

How are we doing? Please help us improve Stack Overflow. [Take our short survey](#)



Flutter: Changing the current tab in tab bar view using a button

Asked 3 years, 7 months ago · Active 8 months ago · Viewed 78k times



57

I am creating an app that contains a tab bar on its homepage. I want to be able to **navigate** to one of the tabs **using** my `FloatingActionButton`. In addition, I want to keep the default methods of navigating to that tab, i.e. by swiping on screen or by clicking the tab.

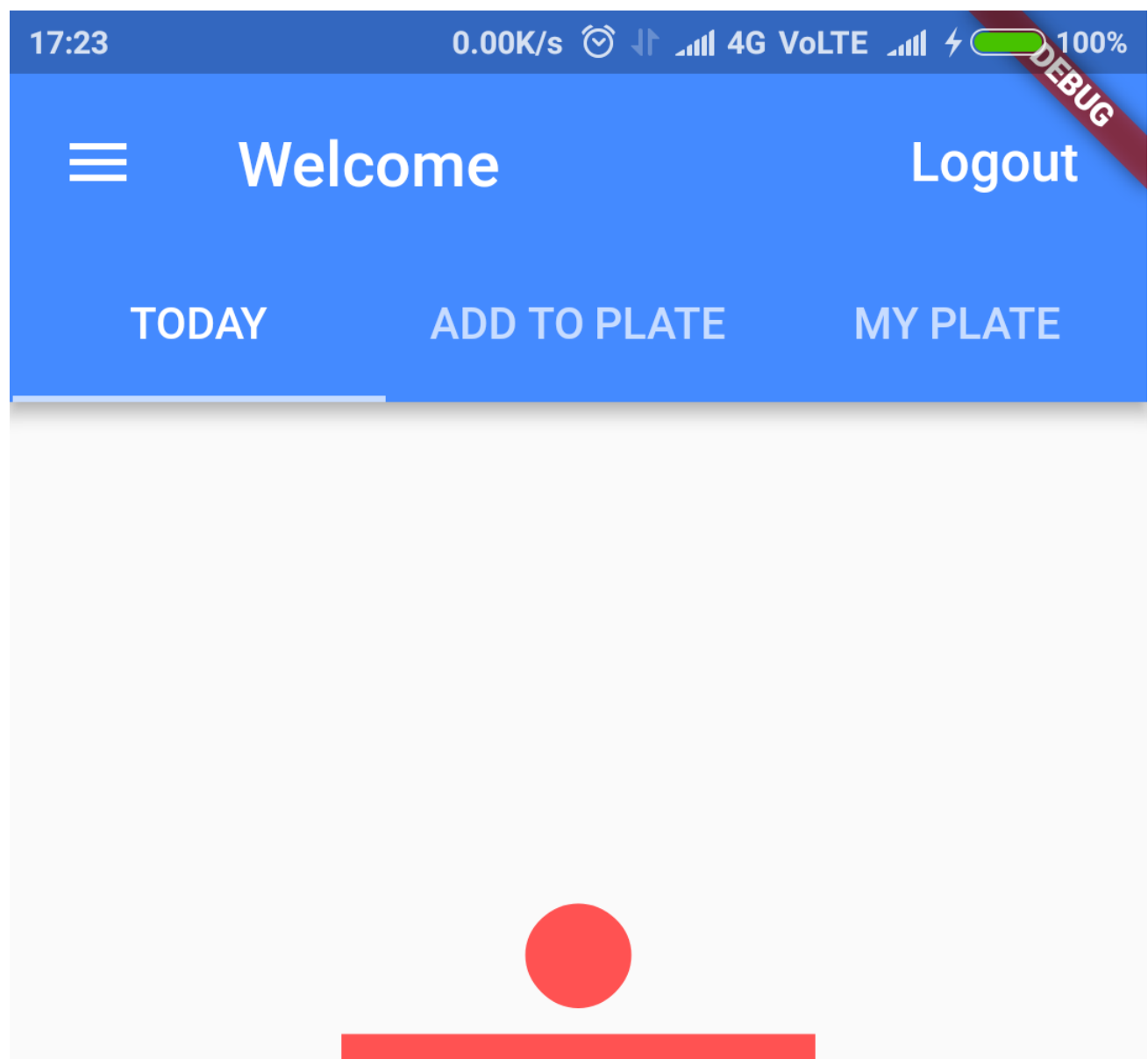


I also want to know how to link that tab to some other button.



