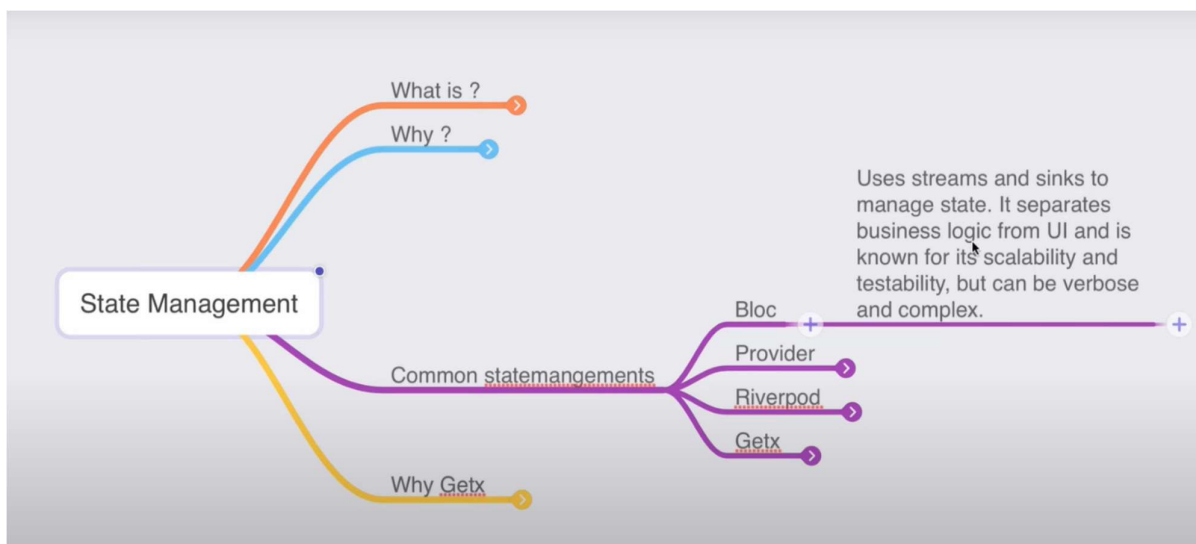
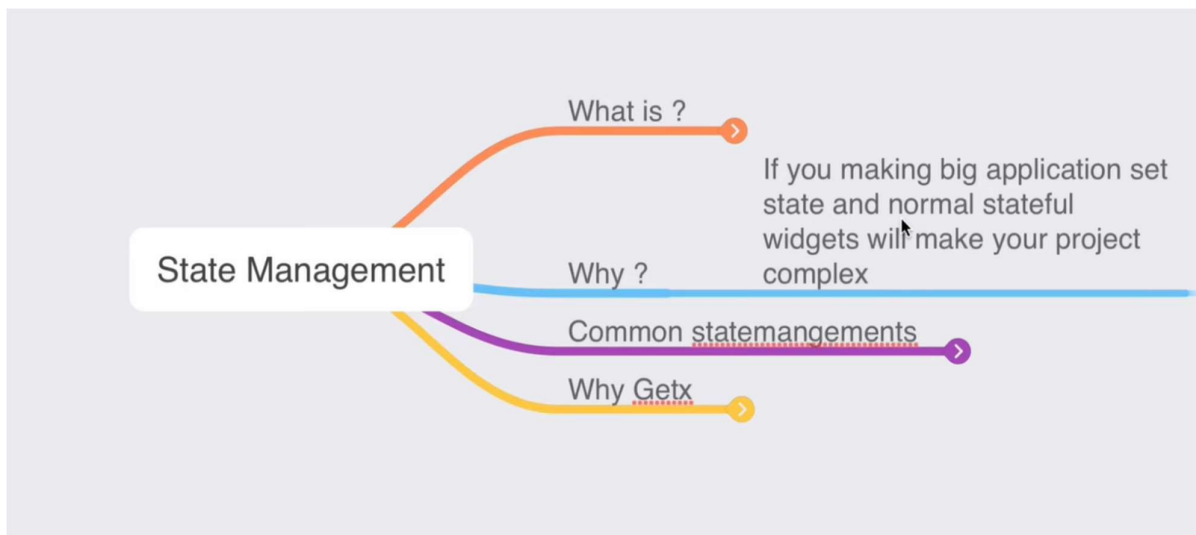
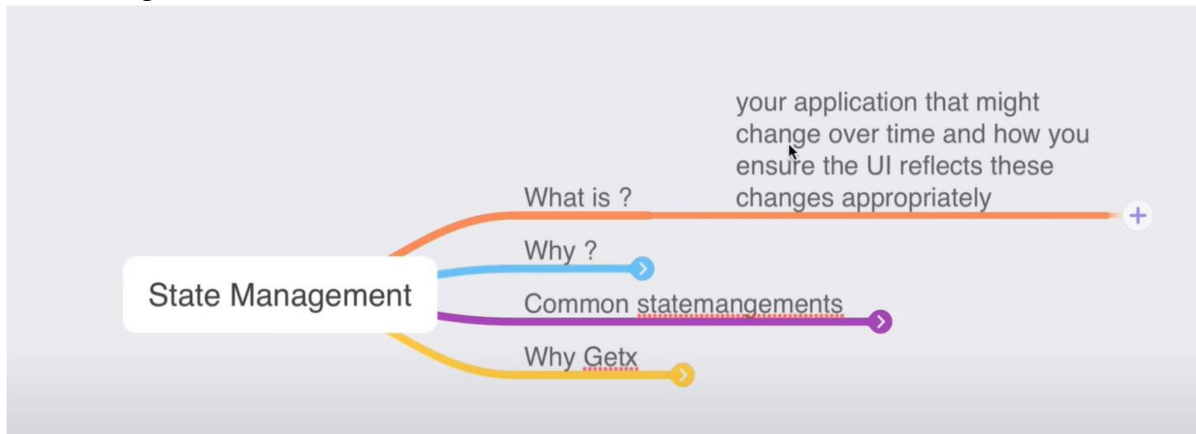
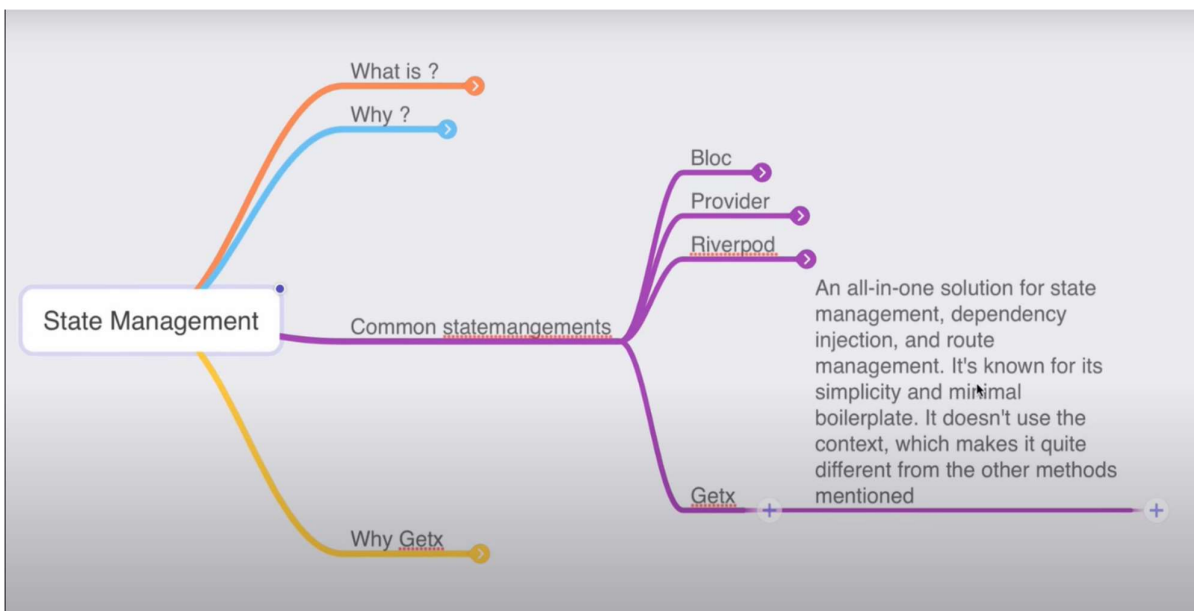
