

# Flutter 笔记 (四): PageView-Flutter 中的 ViewPager

Yun \ Lei

## PageView

使用 PageView 的 **三种方式**

```
PageView({  
  reverse 倒退  
  Key key,  
  this.scrollDirection = Axis.horizontal, //方向  
  this.reverse = false, //是否和阅读方向一样的滚动, 比如中文的  
  阅读习惯系从左往右  
  PageController controller, //控制器  
  this.physics, //页面视图如何响应用户输入  
  this.pageSnapping = true, //使用自定义滚动时禁止页面捕捉  
  this.onPageChanged, //页面切换回调  
  List<Widget> children = const <Widget>[], //页面 (组  
  件) 列表, 页面个数等于长度  
  this.dragStartBehavior = DragStartBehavior.start, //拖  
  拽行为  
})
```

后面两种方式去除重复数据

```
PageView.builder({  
  //...  
  @required IndexedWidgetBuilder itemBuilder, //创建  
  item, 根据回调index返回不同页面  
  int itemCount, //item数量  
  //...  
})
```

```
} )
```

第三种是自定义模式

```
PageView.custom({  
  //...  
  @required this.childrenDelegate, //自定义模式接受一个子页面委托对象  
  //...  
})
```

这里主要以 **PageView.builder** 介绍一下 PageView 的使用

## 1、使用

```
//组件使用  
PageView.builder(  
  itemBuilder: (context, index) {  
    return Center(  
      child: _getPageByIndex(index), //每个页面展示的组件  
    );  
  },  
  itemCount: 4, //页面数量  
  onPageChanged: _onPageChange, //页面切换  
  controller: _pageController, //控制器  
)  
//返回每个页面子组件  
StatefulWidget _getPageByIndex(int index) {  
  switch (index) {  
    case 0:  
      return HomePage();  
    case 1:
```

```

        return CategoryPage();
    case 2:
        return MeiZiPage();
    case 3:
        return AccountPage();
    default:
        return HomePage();
    }
}

```

## 2、关于 controller

```

PageController({
    this.initialPage = 0,      //初始化选择页面
    this.keepPage = true,     //是否保持已经渲染过得页面
    this.viewportFraction = 1.0, //每个页面应占用的视口部分。
                                0~1之间
})

```

在是使用 PageView 的时候会包含一个默认的控制器的

```

var _pageController = PageController(initialPage: 0);

```

控制器跳转有 4 个已有的方法

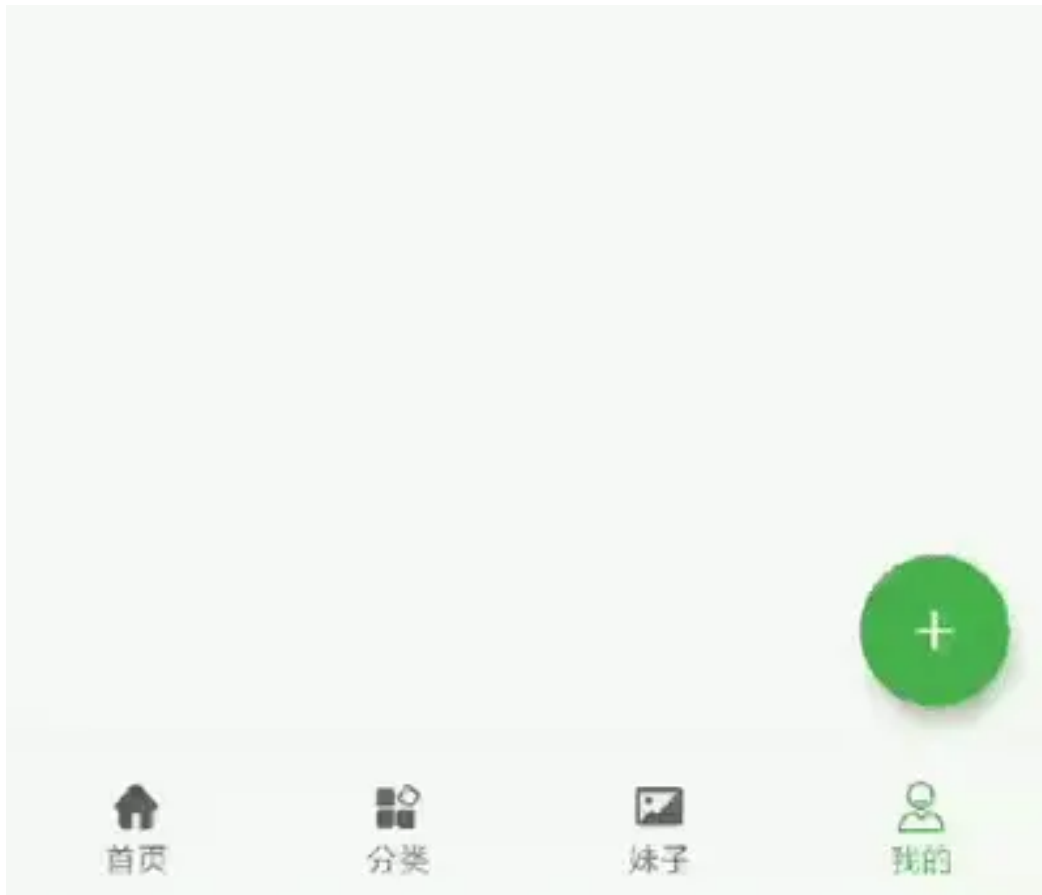
- animateToPage: 带动画跳转
- jumpToPage: 直接改变当前页面无动画
- nextPage: 下一页

- `previousPage`: 上一页

来先看看 `jumpToPage` 的效果

```
_pageController.jumpToPage(_selectedIndex);
```





jumpToPage

再看看 animateToPage 的效果，ps:切换时间 500ms，切换效果 Curves.easeInOut

```
_pageController.animateToPage(_selectedIndex,  
    duration: Duration(milliseconds: 500), curve:  
    Curves.easeInOut);
```



this is HomePage!



