rabbil Hasan

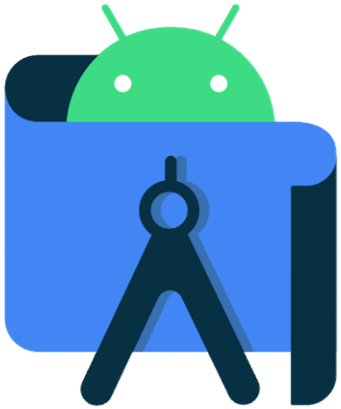
Flutter Features



Flutter Installation

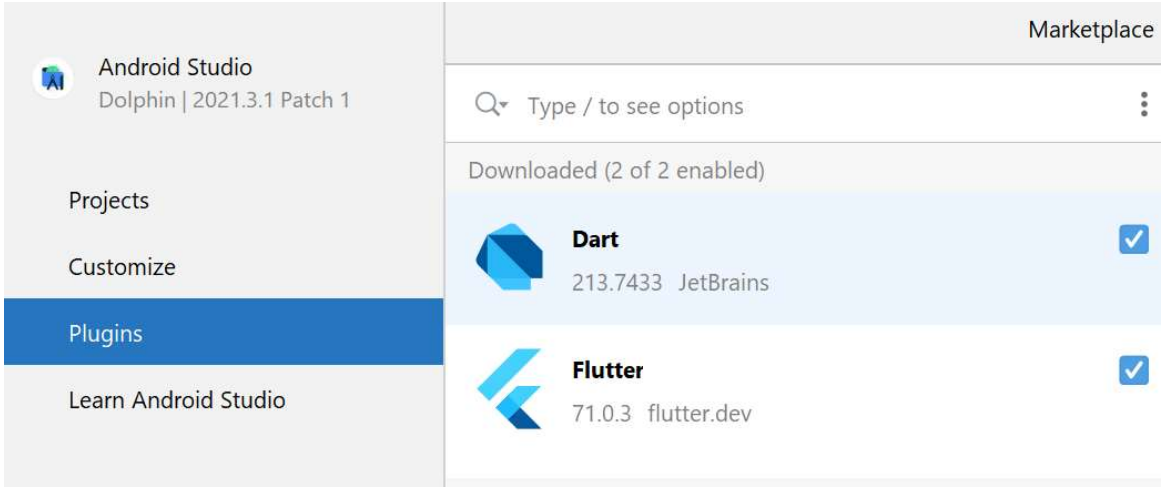


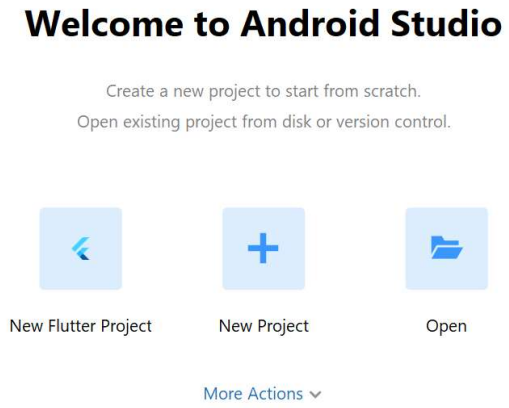
Install Android Studio



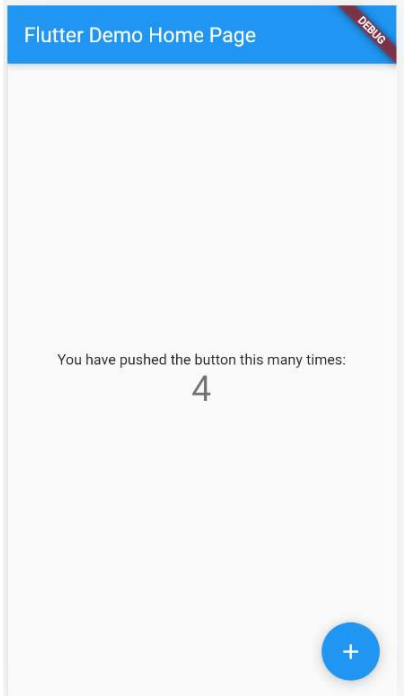
Get the official Integrated Development Environment (IDE) for Android app development.

