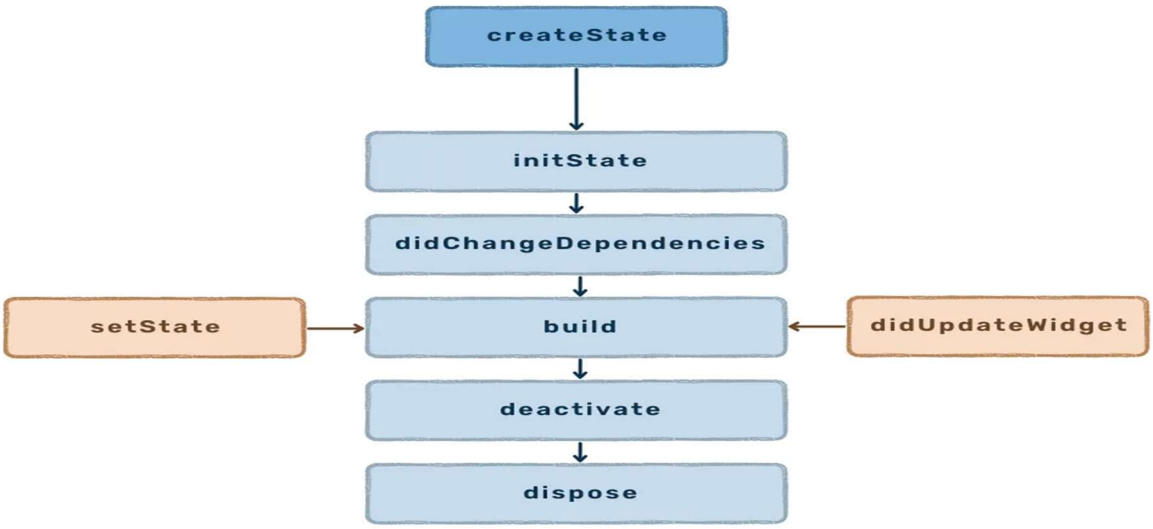
Stateful Lifecycle

A stateful widget in Flutter is a component that can maintain state and update its appearance in response to changes. The lifecycle of a stateful widget consists of a series of methods that are invoked at different stages of its existence.

1. createState : Upon creation of the stateful widget, its constructor is called, which initializes the widget. The createState() method is then invoked, which creates the state object for the widget.
2. initState **:** This method is called after the widget is inserted into the widget tree, when the object is created for the first time.
3. didChangeDependencies **:** This method is called when a dependency of this widget changes. It is useful when the widget depends on some external data or inherits data from its parent widget. It is also called immediately after initState().
4. Build **:** Builds the widget's user interface based on its current state. This method is called initially when the widget is first inserted into the widget tree, and may be called repeatedly during the lifecycle whenever the widget needs to be rebuilt.
5. setState **:** Triggers a rebuild of the widget when the state changes and the associated methods are called again.
6. didUpdateWidget **:** Triggered when the widget is rebuilt with updated properties. It is called after build() and allows you to compare the previous and current widget properties.
7. Deactivate **:** Called when the widget is removed from the tree but might be inserted again.
8. Dispose **:** Called when the widget is removed from the widget tree permanently, allowing you to release resources held by the widget.
9. Reassemble **:** Called when the application is reassembled during hot reload.



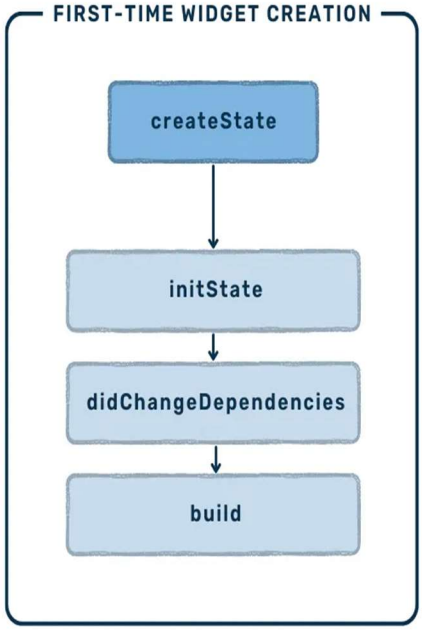
Understanding these lifecycle methods is crucial for managing state in Flutter applications efficiently. They help you control the behavior of your widget throughout its existence, ensuring that resources are used wisely and the UI reflects the latest state.

Demonstrating each lifecycle method:

```
import 'package:flutter/material.dart';

class ExampleStatefulWidget extends StatefulWidget {

@override

_ExampleState createState() => _ExampleState();

}

class _ExampleState extends State<ExampleStatefulWidget> {
```

1. createState

```
  @override

  State<StatefulWidget> createState() =>
_ExampleState();
```



2. initState

```
  @override

  void initState() {

    super.initState();

    print("Init State: Widget is created");

  }
```

3. didChangeDependencies

```
  @override

  void didChangeDependencies() {

    super.didChangeDependencies();

    print("Dependencies Changed");

  }
```

4. build

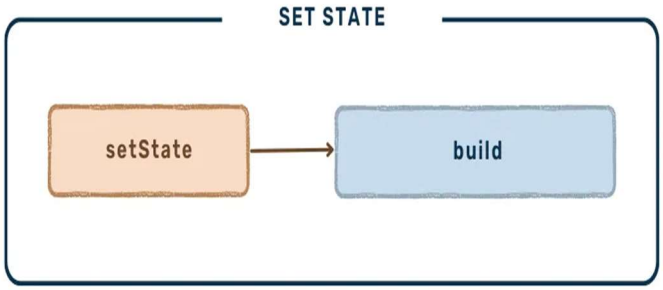
```
@override
Widget build(BuildContext context) {
  print("Build: UI is constructed");
  return Scaffold(
    appBar: AppBar(
      title: Text('Example StatefulWidget'),
    ),
    body: Center(
      child: Text('Hello, Flutter!'),
    ),
  );
}
```

5. didUpdateWidget

```
@override
void didUpdateWidget(covariant ExampleStatefulWidget oldWidget) {
  super.didUpdateWidget(oldWidget);
  print("Widget Updated");
}
```

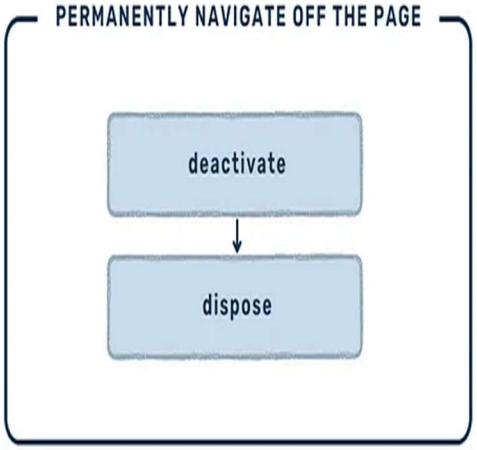
6. setState

```
void updateText() {
  setState(() {
    print("Set State: Internal state changed");
  });
}
```



7. deactivate

```
  @override
  void deactivate() {
    print("Deactivate: Widget removed from the
tree");
    super.deactivate();
  }
```



8. dispose

```
  @override
  void dispose() {
    print("Dispose: Widget permanently removed");
    super.dispose();
  }
}
```

9. reassemble

```
@override
  void reassemble() {
    super.reassemble();
    print('reassemble') ; }
```