Stateful Life Cycle

In Flutter, a Stateful widget is a widget that can maintain state and can be rebuilt multiple times throughout its lifetime. The state of a Stateful widget is stored in a State object, which is created by the StatefulWidget's createState method and is persistent across rebuilds of the widget.

The lifecycle of a Stateful widget and its associated State object can be broken down into several stages:

1. **Creation**: When a Stateful widget is instantiated, Flutter calls its **createState** method to create a new instance of the State object. This is typically done by the framework when the widget is first inserted into the widget tree.
2. **Initialization**: Once the State object is created, Flutter calls its **initState** method. This is where you can perform any initialization that is required for the state of the widget, such as setting up event listeners or loading data from a remote source.
3. **Building**: After the initState method has been called, Flutter calls the State object's **build** method to build the widget's UI. This method should return a widget that represents the current state of the widget.
4. **Updating**: When the state of the widget changes, Flutter calls the State object's **setState** method. This method notifies Flutter that the state of the widget has changed and that it needs to be rebuilt. After calling setState, Flutter will schedule a rebuild of the widget, and the **build** method will be called again to update the UI.
5. **Disposal**: When a Stateful widget is removed from the widget tree, Flutter calls its **dispose** method. This is where you can clean up any resources that were allocated by the State object, such as closing open connections or unsubscribing from event listeners.