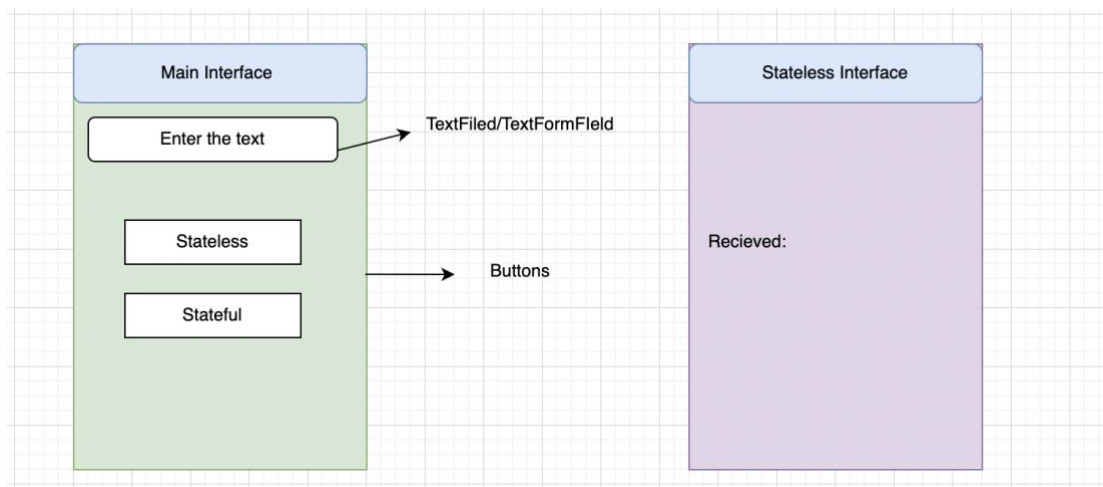


## Passing Data from One Screen to Another:

- There are different ways for passing values from one screen to another. Some of them are as follows.

1. Passing values in Constructor
2. Passing the arguments using Route Settings
3. Passing argument using named routes.



Code for above interface:

main.dart:

```
lib > screen > stateless.dart > ...
1  import 'package:flutter/material.dart';
2
3  class MyStateless extends StatelessWidget {
4    final String? name;
5    const MyStateless({Key? key, @required this.name}) : super(key: key);
6
7    @override
8    Widget build(BuildContext context) {
9      return Scaffold(
10         appBar: AppBar(
11           title: const Text('Stateless Interface'),
12         ), // AppBar
13         body: Center(
14           child: Text(
15             'Recieved $name',
16             textScaleFactor: 2,
17           ), // Text
18         ), // Center
19       ); // Scaffold
20     }
21 }
```

## Interface.dart

```
lib > screen > interface.dart > _InterfaceState > build
1  import 'package:flutter/material.dart';
2  import 'package:pass_val_con/screen/stateless.dart';
3
4  class Interface extends StatefulWidget {
5    const Interface({Key? key}) : super(key: key);
6
7    @override
8    _InterfaceState createState() => _InterfaceState();
9  }
10
11  class _InterfaceState extends State<Interface> {
12    final _formkey = GlobalKey<FormState>();
13
14    String name = '';
15    @override
16    Widget build(BuildContext context) {
17      return Scaffold(
18        appBar: AppBar(
19          title: const Text('Main Interface'),
20        ), // AppBar
21        body: SingleChildScrollView(
22          padding: const EdgeInsets.all(20),
23          child: Form(
24            key: _formkey,
25            child: Column(
26              children: [
27                const SizedBox(height: 50),
28                TextFormField(
29                  onSave: (value) {
30                    name = value!;
31                  },
32                  validator: (value) {
33                    if (value == null || value.isEmpty) {
34                      return '* Required';
35                    }
36                    return null;
37                  },
38                  decoration: const InputDecoration(
39                    labelText: 'Name:',
40                    hintText: 'Enter your name',
41                    border: OutlineInputBorder(),
42                  ), // InputDecoration
43                ), // TextFormField
44                const SizedBox(height: 30),
45                ElevatedButton(
46                  onPressed: () {
47                    if (_formkey.currentState!.validate()) {
48                      _formkey.currentState!.save();
49                      Navigator.push(
50                        context,
51                        MaterialPageRoute(
52                          builder: (context) => MyStateless(name: name),
53                        ), // MaterialPageRoute
54                      );
55                    }
56                  },
57                  child: const Text('Stateless'),
58                ), // ElevatedButton
59              ],
60            ), // Column
61          ), // Form
62        ), // SingleChildScrollView
63      ); // Scaffold
64    }
65  }
```

## stateless.dart

lib > screen > stateless.dart > ...

```
1 import 'package:flutter/material.dart';
2
3 class MyStateless extends StatelessWidget {
4   final String? name;
5   const MyStateless({Key? key, @required this.name}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10      appBar: AppBar(
11        title: const Text('Stateless Interface'),
12      ), // AppBar
13      body: Center(
14        child: Text(
15          'Recieved $name',
16          textScaleFactor: 2,
17        ), // Text
18      ), // Center
19    ); // Scaffold
20  }
21 }
```

For stateful widget we can pass the argument either in constructor or we can use the widget in-order to take the values from widgets to the state.

## Stateful.dart

lib > screen > stateful.dart > \_MyStatefulState > build

```
1 import 'package:flutter/material.dart';
2
3 class MyStateful extends StatefulWidget {
4   final String? name;
5   const MyStateful({Key? key, @required this.name}) : super(key: key);
6
7   @override
8   _MyStatefulState createState() => _MyStatefulState(name!);
9 }
10
11 class _MyStatefulState extends State<MyStateful> {
12   final String? name;
13
14   _MyStatefulState(this.name);
15   @override
16   Widget build(BuildContext context) {
17     debugPrint(widget.name);
18     return Scaffold(
19       appBar: AppBar(
20         title: const Text('Stateful Interface'),
21       ), // AppBar
22       body: Center(
23         child: Column(
24           children: [
25             Text(
26               'Recieved: ${widget.name}',
27               textScaleFactor: 1.5,
28             ), // Text
29             const SizedBox(
30               height: 40,
31             ), // SizedBox
32             Text('const-Recieved $name'),
33           ],
34         ), // Column
35       ), // Center
36     ); // Scaffold
37   }
38 }
```

## 2. Pass the arguments using Route Settings:

```
Navigator.push(context,MaterialPageRoute(builder:( context)=> const MyStateful(),  
Settings:RouteSettings(  
arguments: value or object  
));
```

## 3. Pass arguments to a named route:

```
Navigator.pushNamed(context, '/stateful',arguments: value/object)
```