Basics of Widgets

Flutter Tutorial

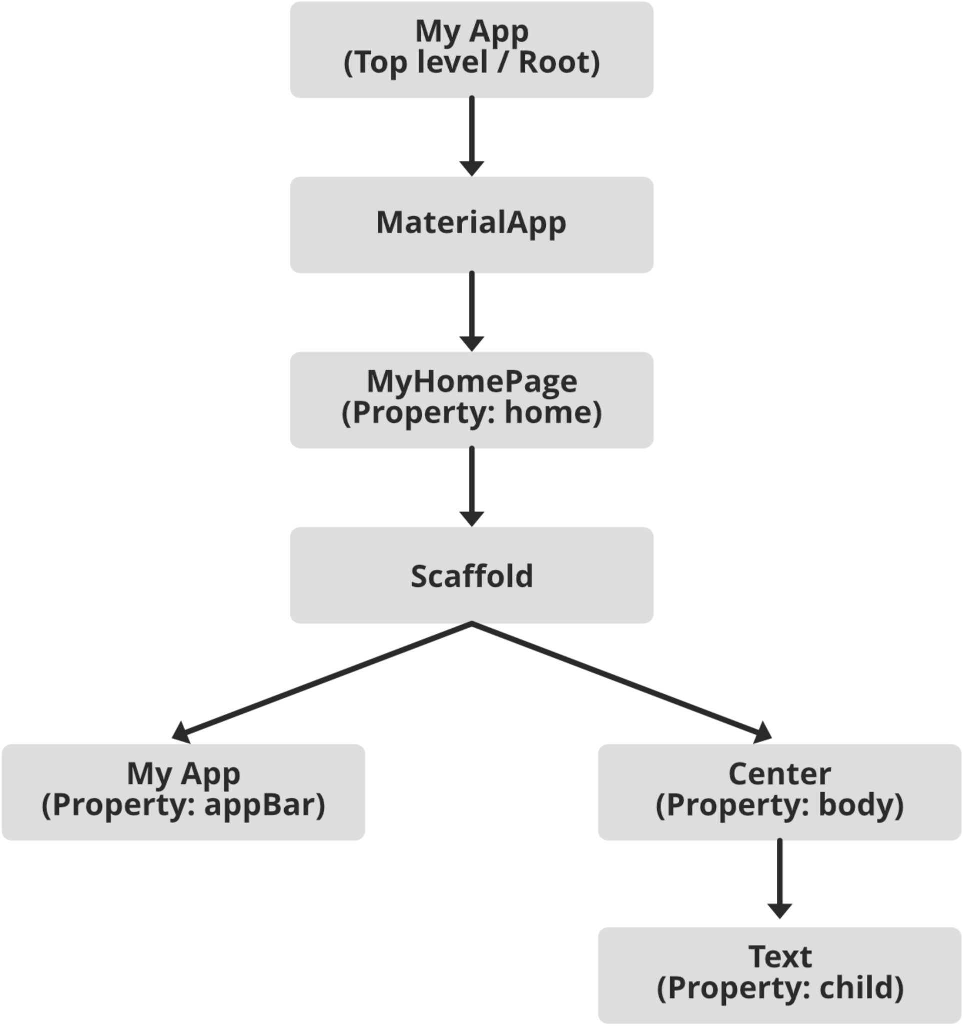
Flutter – Architecture Application

Flutter architecture application mainly consists of:

- **Widgets**
- **Gestures**
- **Concept of State**
- **Layers**

Widgets

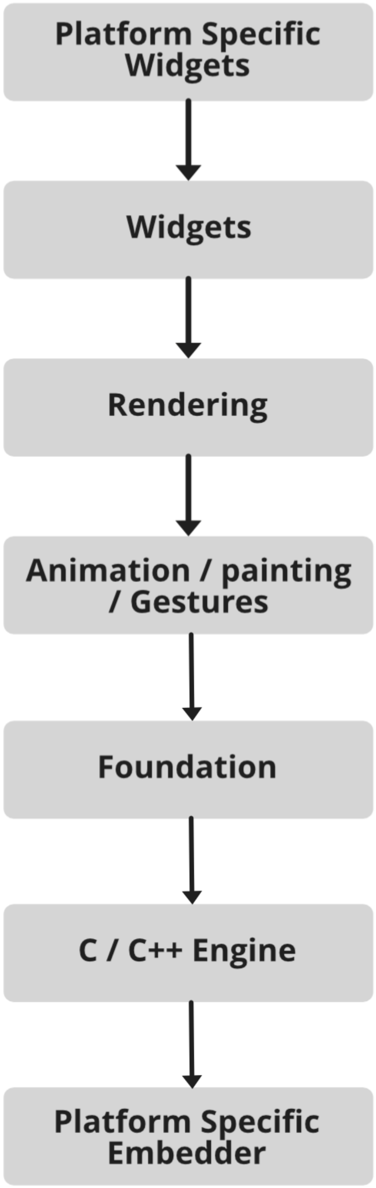
Widgets are the primary component of any flutter application. It acts as a UI for the user to interact with the application. Any flutter application is itself a widget that is made up of a combination of widgets. In a standard application, the root defines the structure of the application followed by a Material App widget which basically holds its internal components in place. This is where the properties of the UI and the application itself is set. The Material App has a Scaffold widget that consists of the visible components (widgets) of the application. The Scaffold has two primary properties namely the body and appbar. It holds all the child widgets and this is where all its properties are defined. The below diagram shows the hierarchy of a flutter application:.