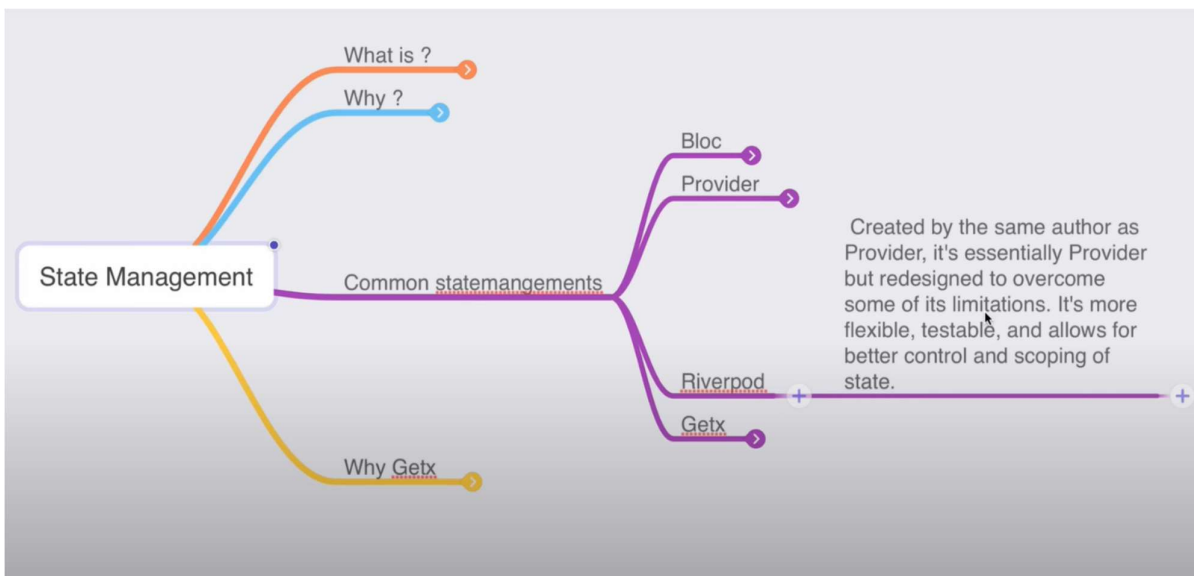
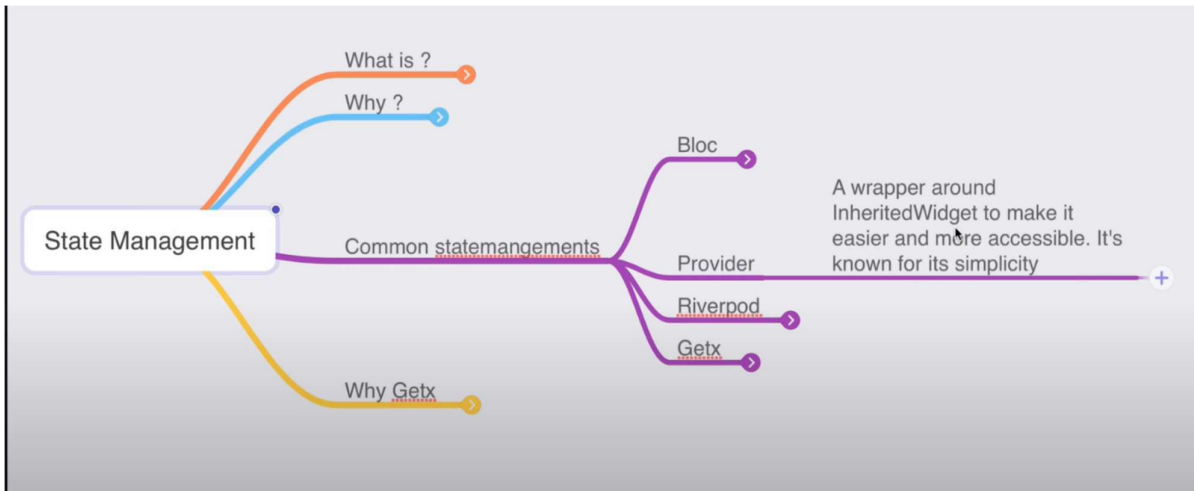
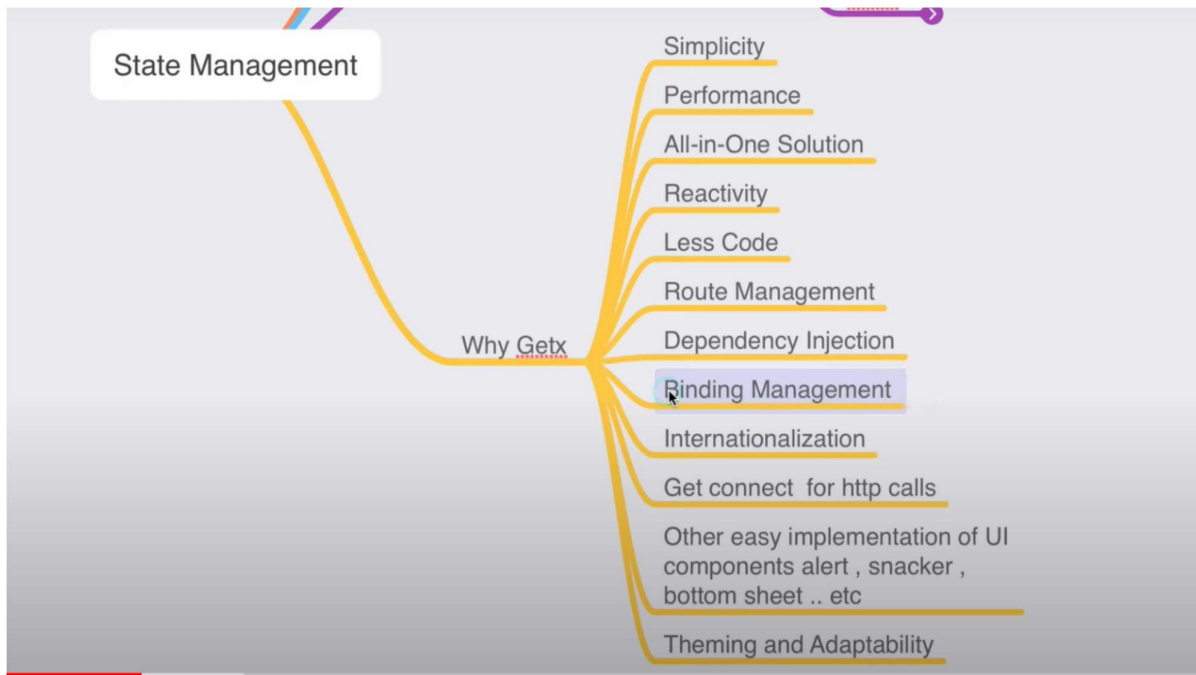


