

Flutter -- PageView 和 PageController

惊世骇俗

在Flutter中，PageView和PageController通常一起使用来创建分页视图，允许用户左右滑动以在不同的页面间切换。以下是PageView和PageController的具体用法：

PageView

PageView是一个可以水平或垂直滑动的组件，用于展示一系列的页面。它支持无限滚动和基于手势的导航。

基本属性：

- `children`：一个包含将要显示的子组件列表。
- `scrollDirection`：滚动方向，可以是`Axis.horizontal`或`Axis.vertical`。
- `physics`：决定滚动行为的物理学模型。
- `pageSnapping`：是否会在用户停止滑动时，页面会吸附到最近的页面。

示例代码：

```
PageView(  
  scrollDirection: Axis.horizontal, // 水平滑动  
  children: <Widget>[  
    // 子组件列表  
    Text('Page 1'),  
    Text('Page 2'),  
    // 更多页面...
```

```
],  
)
```

PageController

PageController用于控制PageView的页面。它可以指定初始页面、控制页面切换、监听页面变化事件等。

基本用法：

- 创建PageController实例，并将其传递给PageView的controller属性。
- 使用PageController的animateToPage方法来跳转到特定页面。
- 使用listen方法来监听页面变化事件。

示例代码：

```
class MyPageView extends StatefulWidget {  
  @override  
  _MyPageViewState createState() => _MyPageViewState();  
}  
  
class _MyPageViewState extends State<MyPageView> {  
  final PageController _pageController = PageController();  
  
  @override  
  void dispose() {  
    _pageController.dispose(); // 重要：释放资源  
  }  
}
```

```

    super.dispose();
}

void _onPageChanged(int page) {
    print('当前页面是 $page');
}

void _goToNextPage() {
    if (_pageController.hasClients) {
        _pageController.nextPage(duration:
Duration(milliseconds: 500), curve: Curves.easeIn);
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        body: PageView(
            controller: _pageController, // 使用PageController控
            制PageView
            onPageChanged: _onPageChanged, // 页面变化监听
            children: <Widget>[
                // 子组件列表
                Text('Page 1'),
                Text('Page 2'),
                // 更多页面...
            ],
        ),
        floatingActionButton: FloatingActionButton(
            onPressed: _goToNextPage, // 点击跳转到下一页
            tooltip: 'Next Page',
            child: Icon(Icons.navigate_next),
        ),
    );
}

```

```
}
```

在这个示例中，我们创建了一个 `PageController` 实例，并将其用于控制 `PageView`。我们监听页面变化，并在控制台打印当前页面。此外，我们还提供了一个按钮来触发跳转到下一页的动画。

高级用法

- **无限滚动：**设置 `PageView` 的 `loop` 属性为 `true`，可以创建无限滚动的效果。
- **动态更新页面：**可以通过修改传递给 `PageView` 的 `children` 列表来动态添加或删除页面。
- **自定义滚动行为：**通过 `physics` 属性，可以自定义滚动的物理效果，例如弹性、阻尼等。

使用 `PageView` 和 `PageController`，你可以创建流畅的分页视图，适用于多种场景，如图片轮播、多步骤表单、滑动导航等。