Install Flutter Extensions





Create & Run Your First Flutter App



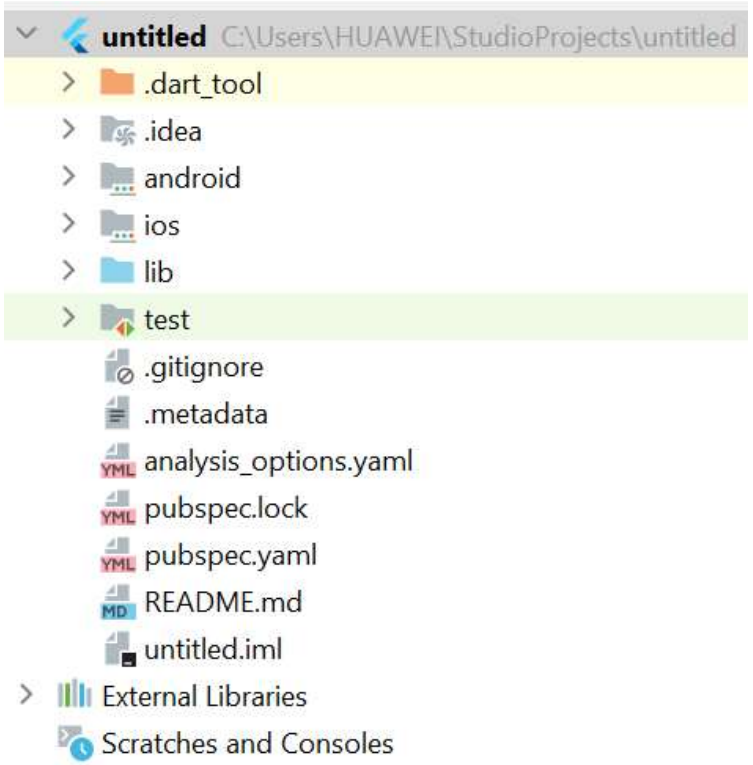
Flutter Project Structure

.idea:

- This folder is at the very top of the project structure.
- Holds the configuration for Android Studio.
- It doesn't matter because we are not going to work with.
- So that the content of this folder can be ignored.

.android:

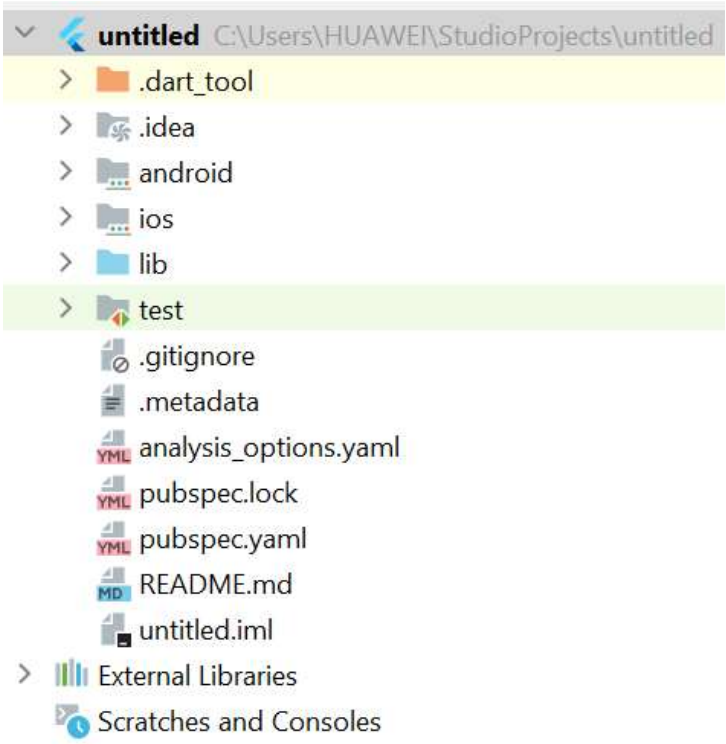
- This folder holds a complete Android project
- Used when you build the Flutter application for Android
- When the Flutter code is compiled into the native code
- It will get injected into this Android project
- That the result is a native Android application



Flutter Project Structure

.ios:

