Flutter 101

An fastboot to Flutter

Content

1. What is flutter, who made it, who uses it and why should I use it?
2. Installation (Linux Only)
3. Flutter cli
4. Understanding the Architecture
5. The main.dart
6. Some Ideologies
7.

What is flutter, who made it, who uses it and why should I use it?

Flutter is a free and open source SDK developed by google to develop native applications for Android and iOS using one code base written in dart.

It supports Material 2.0 standards and provides twice better performance than React Native.

The primary usefulness of flutter is that its extremely fast to develop with. We can build quality applications using flutter with much less code. It also has a significant amount of libraries available within a year of its launch and has been in top ranks in GitHub and Stackoverflow which indicates bright future for this framework.

Installation (Linux Only)

https://storage.googleapis.com/flutter_infra/releases/beta/linux/flutter_linux_v0.5.1-beta.tar.xz

Download the sdk and extract it. Add the path to its bin in environment's PATH.

Install Android SDK and set ANDROID_HOME variable in .bashrc to Android/Sdk's installation path.

Run flutter doctor after connecting an android mobile via USB to Computer.

If any issues resolve with given instructions as per the flutter doctor's output else installation is successful.

Flutter-cli

Flutter just like other frameworks ships with a cli. We can use it for creating projects. Building apks and ipas and also for debugging applications.

```
flutter create APP_NAME ← creates an app with name APP_NAME

cd APP_NAME

flutter run ← build, install and run application on device in debug mode.
```

Understanding The Architecture

pubspec.yaml ← specifications of project. eg. External packages being used.

lib ← Main code lies here

test ← test cases for main code

build ← the files built while testing and while building final releases

android ← created files for android application

ios ← created files for ios application

assets ← we can add this folder to store static media.

The main.dart

main.dart is the entry file for our flutter application.

Main.dart contains a method main which expects an instance of a widget.

(Everything is widget in flutter.)

Widgets can be of two types depending on need-

1. Stateless (if only static data needs to be shown)
2. Stateful (if data needs to be dynamic)

Stateless as it sounds, is immutable.

Some ideologies

1. Everything is a widget.
2. Composition is superior than Inheritance.

   Mostly we will see that every widget is made up of small other widgets.

3. Layering should be prefered-

Internal architecture of flutter contains two main layers - Engine Layer and Framework Layer. Engine Layer contains 3 things- Dart, Skia and Text.

Whereas Framework Layer contains, Material on top of Cuperterino on top of Widgets on top of Rendering on top of Animations & Gestures.

Stateless Widgets

A class that is widget that is immutable.

It describes a part of UI.

It must override a method named build whose return type should be Widget.

```
class XYZ extends StatelessWidget {

    @override

    Widget build(BuildContext context){ }

}
```

Stateless Widgets contd..

All the widgets are kept in a Widget tree in flutter engine, the argument passed to Stateless Widgets is the position of the widget in that widget tree.

We can return any Container from the build methods. Eg. the container class itself?

But most of the times we return a MaterialApp instance from main widget for standard UI.

MaterialApp Class

MaterialApp class declares an outline for our material guidelines based app.

It must declare a home property. Which is basically the view that shows up on root path ie. When the app starts.

Home also requires a scaffolding for other components. This is provided by Scaffold class.