State Management in Flutter: -----

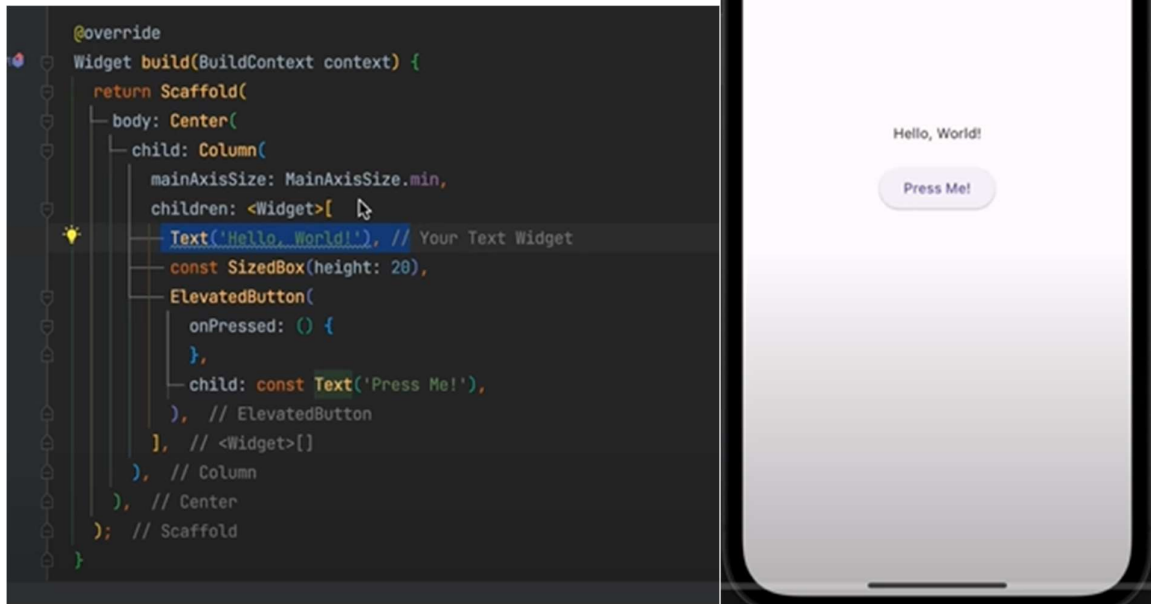






GETX STATE MANAGEMENT

```
main.dart x home_page.dart x pubspec.yaml x
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart'; // Import material.dart for ElevatedButton
3
4 class HomePage extends StatelessWidget {
5   const HomePage({super.key});
6 }
```

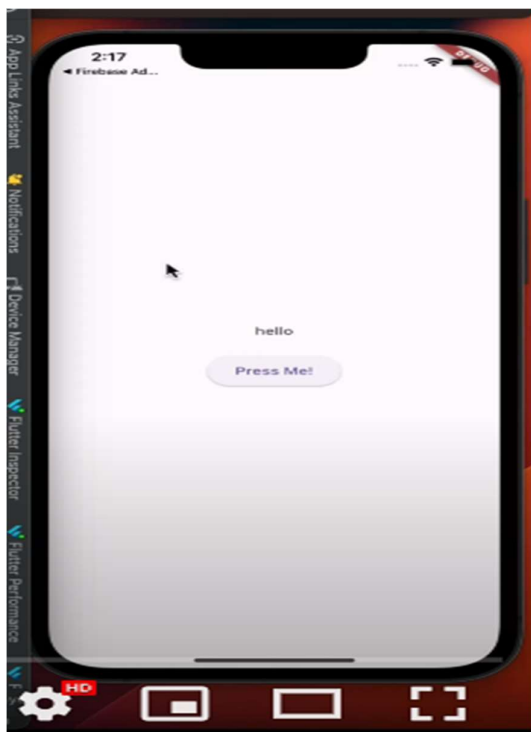


OK ABHI TAK HUMNE KUCH KIYA NAHI HAI BAS EK main.dart hai and home_page.dart jisme humne stateless widget ke help se upar ka ui banaya hai and there is nothing logic in onPressed() method.

2. Now I created a string value jisme hello pass kiya hai and then usko niche text me value variable ko pass kiya hai. And restart the application..

```
main.dart x home_page.dart x pubspec.yaml x
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart'; // Import material.dart for ElevatedButton
3
4 class HomePage extends StatelessWidget {
5   const HomePage({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     String value = 'hello';
10    return Scaffold(
11      body: Center(
12        child: Column(
13          mainAxisAlignment: MainAxisAlignment.min,
14          children: <Widget>[
15            Text(value), // Your Text Widget
16            const SizedBox(height: 20),
17            ElevatedButton(
18              onPressed: () {
19              },
20              child: const Text('Press Me!'),
21            ), // ElevatedButton
22          ], // <Widget>[]
23        ), // Column
24      ), // Center
25    ); // Scaffold
26  }
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Output: -

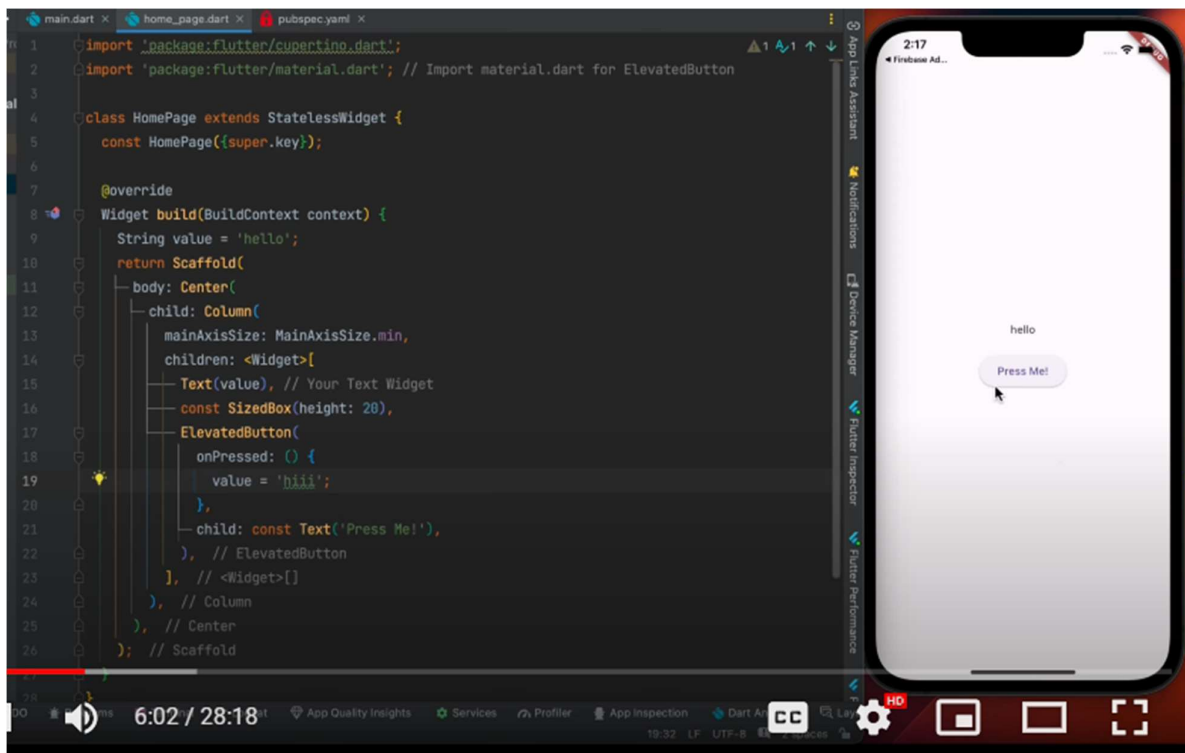


3. Now in onPressed we pass value = 'hiiii'; that means when we click the button so the value will change from hello to hiiii but but ye nahi hua because it's a native behaviour of flutter i.e when we want to change the text then we need to use setState but we are using the stateless widget.....

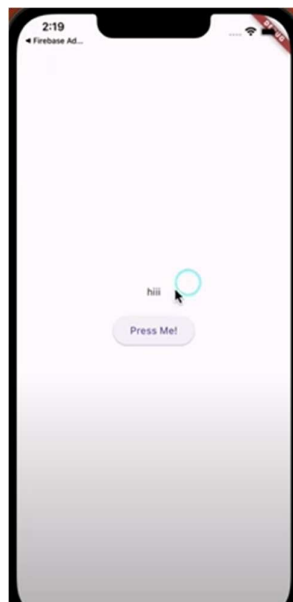
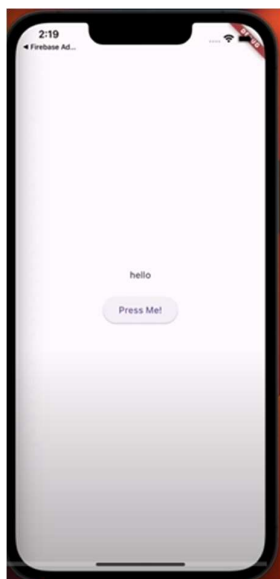
So in conventional method we can change this stateless widget to statefull widget and call the setState jaha par humko ui ko rebuild karna hai i.e in onPressed

Important jo humne string value = 'hello' ko initialize kiya tha at build ke niche now aab hum usko waha se hata ke uder the state likhenge...

Now you can rebuild the application and then you can see that the ui will be changed after onPressed button i.e. from hello to hiiii.



