1. **createState**
2. **initState**
3. **build**
4. **didUpdateWidget**
5. **setState**
6. **deactivate**
7. **dispose**

Let's break down how each stage is applied in the CurrencyConverterPage widget:

1. **createState**:
   * Called when the CurrencyConverterPage widget is created. It creates an instance of \_CurrencyConverterPageState.
2. **initState**:
   * Called once when the state object is created.
   * It's used to initialize state variables or perform any initial setup.
   * Example from your app:

dart

Copy code

final TextEditingController \_controller = TextEditingController();

1. **build**:
   * Called every time the widget needs to be rendered.
   * This method should return a widget tree that represents the current state of the widget.
   * Example from your app:

dart

Copy code

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Currency Converter'),

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text(

\_convertedAmount.toStringAsFixed(2),

style: const TextStyle(fontSize: 40, fontWeight: FontWeight.bold),

),

TextField(

controller: \_controller,

decoration: const InputDecoration(hintText: 'Please enter amount'),

),

// Other UI components

],

),

),

);

}

1. **didUpdateWidget**:
   * Called whenever the widget configuration changes.
   * This method is useful for handling updates when the widget is rebuilt with new properties.
2. **setState**:
   * Called whenever you want to change the state of the widget.
   * Example from your app:

dart

Copy code

void \_convertCurrency() async {

setState(() {

\_isLoading = true;

});

// Perform async operation

setState(() {

\_convertedAmount = amount \* rate;

\_isLoading = false;

});

}

1. **deactivate**:
   * Called when the state object is removed from the widget tree but might be reinserted before the current frame change.
2. **dispose**:
   * Called when the state object is permanently removed from the widget tree.
   * This is where you clean up any resources the state object holds, such as controllers or listeners.
   * Example from your app:

dart

Copy code

@override

void dispose() {

\_controller.dispose();

super.dispose();

}