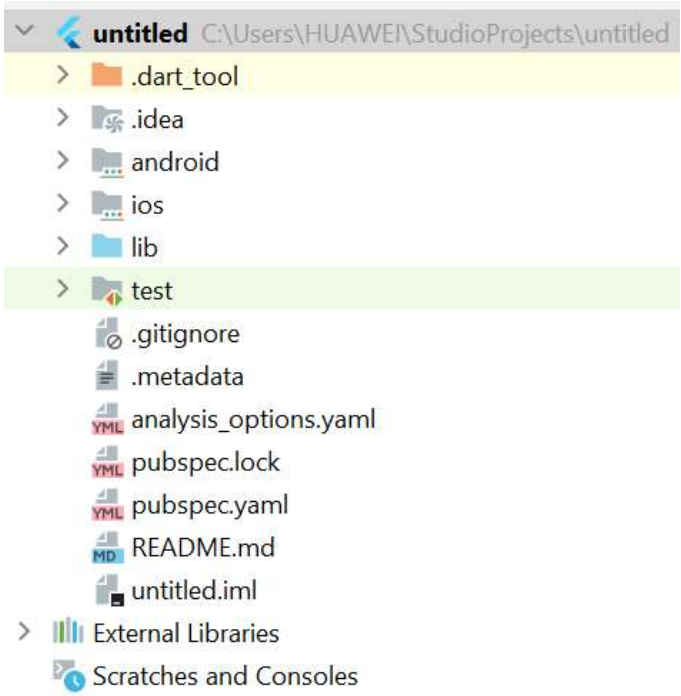
- This folder holds a complete Mac project
- used when you build the Flutter application for iOS
- When the Flutter code is compiled into the native code
- It will get injected into this iOS project
- that the result is a native iOS application
- Building a Flutter application for iOS is only possible when you are working on macOS



Flutter Project Structure

lib:

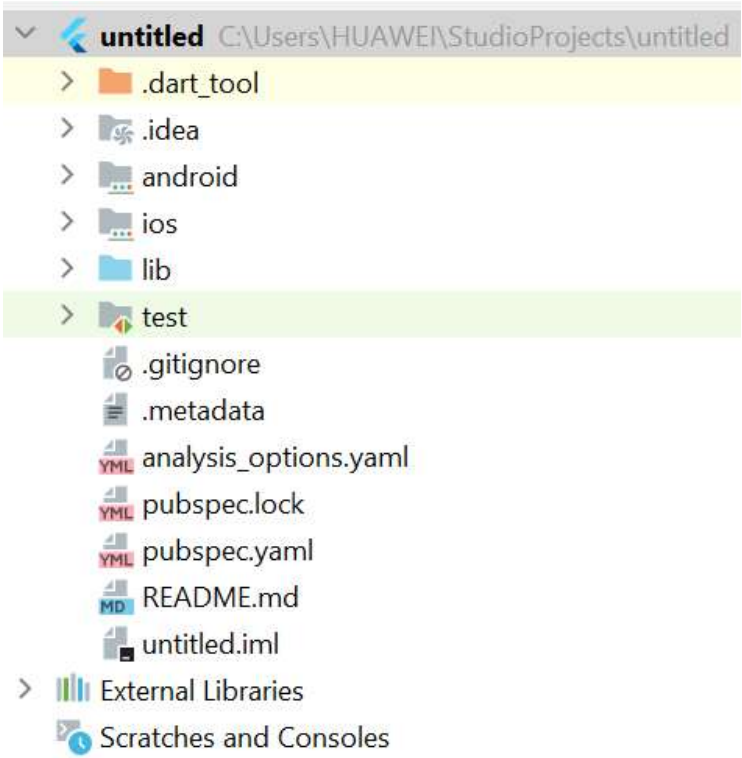
- It is an essential folder, which stands for the library
- It is a folder where we will do our 99 percent of project work
- Inside the lib folder, we will find the Dart files which contain the code of our Flutter application
- By default, this folder contains the file main.dart, which is the entry file of the Flutter application.



