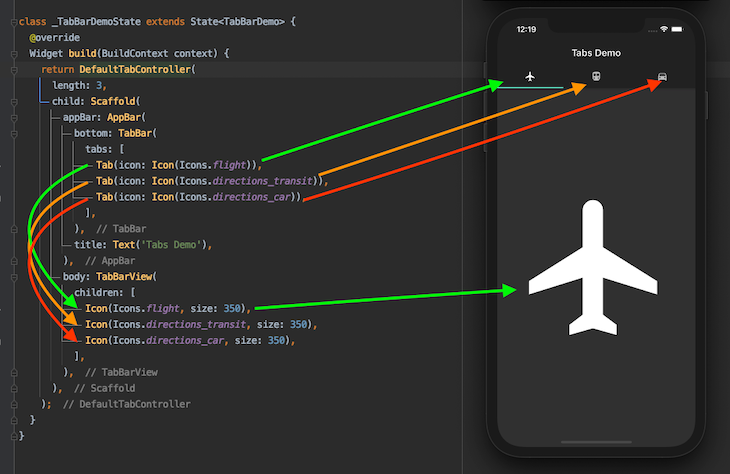
**TabBar in Flutter**

Description:

Apps often have different categories of content or features available to users. Naturally, you want your users to be able to quickly glance at each option and move between categories with a simple swipe. That’s where the Flutter TabBar class comes in.

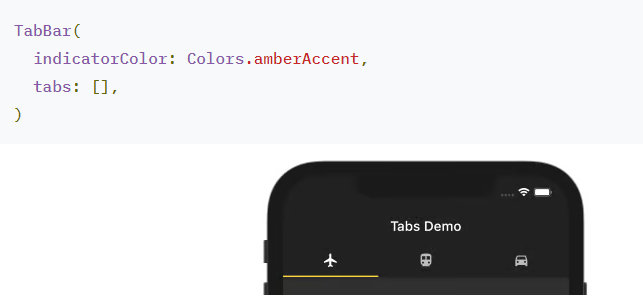
**To implement TabBar in your Flutter app, complete the following steps:**

1. Wrap the Scaffold widget inside the DefaultTabController. This should be used for most simple use cases. **If you want to control the tabs programmatically, you should use TabController and avoid this step.**
2. Place the TabBar widget as the bottom property of AppBar
3. Provide TabBarView in the body of the AppBar. TabBarView is like PageView, which is used mostly with TabBar because it shows the widget based on the currently selected tab



**How to customize the tab indicator in TabBar**

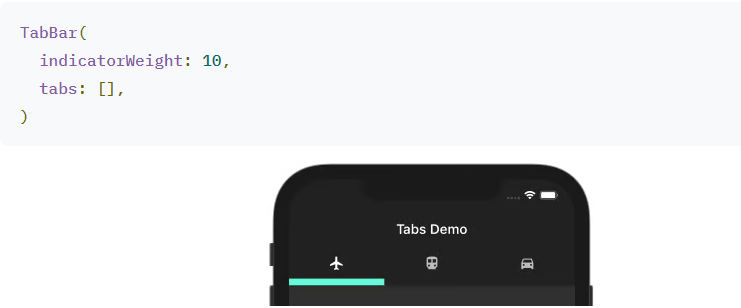
* Tab color



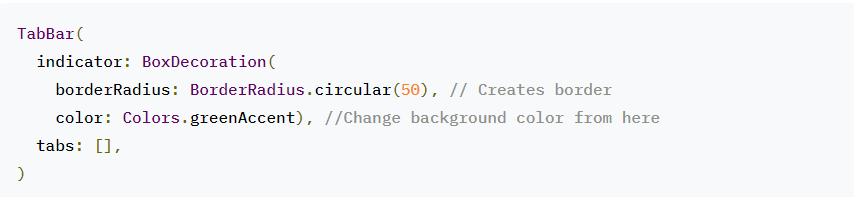
* **Tab size**



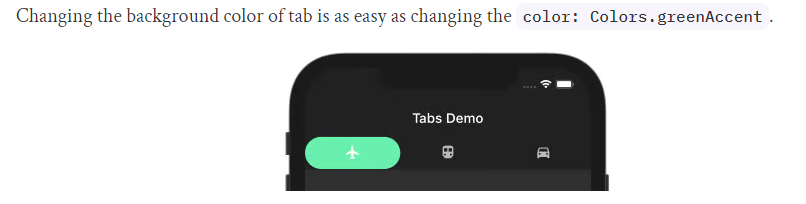
* **Tab height:**



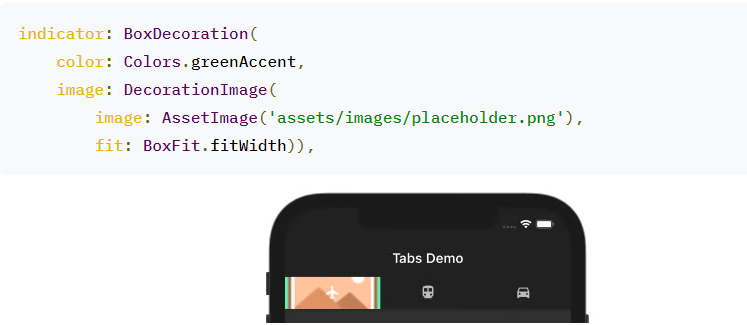
* **Change the indicator:**



* **Flutter TabBar background color:**



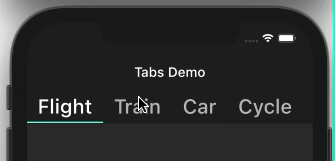
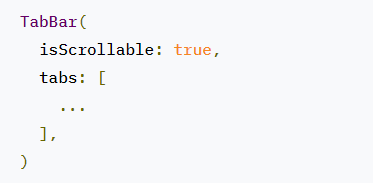
* **Background image:**



* **Create a custom indicator:**



* **Making scrollable tabs with TabBar:**
* **Horizontally scrollable tabs:**

****

* **Vertically scrollable tabs with AppBar:**

DefaultTabController(

length: 5,

child: Scaffold(

body: NestedScrollView(

headerSliverBuilder: (BuildContext context, bool innerBoxIsScrolled) {

return <Widget>[

new SliverAppBar(

 title: Text('Tabs Demo'),

pinned: true,

floating: true,

bottom: TabBar(

isScrollable: true,

tabs: [

Tab(child: Text('Flight')),

Tab(child: Text('Train')),

Tab(child: Text('Car')),

Tab(child: Text('Cycle')),

Tab(child: Text('Boat')),

],

),

),

];

},

body: TabBarView(

children: <Widget>[

Icon(Icons.flight, size: 350),

Icon(Icons.directions\_transit, size: 350),

Icon(Icons.directions\_car, size: 350),

Icon(Icons.directions\_bike, size: 350),

Icon(Icons.directions\_boat, size: 350),

],

),

)),

);

* **Changing tabs programmatically:**

TabController \_controller;

int \_selectedIndex = 0;

@override

void initState() {

super.initState();

\_controller = TabController(length: 5, vsync: this);

}

@override

void dispose() {

\_controller.dispose();

super.dispose();

}

//Assign here

bottom: TabBar(

controller: \_controller,

isScrollable: true,

tabs: [... ],

),

// And here

body: TabBarView(

controller: \_controller,

children: <Widget>[... ],

//controller: \_tabController,

),

Use the TabController to move to the next page with the click of a button:

onPressed: () {

\_controller.animateTo(\_selectedIndex += 1);

},