26

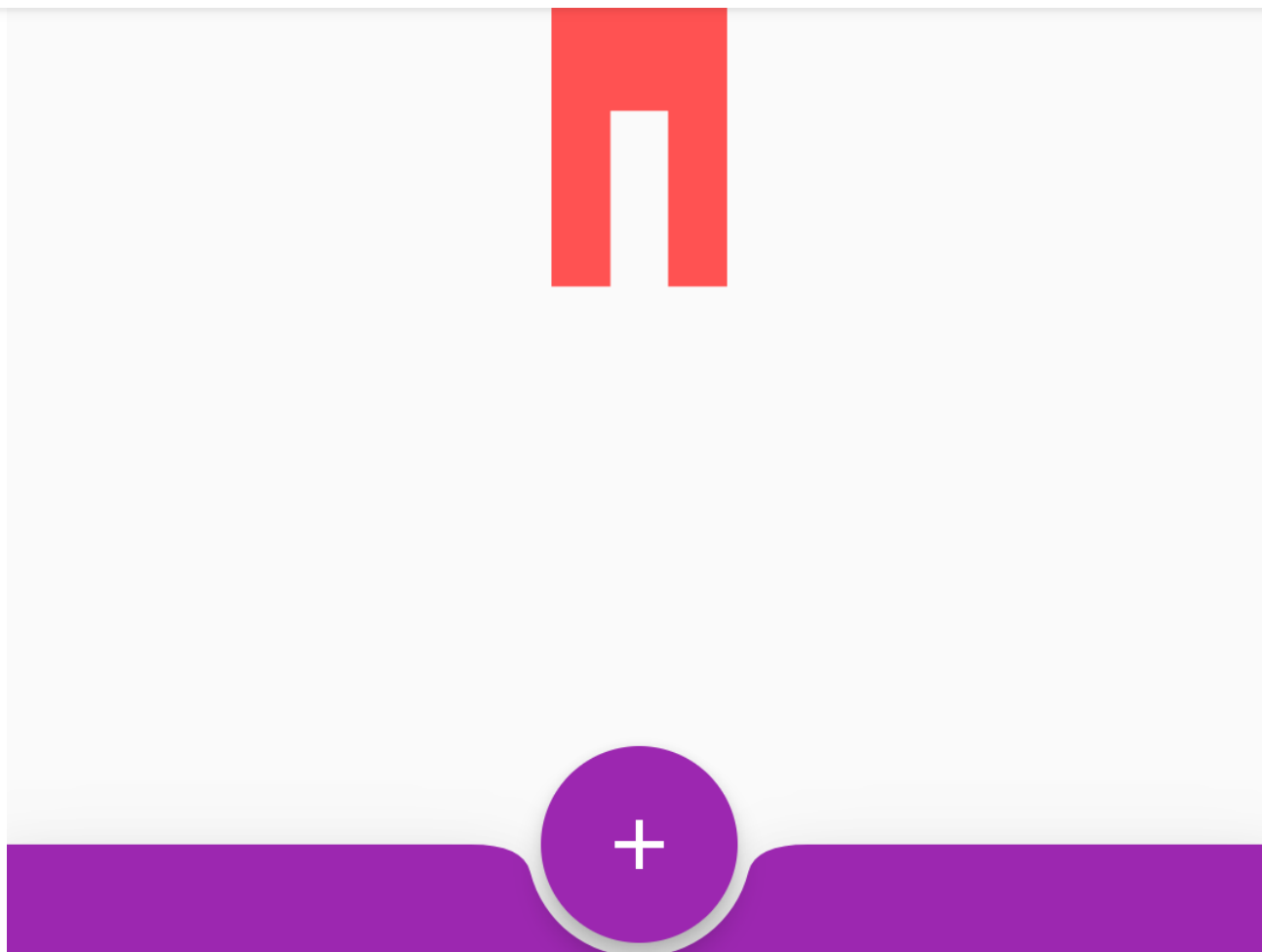
Here is a screenshot of my homepage.



Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up](#)





dart flutter

Share Follow

edited Jun 16 '18 at 15:55



[creativecreatorormaybenot](#)
80.9k ● 44 ● 213 ● 327

asked Jun 16 '18 at 12:12



[Sunit Gautam](#)
3,533 ● 2 ● 14 ● 26

How to achieve that bottom widget effect? Is it an image? – [Muhammad Qasim](#) Sep 29 '21 at 11:23

@Muhammad Qasim this is a standard Flutter widget, however, I am unable to find the code. If anyone recognises this bottom widget's name, please let us know. – [Sunit Gautam](#) Oct 24 '21 at 19:56

you mean floating action button? – [Muhammad Qasim](#) Oct 25 '21 at 6:57

7 Answers

Active Oldest Votes

You need to get the `TabBar` controller and call its `animateTo()` method from the button