```
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart'; // Import material.dart for ElevatedButton
3
4 class HomePage extends StatelessWidget {
5   const HomePage({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     String value = 'hello';
10    return Scaffold(
11      body: Center(
12        child: Column(
13          mainAxisAlignment: MainAxisAlignment.min,
14          children: <Widget>[
15            Text(value), // Your Text Widget
16            const SizedBox(height: 20),
17            ElevatedButton(
18              onPressed: () {
19                value = 'hiiii';
20              },
21              child: const Text('Press Me!'),
22            ), // ElevatedButton
23          ], // <Widget>[]
24        ), // Column
25      ), // Center
26    ); // Scaffold
27  }
```



Below is the code for the same: -

```
import 'package:flutter/material.dart'; // Import material.dart for ElevatedButton

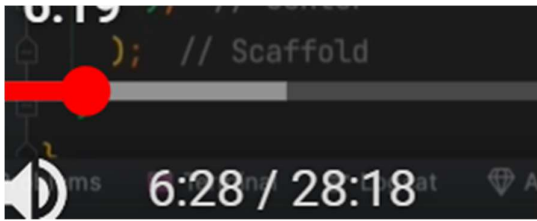
class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    String value = 'hello';
    return Scaffold(
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: <Widget>[
            Text(value), // Your Text Widget
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                value = 'hiiii';
              },
              child: const Text('Press Me!'),
            ), // ElevatedButton
          ], // <Widget>[]
        ), // Column
      ), // Center
    );
  }
}
```

6:26 / 28:18

App Quality Insights Services Profile



Replace code line from 21 to 27 with below code i.e. elevated button part...

```

21 ElevatedButton(
22   onPressed: () {
23     value = 'hihi';
24     setState(() {
25
26     });
27   },

```

So, above was the one native solution but if there will be multiple pages then this solution is possible, but code length will be more thus we have concept of state management in flutter to tackle this problem....

So, now we will study the GetX State Management...

Step 1: -adding get package from pub.dev into pubspec.yaml dependencies and click pub get.

Step 2: -creating controller different file. Example home_controller.dart

Controller is nothing but it is a class.. so create a class and extend GetxController . so now you can use all the in built method of it.

```

main.dart x home_page.dart x home_controller.dart x
1 import 'package:get/get.dart';
2
3 class HomeController extends GetxController{
4
5   @override
6   void onInit() {
7     // TODO: implement onInit
8     super.onInit();
9   }
10
11   @override
12   void update([List<Object>? ids, bool condition = true]) {
13     // TODO: implement update
14     super.update(ids, condition);
15   }
16
17 }

```

```

main.dart x home_page.dart x home_controller.dart x
1 import 'package:get/get.dart';
2
3 class HomeController extends GetxController{
4
5   @override
6   void onInit() {
7     // TODO: implement onInit
8     super.onInit();
9   }
10
11   test(){
12     update();
13   }
14
15
16 }

