

Flutter State Management

Tushar B. Kute,
<http://tusharkute.com>



State

- We know that in Flutter, everything is a widget.
- The widget can be classified into two categories, one is a Stateless widget, and another is a Stateful widget.
- The Stateless widget does not have any internal state. It means once it is built, we cannot change or modify it until they are initialized again.
- On the other hand, a Stateful widget is dynamic and has a state. It means we can modify it easily throughout its lifecycle without reinitialized it again.

State

- A state is information that can be read when the widget is built and might change or modified over a lifetime of the app.
- If you want to change your widget, you need to update the state object, which can be done by using the `setState()` function available for Stateful widgets.
- The `setState()` function allows us to set the properties of the state object that triggers a redraw of the UI.

State

- The state management is one of the most popular and necessary processes in the lifecycle of an application.
- According to official documentation, Flutter is declarative. It means Flutter builds its UI by reflecting the current state of your app.
- The following figure explains it more clearly where you can build a UI from the application state.

$$\text{UI} = f(\text{state})$$

The layout on the screen Your build methods The application state

State

- Let us take a simple example to understand the concept of state management. Suppose you have created a list of customers or products in your app.
- Now, assume you have added a new customer or product dynamically in that list. Then, there is a need to refresh the list to view the newly added item into the record.
- Thus, whenever you add a new item, you need to refresh the list. This type of programming requires state management to handle such a situation to improve performance.
- It is because every time you make a change or update the same, the state gets refreshed.

State

- In Flutter, the state management categorizes into two conceptual types, which are given below:
 - Ephemeral State
 - App State

Ephemeral State

- This state is also known as UI State or local state.
- It is a type of state which is related to the specific widget, or you can say that it is a state that contains in a single widget.
- In this kind of state, you do not need to use state management techniques.
- The common example of this state is Text Field.

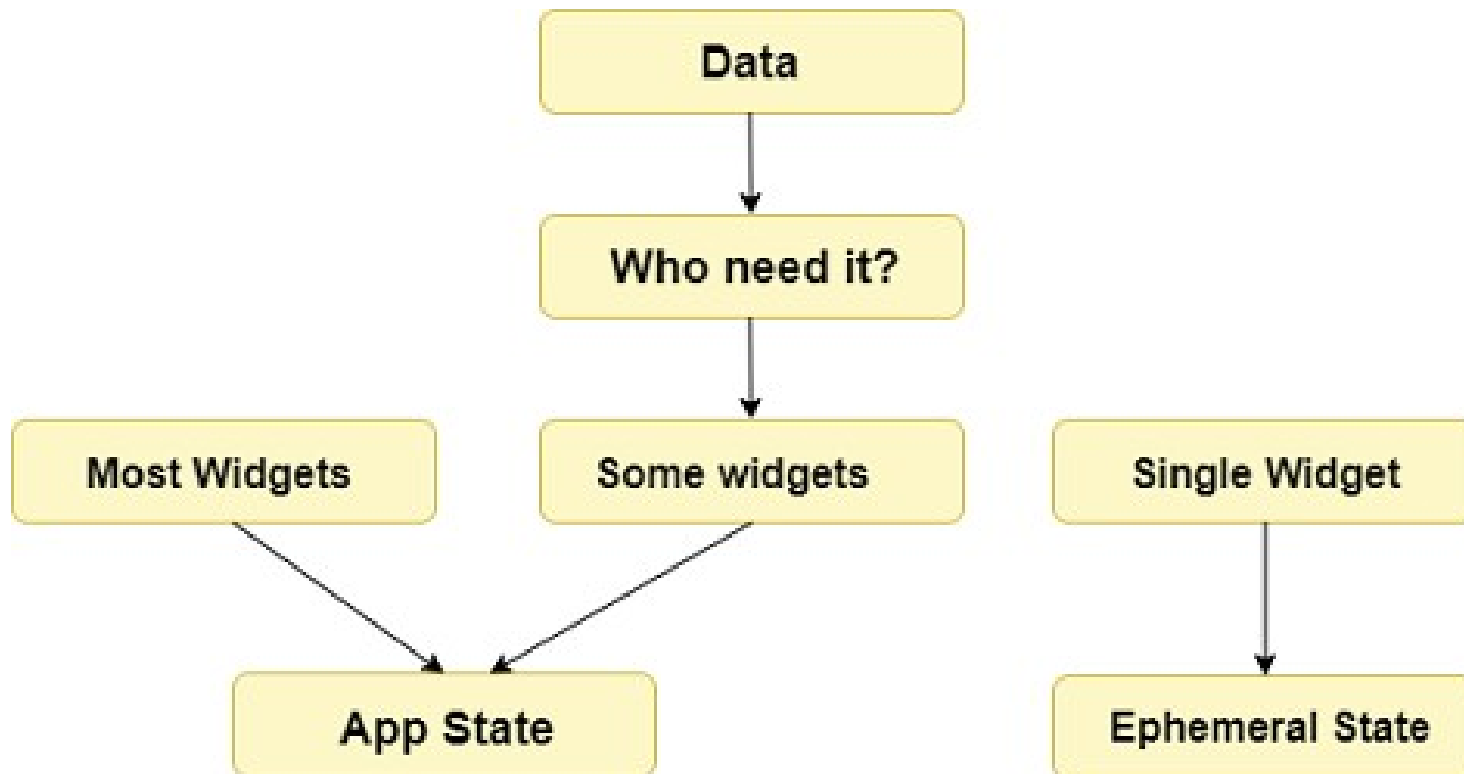
Example

- In the above example, the `_name` is an ephemeral state. Here, only the `setState()` function inside the `StatefulWidget`'s class can access the `_name`.
- The `build` method calls a `setState()` function, which does the modification in the state variables.
- When this method is executed, the widget object is replaced with the new one, which gives the modified variable value.

app State

- It is different from the ephemeral state. It is a type of state that we want to share across various parts of our app and want to keep between user sessions.
- Thus, this type of state can be used globally. Sometimes it is also known as application state or shared state.
- Some of the examples of this state are User preferences, Login info, notifications in a social networking app, the shopping cart in an e-commerce app, read/unread state of articles in a news app, etc.

States



States

- The simplest example of app state management can be learned by using the provider package.
- The state management with the provider is easy to understand and requires less coding.
- A provider is a third-party library. Here, we need to understand three main concepts to use this library.
 - ChangeNotifier
 - ChangeNotifierProvider
 - Consumer

ChangeNotifier

- ChangeNotifier is a simple class, which provides change notification to its listeners.
- It is easy to understand, implement, and optimized for a small number of listeners.
- It is used for the listener to observe a model for changes.
- In this, we only use the `notifyListener()` method to inform the listeners.
- Example.

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License



@mitu_skillologies



/MITuSkillologies



@mitu_group



/company/mitu-
skillologies



MITUSkillologies

Web Resources

<https://mitu.co.in>

<http://tusharkute.com>

contact@mitu.co.in

tushar@tusharkute.com