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up





```
void main() => runApp(new MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: 'Flutter Demo',
      home: new MyTabbedPage(),
    );
  }
}

class MyTabbedPage extends StatefulWidget {
  const MyTabbedPage({Key key}) : super(key: key);

  @override
  _MyTabbedPageState createState() => new _MyTabbedPageState();
}

class _MyTabbedPageState extends State<MyTabbedPage> with
SingleTickerProviderStateMixin {
  final List<Tab> myTabs = <Tab>[
    new Tab(text: 'LEFT'),
    new Tab(text: 'RIGHT'),
  ];

  TabController _tabController;

  @override
  void initState() {
    super.initState();
    _tabController = new TabController(vsync: this, length: myTabs.length);
  }

  @override
  void dispose() {
    _tabController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text("Tab demo"),
        bottom: new TabBar(
          controller: _tabController,
          tabs: myTabs,
        ),
      ),
      body: new TabBarView(
        controller: _tabController,
        children: myTabs.map((Tab tab) {
          return new Center(child: new Text(tab.text));
        }).toList(),
      ),
      floatingActionButton: new FloatingActionButton(
```

```
);  
  
}  
}
```

If you use a `GlobalKey` for the `MyTabbedPageState` you can get the controller from any place, so you can call the `animateTo()` from any button.

```
class MyApp extends StatelessWidget {  
  static final _myTabbedPageKey = new GlobalKey<MyTabbedPageState>();  
  
  @override  
  Widget build(BuildContext context) {  
    return new MaterialApp(  
      title: 'Flutter Demo',  
      home: new MyTabbedPage(  
        key: _myTabbedPageKey,  
      ),  
    );  
  }  
}
```

You could call it from anywhere doing:

```
MyApp._myTabbedPageKey.currentState._tabController.animateTo(...);
```

Share Follow

edited Jun 16 '18 at 18:00

answered Jun 16 '18 at 17:24



chemamolins

15.7k ● 3 ● 47 ● 43

Is there callback to update tabView page content from other stateful widget call back trigger. – [TejaDroid](#) Mar 6 '19 at 11:18

3 But how to pass data between them? – [Aman gautam](#) Sep 23 '19 at 16:00

Yes, how are you able to pass parameters to the destination Widget? – [FloatingRock](#) Mar 1 '20 at 14:17

`_tabController.animateTo(_tabController.index + 1)` helped me to navigate to next tab. Thanks a lot Chemamolins. – [Kamlesh](#) May 21 '21 at 3:50



37



```
DefaultTabController.of(context).animateTo(1);
```



Share Follow

answered Dec 12 '20 at 11:07

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up



- 4 I always love one-line solution :) Thanks a lot man! – [Mano Haran](#) Jan 6 '21 at 23:57
- 1 exactly what i wanted – [Umesh Chakradhar](#) Feb 12 '21 at 8:50
- 4 not worked for me it's said that called on null The method 'animateTo' was called on null. Receiver: null – [Omar Essam](#) Mar 18 '21 at 14:53
- 4 @OmarEssam Please use it inside `DefaultTabController(length: 3,child: Builder(builder: (BuildContext context) { // here before return }` – [NaKib](#) Apr 24 '21 at 8:37

You can use `TabController`:

```
TabController _controller = TabController(
  vsync: this,
  length: 3,
  initialIndex: 0,
);

_controller.animateTo(_currentTabIndex);

return Scaffold(
  appBar: AppBar(
    bottom: TabBar(
      controller: _controller,
      tabs: [
        ...
      ],
    ),
  ),
  body: TabBarView(
    controller: _controller,
    children: [
      ...
    ],
  ),
);
```

And than, `setState` to update screen:

```
int _currentTabIndex = 0;

setState(() {
  _currentTabIndex = 1;
});
```

Share Follow

answered Jun 1 '20 at 21:36



[Waister Nunes](#)

71 ● 1 ● 3



4

[chemamolin's answer above](#) is correct, but for additional clarification/tip, if you want to call your tabcontroller "from anywhere", also make sure the tabcontroller is not a private property of the class by removing the underscore, otherwise the distant class will not be able to see the tabcontroller with the example provided even when using the GlobalKey.



In other words, change

```
TabController _tabController;
```

to:

```
TabController tabController;
```

and change

```
MyApp._myTabbedPageKey.currentState._tabController.animateTo(...);
```

to:

```
MyApp._myTabbedPageKey.currentState.tabController.animateTo(...);
```

and everywhere else you reference tabcontroller.