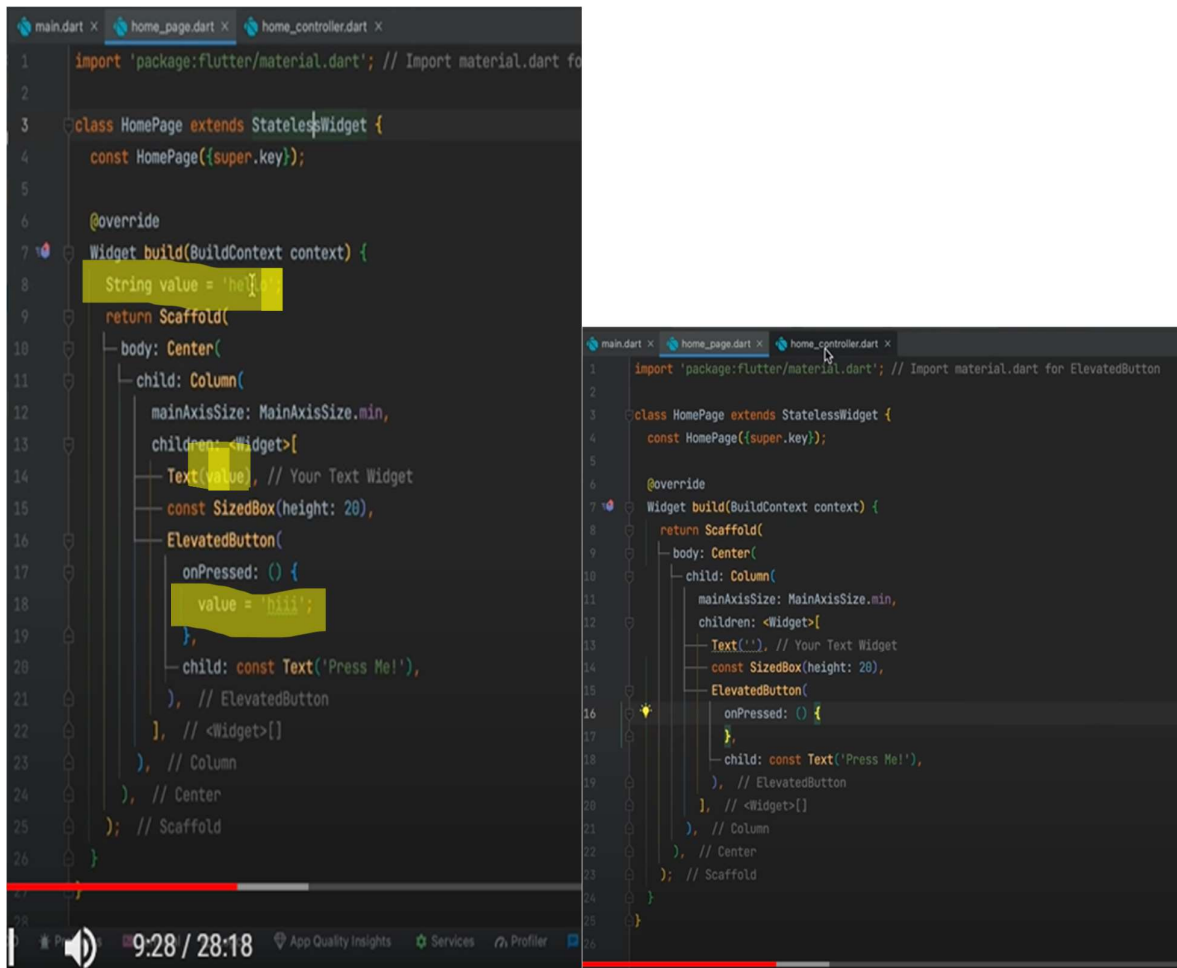
```

So here update and onInit are method under the GetX package.

Therefore when we use the getx we can separate our logic code and ui code.i.e. perform this step

Therefore home_page.dart looks like this as below after removing the logic code.

Before



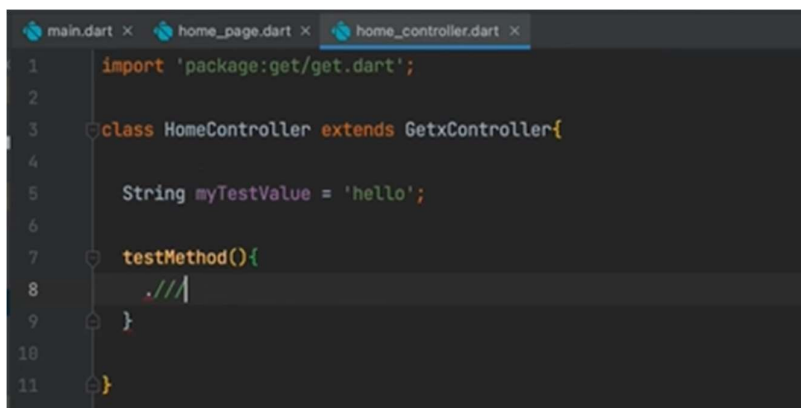
i.e. yellow part is removed... which is logic code and it is added into a home_controller.dart

and then connect the controller with the page.

Steps to connect is that :-

Scaffold ko wrap with Builder and write Get ahead of it. Then add <> after just GetBuilder and inside it write controller name i.e. <HomeController> and instead of context write as controller and then to access the methods and variables of HomeController inside the home_page

We write **controller**, then all options dikhega. Example **controller.myTestValue**



```
main.dart x home_page.dart x home_controller.dart x
3 import 'package:youtube_tutorial/home_controller.dart'; // Import material.dart for El
4
5 class HomePage extends StatelessWidget {
6   const HomePage({super.key});
7
8   @override
9   Widget build(BuildContext context) {
10    return GetBuilder<HomeController>(builder: (controller) {
11      return Scaffold(
12        body: Center(
13          child: Column(
14            mainAxisAlignment: MainAxisAlignment.min,
15            children: <Widget>[
16              Text(' '), // Your Text Widget
17              const SizedBox(height: 20),
18              ElevatedButton(
19                onPressed: () {
20                  controller.testMethod();
21                  controller.myTestValue;
22                },
23                child: const Text('Press Me!'),
24              ), // ElevatedButton
25            ], // <Widget>[]
26          ), // Column
27        ), // Center
28      ); // Scaffold
29    }); // GetBuilder
30  }
```

Now abhi humne GetBuilder ka use karke kiya tha now we will implement one more shortcut method i.e.

Step 1: - In android studio go into the setting search for plugins and install the GetX and reinstall your ide.

```
main.dart x home_page.dart x home_controller.dart x
5 class HomePage extends StatelessWidget {
6   const HomePage({super.key});
7
8   @override
9   Widget build(BuildContext context) {
10    return Scaffold(
11      body: Center(
12        child: Column(
13          mainAxisAlignment: MainAxisAlignment.min,
14          children: <Widget>[
15            Text(' '), // Your Text Widget
16            const SizedBox(height: 20),
17            ElevatedButton(
18              onPressed: () {
19                controller.testMethod();
20                controller.myTestValue;
21              },
22              child: const Text('Press Me!'),
23            ), // ElevatedButton
24          ], // <Widget>[]
25        ), // Column
26      ), // Center
27    ); // Scaffold
28  }
```

Then jo normal code hai i.e below and wrap scaffold with GetBuilder aab direct option dikehga and then usme change <SubjectLogic> with home_controller name i.e. <HomeController> and also replace (logic) with (controller). and then connect page with controller..

Also we have to register the controller before it is used... if you are using a single page or less pages then you can register it in main.dart

Before registering: -

```

1 import 'package:flutter/material.dart';
2 import 'package:youtube_tutorial/home_page.dart';
3
4
5
6 void main() => runApp(const MyApp());
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return const MaterialApp(
14       home: HomePage(),
15     ); // MaterialApp
16   }
17 }
18
19

```

After registering: -

```

3 import 'package:youtube_tutorial/home_controller.dart';
4 import 'package:youtube_tutorial/home_page.dart';
5
6
7
8 void main() {
9   Get.put(HomeController());
10  runApp(const MyApp());
11 }
12
13 class MyApp extends StatelessWidget {
14   const MyApp({super.key});
15
16   @override
17   Widget build(BuildContext context) {
18     return const MaterialApp(
19       home: HomePage(),
20     ); // MaterialApp
21   }
22 }
23

```

Adding logic to home_page.dart inside the onPressed... note inside the home_controller.dart there is no such update() but we used it in home_page.dart this is inbuilt of GetX package.