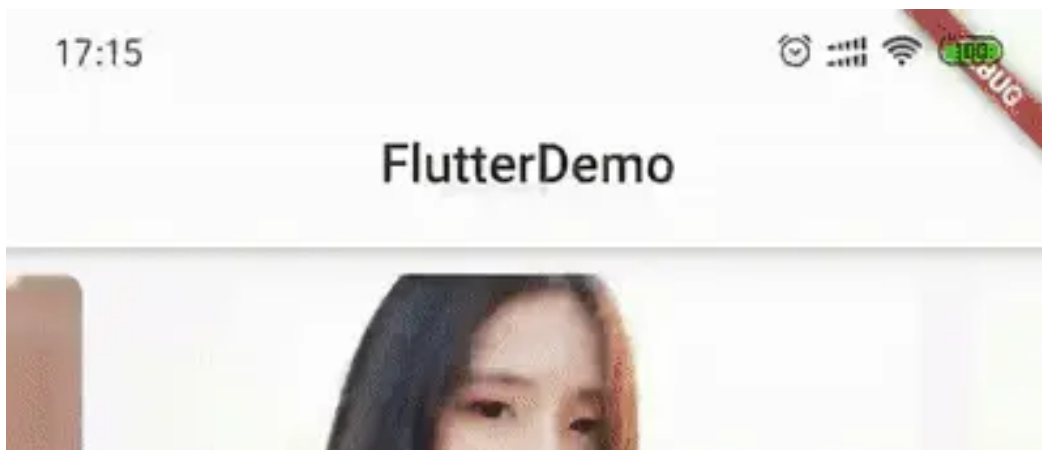


animateToPage

### 3、PageController的viewportFraction

每个页面应占用的视口部分。0~1之间，比如设置为0.8可以得到以下效果，直接上图是最好的解释。再加点自动播放，循环播放之类的就是一个很nice的banner，但是这里只做demo演示，讲解以下这个属性，一般在flutter中banner不这样做，这样会有很多问题，比如拖拽时停止自动，pageView嵌套等问题。Flutter的概念里一切都是组件，banner组件我推荐轮子[flutter\\_swiper](#)

```
var _pageController = PageController(initialPage: 1, viewportFraction: 0.8);
```





首页



分类



妹子



我的



viewportFraction: 0.8

## 4、配合 BottomNavigationBar 使用

关于 BottomNavigationBar 的使用可以参考我之前的

[Flutter 笔记 \(一\): BottomNavigationBar 常见问题](#)

PageView 和 BottomNavigationBar 联动的完整代码如下

```
class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);
  final String title;

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

class _MyHomePageState extends State<MyHomePage> {
  int _selectedIndex = 0;
  var _pageController = PageController(initialPage: 0);

  void _incrementCounter() {
    setState(() {
      _selectedIndex++;
    });
  }

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
    if (_pageController.hasClients) {

```

```

        _pageController.animateToPage(_selectedIndex,
            duration: Duration(milliseconds: 500), curve:
Curves.easeInOut);
//        _pageController.jumpToPage(_selectedIndex);
    }
}

void _onPageChange(int index) {
    print("_onPageChange");
    setState(() {
        _selectedIndex = index;
    });
}

StatefulWidget _getPageByIndex(int index) {
    switch (index) {
        case 0:
            return HomePage();
        case 1:
            return CategoryPage();
        case 2:
            return MeiZiPage();
        case 3:
            return AccountPage();
        default:
            return HomePage();
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: PreferredSize(
            child: AppBar(

```

```

        backgroundColor: Colors.white,
        brightness: Brightness.light,
        title: Center(
          child: Text(
            "FlutterDemo",
          ),
        ),
      ),
      preferredSize: Size(double.infinity, 60)),
    body: PageView.builder(
      itemBuilder: (context, index) {
        return Center(
          child: _getPageByIndex(index), //每个页面展示的组
件
        );
      },
      itemCount: 4, //页面数量
      onPageChanged: _onPageChange, //页面切换
      controller: _pageController, //控制器
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: _incrementCounter,
      tooltip: 'Increment',
      child: Icon(Icons.add),
    ),
    bottomNavigationBar: new BottomNavigationBar(
      items: [
        BottomNavigationBarItem(
          icon: new Icon(Icons.home), title: new
Text("首页")),
        BottomNavigationBarItem(
          icon: new Icon(Icons.category), title:
new Text("分类")),
        BottomNavigationBarItem(
          icon: new Icon(Icons.picture_in_frame), title: new

```

```
Text("妹子")),  
      BottomNavigationBarItem(  
        icon: new Icon(Icons.account_circle), title:  
new Text("我的")),  
    ],  
    selectedItemColor: Color(0xFF4CAF50),  
    unselectedItemColor: Color(0xFF666666),  
    type: BottomNavigationBarType.fixed,  
    showUnselectedLabels: true,  
    onTap: _onItemTapped,  
    currentIndex: _selectedIndex,  
    selectedFontSize: 12.0,  
  ),  
);  
}  
}
```

## 完整代码

<https://github.com/leiyun1993/FlutterDemo-GankIO>