Share Follow

answered Sep 29 '19 at 6:01



FPI.MD

51 ● 3



2

If you want to jump to a specific page, you can use

```
PageController.jumpToPage(int)
```



However if you need animation, you'd use

```
PageController.animateToPage(page, duration: duration, curve: curve)
```

Simple example demonstrating it.

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up



```

bool _shouldAnimate = true; // whether we animate or jump

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(),
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        if (_shouldAnimate) {
          // animates to page1 with animation
          _controller.animateToPage(1, duration: Duration(seconds: 1), curve:
Curves.easeOut);
        } else {
          // jump to page1 without animation
          _controller.jumpToPage(1);
        }
      },
    ),
    body: PageView(
      controller: _controller, // assign it to PageView
      children: <Widget>[
        FlutterLogo(colors: Colors.orange), // page0
        FlutterLogo(colors: Colors.green), // page1
        FlutterLogo(colors: Colors.red), // page2
      ],
    ),
  );
}

```

Share Follow

answered Sep 9 '19 at 17:15

**CopsOnRoad**

151k ● 43 ● 479 ● 342

Is there a way we're able to pass parameters to the destination page the way we can with e.g.

`Navigator.push(context, MaterialPageRoute(builder: (_) => LogoScreen(color: "red"))` ?

– [FloatingRock](#) Mar 1 '20 at 14:21

There is `onPageChanged` callback which would you give you the current page, and hence you can retrieve the parameters accordingly, if I didn't understand your question properly, you can simply ask a new question. – [CopsOnRoad](#) Mar 1 '20 at 15:38

```

DefaultTabController(
  length: 4,
  initialIndex: 0,
  child: TabBar(
    tabs: [
      Tab(
        child: Text(
          "People",
          style: TextStyle(
            color: Colors.black,

```

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up](#)


```

        child: Text(
          "Events",
          style: TextStyle(
            color: Colors.black,
          ),
        ),
      ),
      Tab(
        child: Text(
          "Places",
          style: TextStyle(
            color: Colors.black,
          ),
        ),
      ),
      Tab(
        child: Text(
          "HashTags",
          style: TextStyle(
            color: Colors.black,
          ),
        ),
      ),
    ],
  ),
)

```

Share Follow

edited Apr 28 '21 at 13:38



Community Bot

1 • 1

answered Mar 10 '21 at 8:07



Hanshir

39 • 3

Hi and welcome to Stack Overflow! Please take the [tour](#). Thanks for answering but can you also add an explanation on how your code solves the issue? Check the [help center](#) for info on how to format code.

– Tyler2P Mar 10 '21 at 9:05

class Tab bar

```

class TabBarScreen extends StatefulWidget {
  TabBarScreen({Key key}) : super(key: key);

  @override
  _TabBarScreenState createState() => _TabBarScreenState();
}

```

```

final List<Tab> tabs = <Tab>[
  Tab(text: 'Page1'),
  Tab(text: 'Page2'),
];

```

```

class _TabBarScreenState extends State<TabBarScreen> with
SingleTickerProviderStateMixin {

```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up




```

    super.initState();
    tabController = new TabController(vsync: this, length: tabs.length);
  }

  @override
  void dispose() {
    tabController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        backgroundColor: Theme.of(context).primaryColor,
        appBar: AppBar(
          backgroundColor: Theme.of(context).primaryColor,
          centerTitle: true,
          shape: Border(bottom: BorderSide(color: Colors.white)),
          title: Text("Tab Bar"),
          bottom: TabBar(
            controller: tabController,
            tabs: tabs,
            indicatorWeight: 5,
            indicatorColor: Colors.white,
            labelColor: Colors.white,
          ),
        ),
        body: TabBarView(
          controller: tabController,
          children: [
            PageOneScreen(controller: tabController),
            PageTwoScreen(controller: tabController),
          ],
        ),
      ),
    );
  }
}

class PageOne

class PageOneScreen extends StatefulWidget {
  @override
  _PageOneScreenState createState() => _PageOneScreenState();

  PageOneScreen({controller}) {
    tabController = controller;
  }
}

TabController tabController;

class _PageOneScreenState extends State<PageOneScreen> {
  @override
  Widget build(BuildContext context) {
    return Column(

```

```
    },  
    child: Text(  
      "Go To Page 2",  
    ),  
  ],  
);  
}
```

[Share](#) [Follow](#)

edited Dec 7 '20 at 9:02



shreyasm-dev

2,336 ● 5 ● 13 ● 31

answered Dec 5 '20 at 21:04



ALNAJJAR

51 ● 5

While this code may answer the question, providing additional context regarding how and/or why it solves the problem would improve the answer's long-term value. – [Donald Duck](#) Dec 7 '20 at 9:02