Inside Scaffold, there is usually an appbar widget, which as the name suggests define appbar of the application. The scaffold also has a body where all the component widgets are placed. This is where these widget's properties are set. All these widgets combined form the Homepage of the application itself. The Center widget has a property, Child, which refers to the actual content and it is built using the Text widget.

Layers

The Flutter framework is categorized based on its complexity and establishes a hierarchy based on the decreasing level of these complexities. These categories are often called Layers. These layers are built on top of one another. The topmost layer is a widget specific to the operating system of the device (ie, Android or iOS). The second layer consists of the native flutter widgets, which comprise structural UI components, gesture detectors, state management components, etc. This third layer is where all the Ui and state rendering occurs. It is the layer that includes all the visible components of the flutter application. The following layer consists of animations used in transitions, image flow, and gestures. These further go on to the very high level of system design that is not the target of this article. The below diagram gives an overview of the same:


Platform Specific Widgets
Widgets
Rendering
Animation / painting / Gestures
Foundation
C / C++ Engine
Platform Specific Embedder

Gestures

All physical form of interaction with a flutter application is done through pre-defined gestures. Gesture-Detectors are used for the same. It is an invisible widget that is used to process physical interaction with the flutter application. The interaction includes gestures like tapping, dragging, and swiping, etc. These features can be used to creatively enhance the user experiences of the app by making it perform desired actions based on simple gestures.

Concept of State

If you have ever worked with React-js, you might be familiar with the concept of a state. The states are nothing but data objects. Flutter also operates on similar turf. For the management of state in a Flutter application, Stateful-Widget is used. Similar to the concept of state in React-js, the re-rendering of widgets specific to the state occurs whenever the state changes. This also avoids the re-rendering of the entire application, every time the state of a widget changes.

The architecture of a Flutter app or the flutter framework generally consists of a combination of small and larger widgets interacting in conjuncture to build the application. All of its layers are integral to its design and functioning. As simple as building an application in flutter is, it is built with equally complex components at its core.

What is widgets in Flutter?

Flutter is Google's UI toolkit for crafting beautiful, natively compiled iOS and Android apps from a single code base. To build any application we start with widgets – The building block of flutter applications. Widgets describe what their view should look like given their current configuration and state. It includes a text widget, row widget, column widget, container widget, and many more.

Widgets: Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. And the structure of the code of an apps is a tree of widgets.

Category of Widgets:

There are mainly 14 categories in which the flutter widgets are divided. They are mainly segregated on the basis of the functionality they provide in a flutter application.

1. **Accessibility:** These are the set of widgets that make a flutter app more easily accessible.
2. **Animation and Motion:** These widgets add animation to other widgets.
3. **Assets, Images, and Icons:** These widgets take charge of assets such as display images and show icons.
4. **Async:** These provide async functionality in the flutter application.
5. **Basics:** These are the bundle of widgets that are absolutely necessary for the development of any flutter application.
6. **Cupertino:** These are the iOS designed widgets.
7. **Input:** This set of widgets provides input functionality in a flutter application.
8. **Interaction Models:** These widgets are here to manage touch events and route users to different views in the application.
9. **Layout:** This bundle of widgets helps in placing the other widgets on the screen as needed.
10. **Material Components:** This is a set of widgets that mainly follow material design by Google.
11. **Painting and effects:** This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
12. **Scrolling:** This provides scrollability of to a set of other widgets that are not scrollable by default.
13. **Styling:** This deals with the theme, responsiveness, and sizing of the app.
14. **Text:** This displays text.