Flutter Project Structure

test:

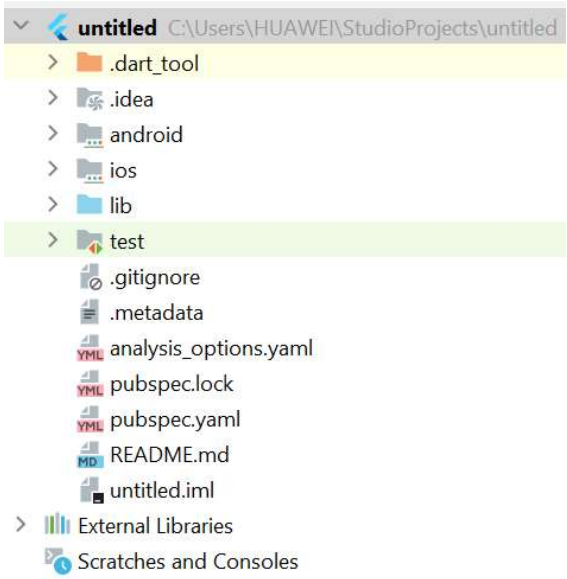
- This folder contains a Dart code,
- Which is written for the Flutter application.
- Perform the automated test when building the app.
- It won't be too important for us here.



Flutter Project Structure

.gitignore:

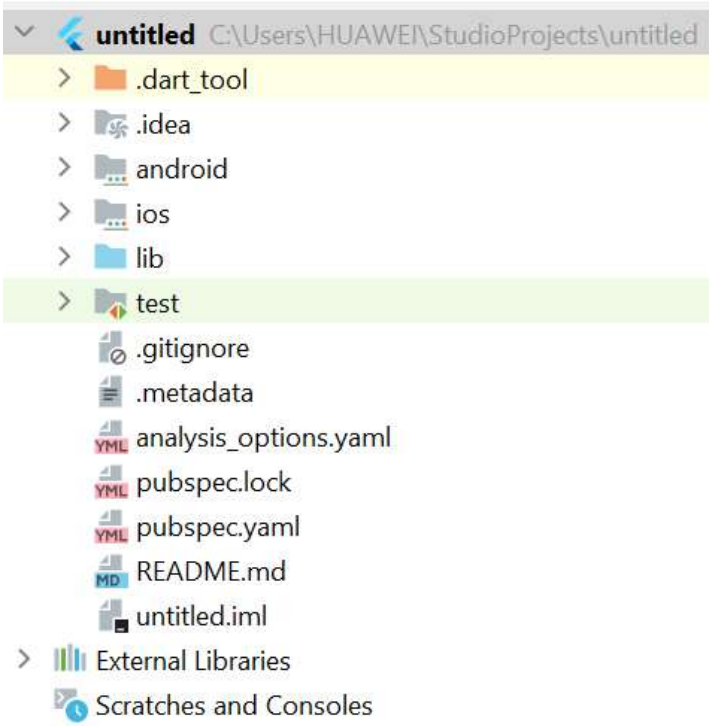
- It is a text file containing a list of files, file extensions, and folders
- that tells Git which files should be ignored in a project.
- Git is a version-control file for tracking changes in source



Flutter Project Structure

.metadata:

- It is an auto-generated file by the flutter tools.
- Used to track the properties of the Flutter project.
- This file performs the internal tasks.
- So you do not need to edit the content manually at any time.

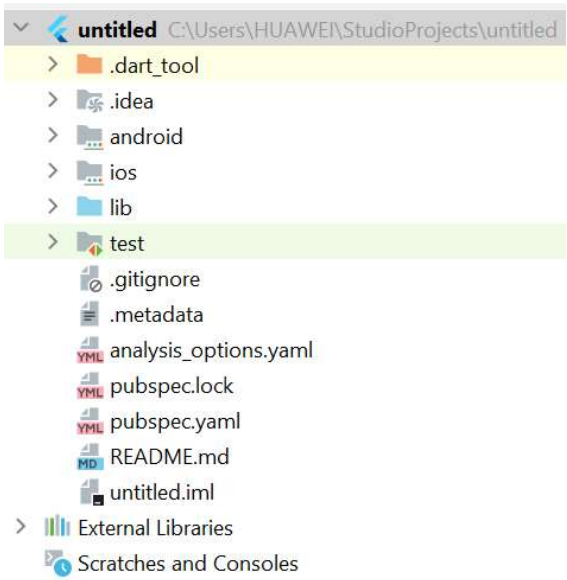


Flutter Project Structure

pubspec.yaml:

It is the project's configuration file that will use a lot during working with the Flutter project. It allows you how your application works. This file contains-

