

Flutter AppBar 自定义顶部导航按钮 图标、颜色 以及 TabBar 定义顶部 Tab 切换

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html> （IT 营）

一、Flutter AppBar 自定义顶部按钮图标、颜色.....	1
二、Flutter AppBar 中自定义 TabBar 实现顶部 Tab 切换.....	3
三、Flutter 把 TabBar 放在导航最顶部.....	4
四、Flutter AppBar 中自定义 TabBar 实现 Tabs 的另一种方法。.....	6
五、Flutter Vscode 调试 Flutter 项目.....	7

一、Flutter AppBar 自定义顶部按钮图 标、颜色

属性	描述
leading	在标题前面显示的一个控件，在首页通常显示应用的 logo；在其他界面通常显示为返回按钮
title	标题，通常显示为当前界面的标题文字，可以放组件
actions	通常使用 IconButton 来表示，可以放按钮组
bottom	通常放 tabBar，标题下面显示一个 Tab 导航栏
backgroundColor	导航背景颜色
iconTheme	图标样式
textTheme	文字样式
centerTitle	标题是否居中显示

```
import 'package:flutter/material.dart';
class AppBardemoPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.red,
        leading: IconButton(
          icon: Icon(Icons.menu),
          tooltip: "Search",
          onPressed: () {
            print('menu Pressed');
          }
        ),
        title: Text('FlutterDemo'),
        actions: <Widget>[
          IconButton(
            icon: Icon(Icons.search),
            tooltip: "Search",
            onPressed: () {
              print('Search Pressed');
            }
          ),
          IconButton(
            icon: Icon(Icons.more_horiz),
            tooltip: "more_horiz",
            onPressed: () {
              print('more_horiz Pressed');
            }
          )
        ],
      ),
      body: Text('这是 AppBar'),
    );
  }
}
```

二、Flutter AppBar 中自定义 TabBar 实现顶部 Tab 切换

TabBar 常见属性:

属性	描述
tabs	显示的标签内容, 一般使用 Tab 对象,也可以是其他的 Widget
controller	TabController 对象
isScrollable	是否可滚动
indicatorColor	指示器颜色
indicatorWeight	指示器高度
indicatorPadding	底部指示器的 Padding
indicator	指示器 decoration, 例如边框等
indicatorSize	指示器大小计算方式, TabBarIndicatorSize.label 跟文字等宽,TabBarIndicatorSize.tab 跟每个 tab 等宽
labelColor	选中 label 颜色
labelStyle	选中 label 的 Style
labelPadding	每个 label 的 padding 值
unselectedLabelColor	未选中 label 颜色
unselectedLabelStyle	未选中 label 的 Style

```
import 'package:flutter/material.dart';
class AppBardemoPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: DefaultTabController(
        length: 2,
        child: Scaffold(
          appBar: AppBar(
            title: Text('FlutterDemo'),
            bottom: TabBar(
              tabs: <Widget>[
                Tab(text: "热门"),
                Tab(text: "推荐")
              ]
            ),
          ),
          body: TabBarView(
```

```
children: <Widget>[
  ListView(
    children: <Widget>[
      ListTile(title: Text("这是第一个 tab")),
      ListTile(title: Text("这是第一个 tab")),
      ListTile(title: Text("这是第一个 tab"))
    ],
  ),
  ListView(
    children: <Widget>[
      ListTile(title: Text("这是第二个 tab")),
      ListTile(title: Text("这是第二个 tab")),
      ListTile(title: Text("这是第二个 tab"))
    ],
  ),
],
);
}
```

三、Flutter 把 TabBar 放在导航最顶部

```
import 'package:flutter/material.dart';
class AppBardemoPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: DefaultTabController(
        length: 2,
        child: Scaffold(
          appBar: AppBar(
            // backgroundColor: Colors.red,
            leading: IconButton(
              icon: Icon(Icons.arrow_back),
              tooltip: "Search",
              onPressed: () {
                Navigator.of(context).pop();
              }
            )
          )
        )
      )
    );
  }
}
```

```
    }},
    title: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Expanded(
          flex: 1,
          child: TabBar(
            tabs: <Widget>[Tab(text: "热门"), Tab(text: "推荐")]
          ),
        )
      ],
    ),
  ),
  body: TabBarView(
    children: <Widget>[
      ListView(
        children: <Widget>[
          ListTile(title: Text("这是第一个 tab")),
          ListTile(title: Text("这是第一个 tab")),
          ListTile(title: Text("这是第一个 tab"))
        ],
      ),
      ListView(
        children: <Widget>[
          ListTile(title: Text("这是第二个 tab")),
          ListTile(title: Text("这是第二个 tab")),
          ListTile(title: Text("这是第二个 tab"))
        ],
      )
    ],
  ),
);
}
```

四、Flutter AppBar 中自定义 TabBar 实现 Tabs 的另一种方法。

```
import 'package:flutter/material.dart';

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

  _AppBardemoPageState createState() => _AppBardemoPageState();
}

class _AppBardemoPageState extends State<AppBardemoPage> with SingleTickerProviderStateMixin {

  TabController _tabController;

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

  void initState() {
    super.initState();
    _tabController = new TabController(vsync: this, length: 3);
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text('顶部 tab 切换'),
        bottom: new TabBar(
          tabs: <Widget>[
            new Tab(
              icon: new Icon(Icons.directions_bike),
            ),
            new Tab(
              icon: new Icon(Icons.directions_boat),
            ),
            new Tab(
              icon: new Icon(Icons.directions_bus),
            ),
          ],
        ),
      ),
    );
  }
}
```

```
    ],  
    controller: _tabController,  
  ),  
),  
body: new TabBarView(  
  controller: _tabController,  
  children: <Widget>[  
    new Center(child: new Text('自行车')),  
    new Center(child: new Text('船')),  
    new Center(child: new Text('巴士')),  
  ],  
),  
);  
}  
}
```

五、Flutter Vscode 调试 Flutter 项目

1、Vscode 中打开 flutter 项目进行开发

2、运行 Flutter 项目

```
flutter run
```

r 键： 点击后热加载，也就算是重新加载吧。

p 键： 显示网格，这个可以很好的掌握布局情况，工作中很有用。

o 键： 切换 android 和 ios 的预览模式。

q 键： 退出调试预览模式。

3、Vscode 默认连不上第三方模拟器解决方案

cd 到对应夜神模拟器 D:\Program Files\Nox\bin 目录 然后运行

```
nox_adb.exe connect 127.0.0.1:62001
```