```

8 @override
9 Widget build(BuildContext context) {
10   return GetBuilder<HomeController>(builder: (controller) {
11     return Scaffold(
12       body: Center(
13         child: Column(
14           mainAxisAlignment: MainAxisAlignment.min,
15           children: <Widget>[
16             Text(controller.myTestValue), // Your Text Widget
17             const SizedBox(height: 20),
18             ElevatedButton(
19               onPressed: () {
20                 controller.myTestValue = 'hi! hi!';
21                 controller.update();
22               },
23               child: const Text('Press Me!'),
24             ), // ElevatedButton
25           ], // <Widget>[]
26         ), // Column
27       ), // Center
28     );
29   });
30 }
31

```

So itne se jo hamara desire tha karna v oho gaya i.e output: -



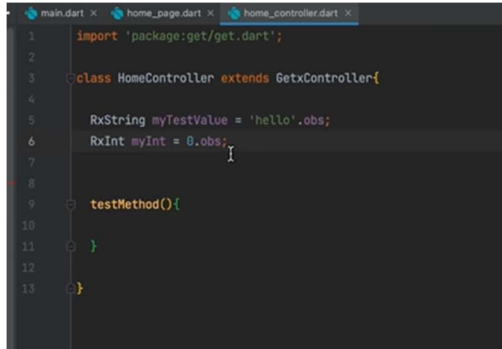
after Press Me! Ui is changed



So, this is the basics of the GetX state management ..

Now instead of using the update() we also have one more option i.e. using concept of reactive element.

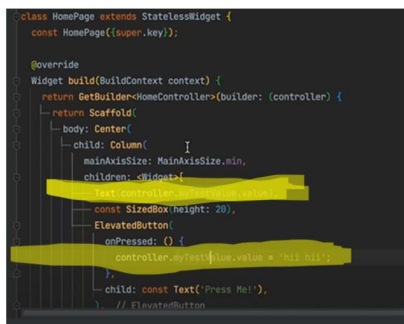
i.e. make variables as Rx in home_controller.dart



```
1 import 'package:get/get.dart';
2
3 class HomeController extends GetxController{
4
5   RxString myTestValue = 'hello'.obs;
6   RxInt myInt = 0.obs;
7
8
9   testMethod(){
10
11 }
12
13 }
```

And then make changes in home_page.dart

update() hata do. And do niche wale highlight part

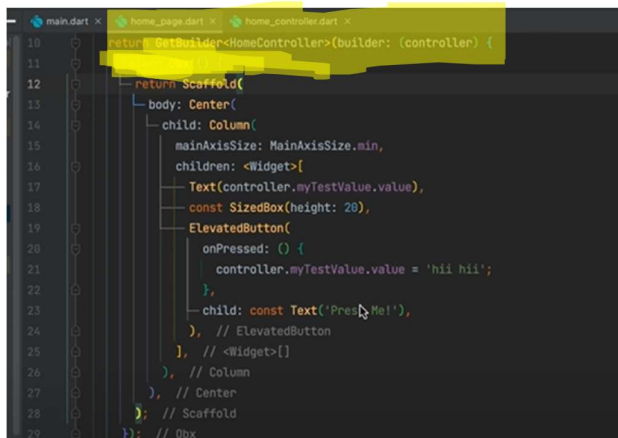


```
class HomePage extends StatelessWidget {
  const HomePage({super.key});

  @override
  Widget build(BuildContext context) {
    return GetBuilder<HomeController>(builder: (controller) {
      return Scaffold(
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: <Widget>[
              Text(controller.myTestValue.value),
              const SizedBox(height: 20),
              ElevatedButton(
                onPressed: () {
                  controller.update(() {
                    controller.myTestValue.value = 'hi hi!';
                  });
                },
                child: const Text('Press Me!'),
              ), // ElevatedButton
            ], // <Widget>[]
          ), // Column
        ), // Center
      ); // Scaffold
    }); // Obx
  }
}
```

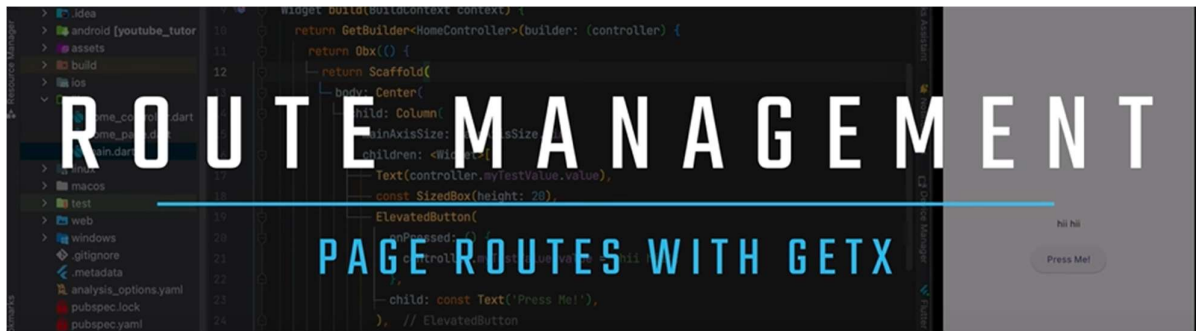
But itna karne pe run nahi hoga i.e. ui will not changed therefore wrap(alt+enter) {multiple options opened} the ui or scaffold with **obs** done. Agar ye option obs wala nahi dikhega then may be obs ka package nahi import kiya hai therefore usko add kar lena...

i.e refer below image: -



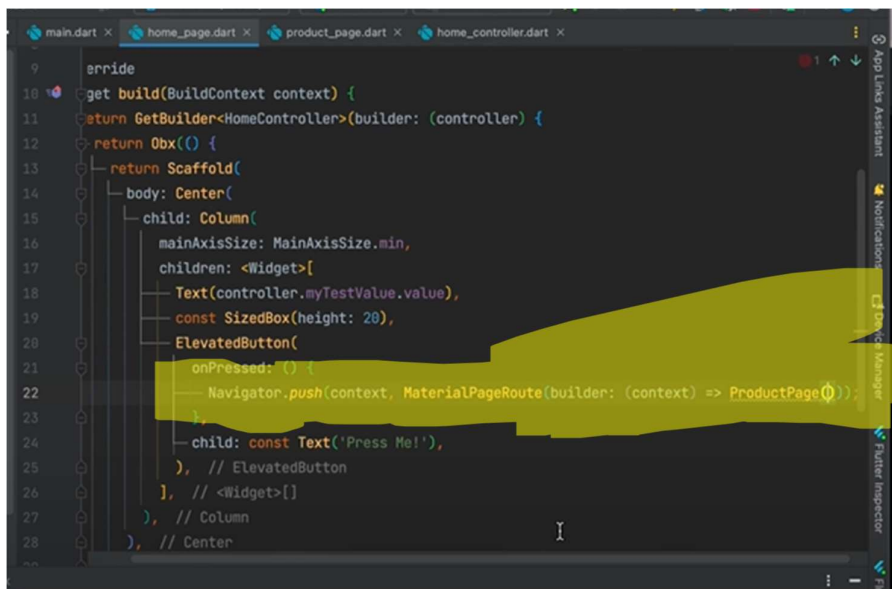
```
10 return GetBuilder<HomeController>(builder: (controller) {
11   return Scaffold(
12     body: Center(
13       child: Column(
14         mainAxisAlignment: MainAxisAlignment.min,
15         children: <Widget>[
16           Text(controller.myTestValue.value),
17           const SizedBox(height: 20),
18           ElevatedButton(
19             onPressed: () {
20               controller.myTestValue.value = 'hi hi!';
21             },
22             child: const Text('Press Me!'),
23           ), // ElevatedButton
24         ], // <Widget>[]
25       ), // Column
26     ), // Center
27   ); // Scaffold
28 }); // Obx
```

Now we will study the i.e



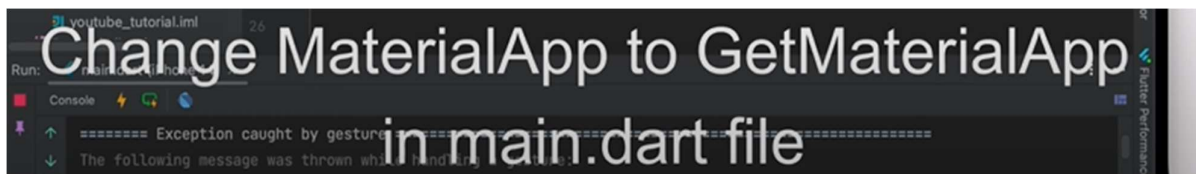
Added the new product.dart page (so now button ko press karne ke baad hum home_page se product_page pe jayenge).

Normal method is below i.e write one that highlighted code.



2nd method i.e. by using getX :-

Before that using these **getx services** like alert, snakebar, route, and etc... we need to do one change i.e. vvvvimp



Before :-