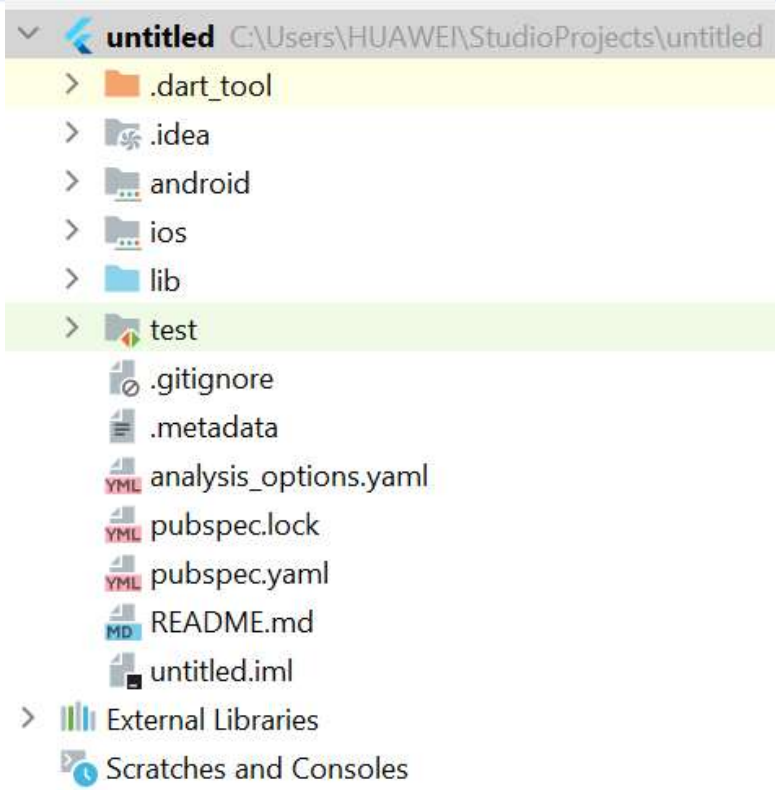
- Project general settings such as name
- Description, and version of the project.
- Project dependencies.
- Project assets (e.g., images).



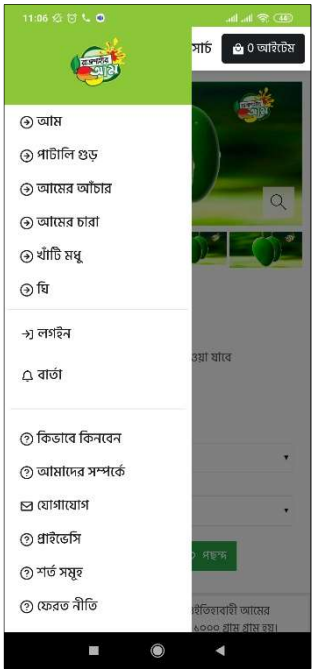
Flutter Project Structure

pubspec.lock:

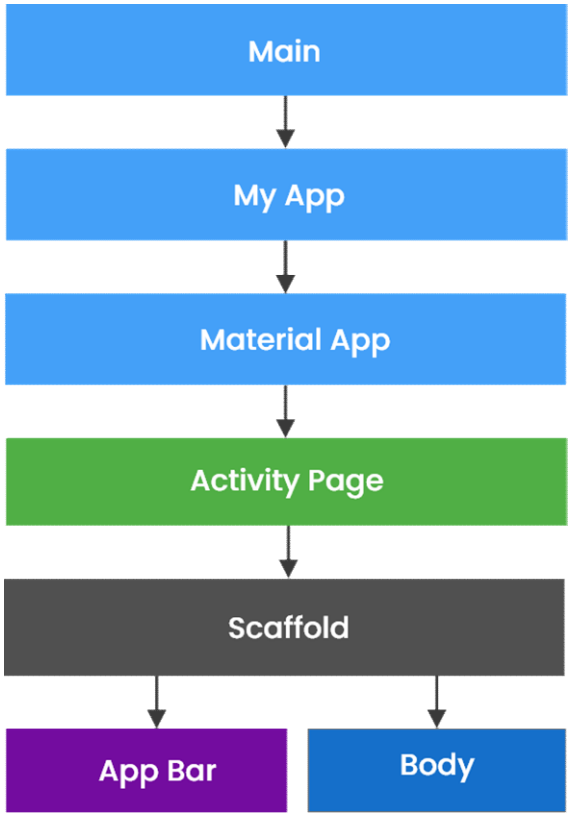
- It is an auto-generated file based on the .yaml file.
- It holds more detail setup about all dependencies.



Activity & widgets concepts



Flutter Main Source Code Flow




```dart
void main(){
  runApp(const MyApp());
}

class MyApp extends StatelessWidget{
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
     return const MaterialApp(home: HomeActivity());
  }
}

class HomeActivity extends StatelessWidget{
  const HomeActivity({super.key});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Hello')),
      body: const Text('Hello World') ,
    );  // Scaffold
  }
}
```

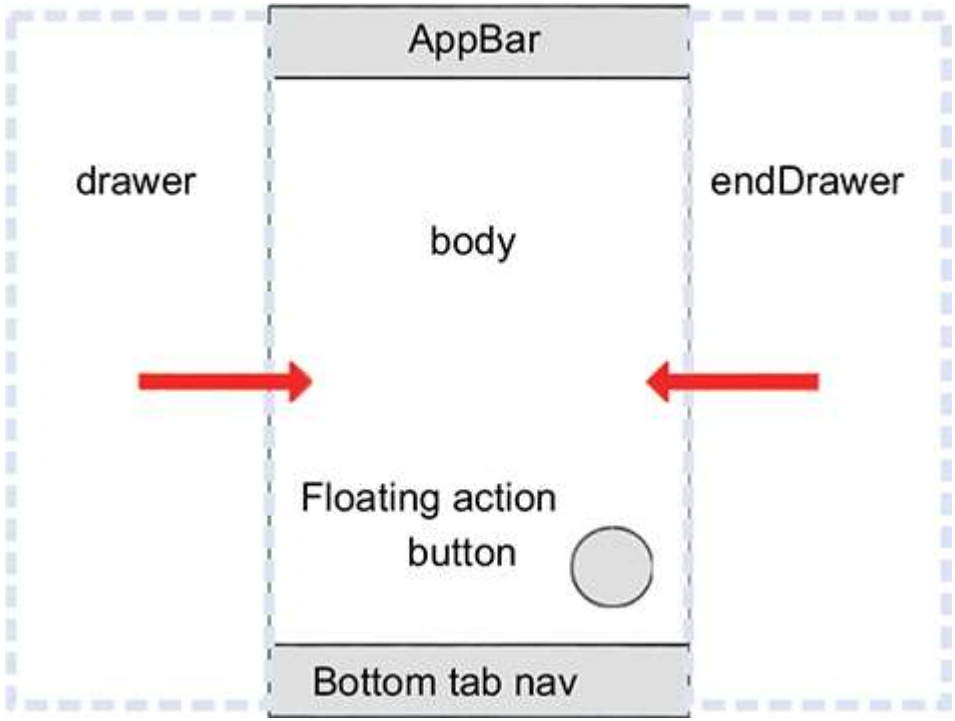
Material App Class

MaterialApp is a predefined class in a flutter. Main or core component of flutter.

- **color:** It controls the primary color used in the application.
- **darkTheme:** It provided theme data for the dark theme for the application.
- **debugShowCheckedModeBanner:** This property takes in a boolean as the object to decide whether to show the debug banner or not.
- **home:** This property takes in widget as the object to show on the default route of the app.
- **title:** The title property takes in a string as the object to decide the one-line description of the app for the device.

Scaffold Class

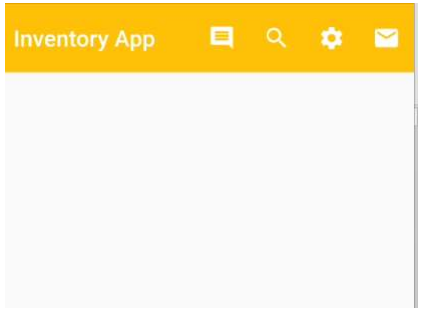
Scaffold will expand or occupy the whole device screen.