```

1 import 'package:youtube_tutorial/home_page.dart';
2
3
4
5
6
7
8
9 void main() {
10   Get.put(HomeController());
11   runApp(const MyApp());
12 }
13
14 class MyApp extends StatelessWidget {
15   const MyApp({super.key});
16
17   @override
18   Widget build(BuildContext context) {
19     return const MaterialApp(
20       home: HomePage(),
21     ); // MaterialApp
22 }

```

After: -

```

7
8 void main() {
9   Get.put(HomeController());
10  runApp(const MyApp());
11 }
12
13 class MyApp extends StatelessWidget {
14   const MyApp({super.key});
15
16   @override
17   Widget build(BuildContext context) {
18     return const GetMaterialApp(
19       home: HomePage(),
20     ); // GetMaterialApp
21 }
22
23

```

i.e. home_page.dart code changes: - Get.to(ProductPage()); line no. 22...

```

9
10 @override
11 Widget build(BuildContext context) {
12   return GetBuilder<HomeController>(builder: (controller) {
13     return Obx(() {
14       return Scaffold(
15         body: Center(
16           child: Column(
17             mainAxisAlignment: MainAxisAlignment.min,
18             children: <Widget>[
19               Text(controller.myTestValue.value),
20               const SizedBox(height: 20),
21               ElevatedButton(
22                 onPressed: () {
23                   Get.to(ProductPage());
24                   // Navigator.push(context, MaterialPageRoute(builder: (context) => ProductPage()))
25                 },
26                 child: const Text('Press Me!'),
27               ), // ElevatedButton
28             ], // <Widget>[]
29           ), // Column
30         ), // Scaffold
31       );
32     });
33   });
34 }

```

Therefore output desired is achieved..



i.e. after clicking on Press Me! We route to a new page..

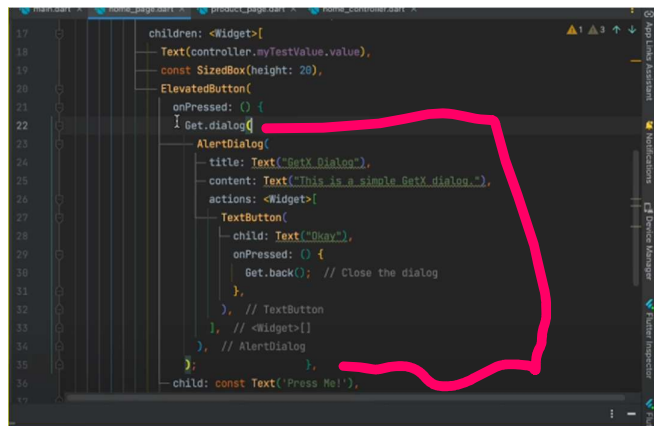


Also there are different types of getx routing i.e. can be seen by get.

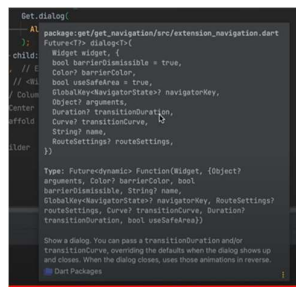
Now we will study: -



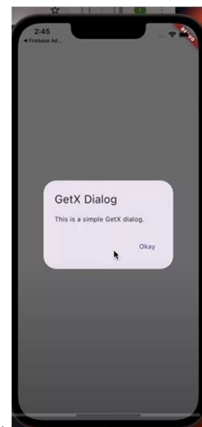
Code changes:-



getx dialog has lot of parameters



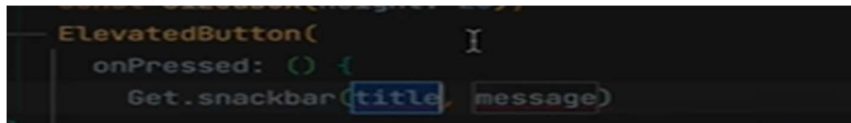
i.e. when we click Press Me! We get an alert.



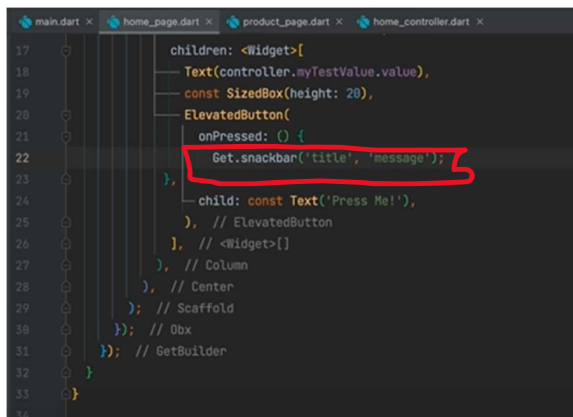
Now we will study about the



Code changes :- i.e.



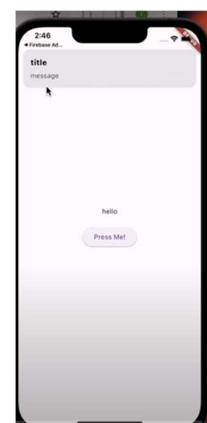
Now pass the title and message accordingly..



Output: -



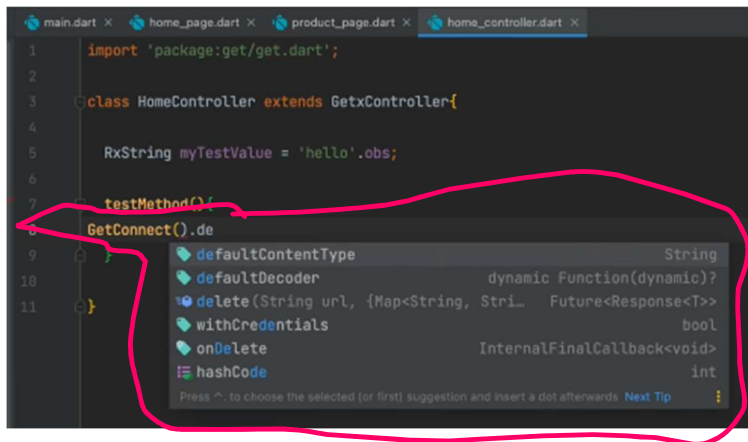
when we click Press Me! Then snack bar is shown at top..



Now we will study



Get method, post method, put method, delete method, all method is available.

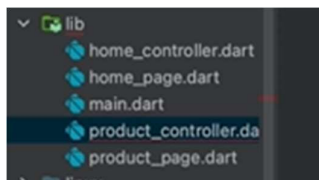


In detail need to study...

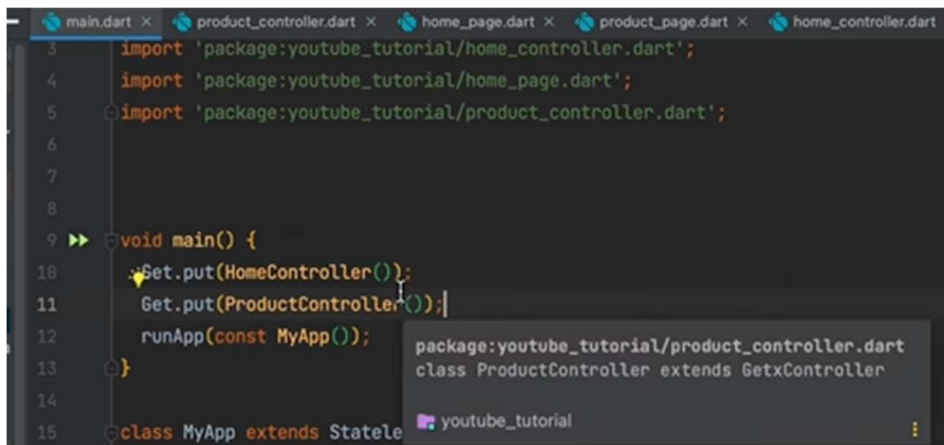
Now we will study : -



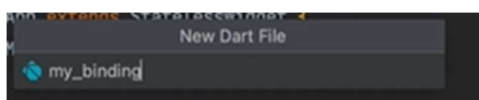
i..e when we have multiple controller



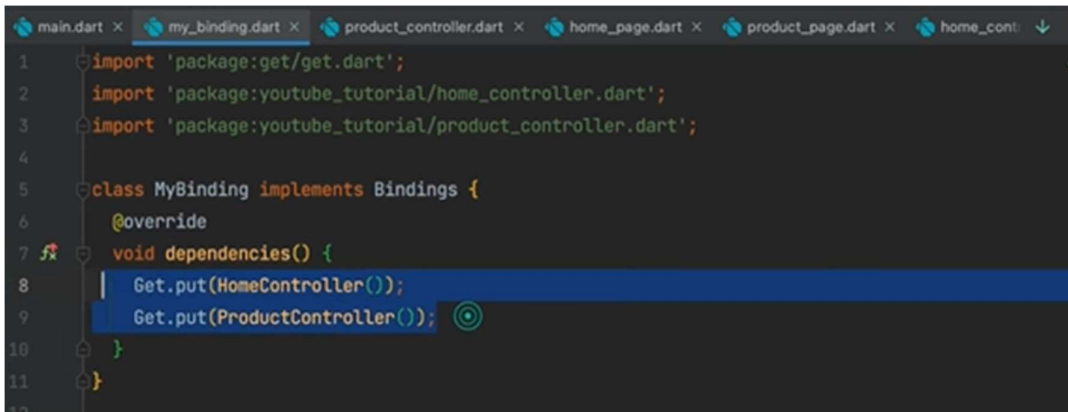
If you are making small application so you can register controller over there no problem like this



large application we have problem... therefore we will make one another file called binding



and write code into binding



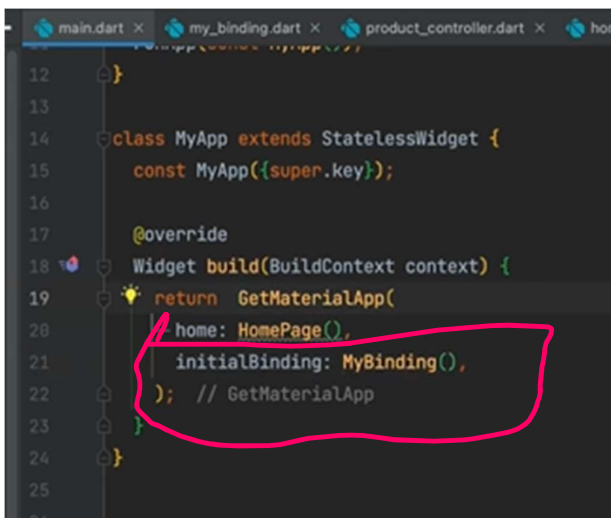
```