Flutter AppBar Widget

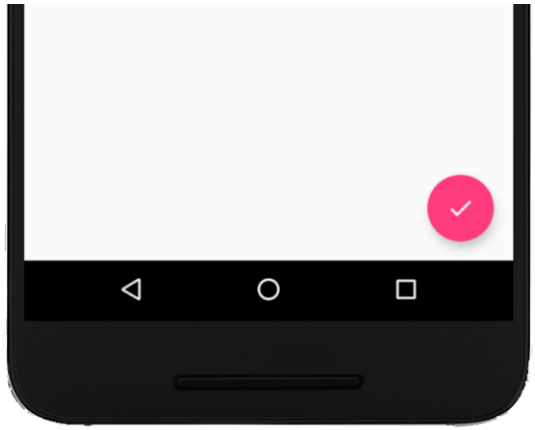
AppBar is usually the topmost component of the app . it contains the toolbar and some other common action buttons.

- **actions:** This property takes in a list of widgets as a parameter to be displayed after the title if the **AppBar** is a row.
- **title:** This property usually takes in the main widget as a parameter to be displayed in the **AppBar**.
- **backgroundColor:** This property is used to add colors to the background of the **Appbar**.
- **elevation:** This property is used to set the z-coordinate at which to place this app bar relative to its parent.
- **shape:** This property is used to give shape to the **Appbar** and manage its shadow.



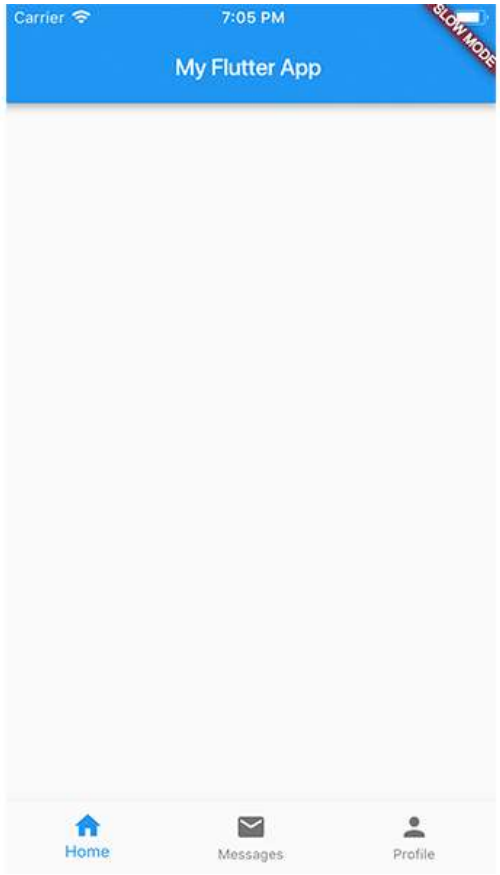
Flutter Floating Action Button

- Floating Action Button is a button that is placed at the right bottom corner by default.
- Floating Action Button is an icon button that floats over the content of the screen at a fixed place.

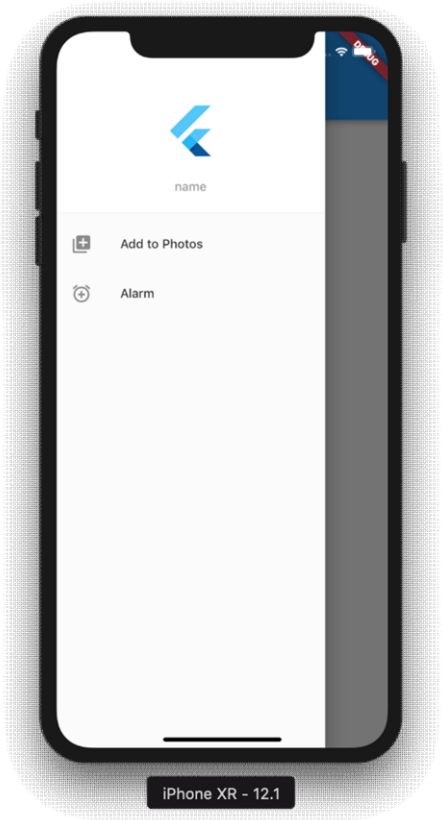


Flutter Bottom Tab

- Menu at the bottom of the Scaffold.
- We have seen this Navigation bar in most of the applications.
- We can add multiple icons or texts or both in the bar as items.



Navigation drawer flutter



Navigation end drawer flutter