1 import 'package:get/get.dart';
2 import 'package:youtube_tutorial/home_controller.dart';
3 import 'package:youtube_tutorial/product_controller.dart';
4
5 class MyBinding implements Bindings {
6   @override
7   void dependencies() {
8     Get.put(HomeController());
9     Get.put(ProductController());
10  }
11 }

```

and

remove from main.dart and then make like below changes i.e. main.dart me at GetMaterialApp
Me.



```

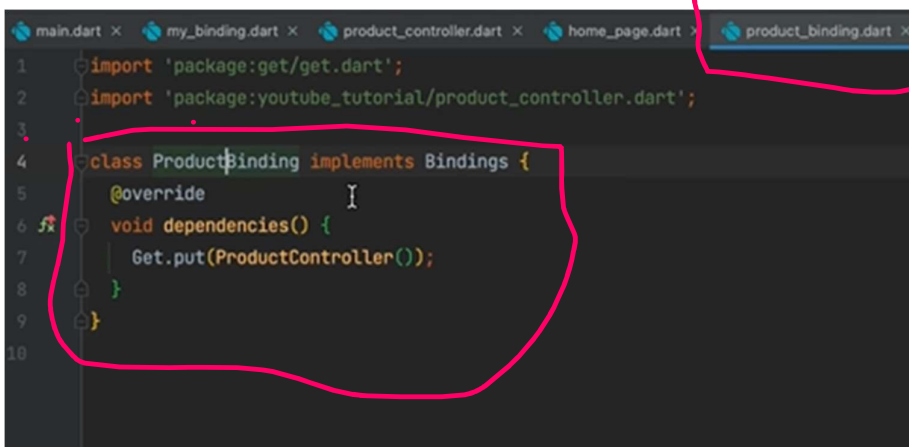
12 }
13
14 class MyApp extends StatelessWidget {
15   const MyApp({super.key});
16
17   @override
18   Widget build(BuildContext context) {
19     return GetMaterialApp(
20       home: HomePage(),
21       initialBinding: MyBinding(),
22     ); // GetMaterialApp
23   }
24 }
25
26

```

so this is professional way to use the controller in production level applications. Don't use directly in the run app or void main method.

Now one problem with binding is that humne sab controller my_binding me daal diya but what if sab controller nahi use ho ga then it will consume unnecessary memory which will result into slow the application.

Hence to tackle this problem we create a separate binding for each controller and then jaha par use karna hai to call kar lete hai uska fix syntax hai...



```

1 import 'package:get/get.dart';
2 import 'package:youtube_tutorial/product_controller.dart';
3
4 class ProductBinding implements Bindings {
5   @override
6   void dependencies() {
7     Get.put(ProductController());
8   }
9 }

```

```
18 children: <Widget>[
19   Text(controller.myTestValue.value),
20   const SizedBox(height: 20),
21   ElevatedButton(
22     onPressed: () {
23       Get.to(ProductPage(), binding: ProductBinding());
24     },
25     child: const Text('Press Me!'),
26   ), // ElevatedButton
27 ], // <Widget>[]
28 ], // Column
29 ], // Center
30 ); // Scaffold
31 ); // Obx
32 ); // GetBuilder
33 }
34 }
```

```
Run: main.dart (iPhone 14) x
Console
Performing hot restart...
Syncing files to device iPhone 14...
Restarted application in 1,191ms.
[GETX] Instance "HomeController" has been created
[GETX] Instance "HomeController" has been initialized
[GETX] Instance "GetMaterialController" has been created
[GETX] Instance "GetMaterialController" has been initialized
```

now you

can see only home controller is created and initialized.

Jab tum product page pe jaooge then only product controller is created and initialized....

```
[GETX] GOING TO ROUTE ProductPage
[GETX] Instance "ProductController" has been created
[GETX] Instance "ProductController" has been initialized
```