

Flutter 可滚动 Widget —— PageView

PageView 是可以一页一页滑动的 可滚动Widget。其 子Widget 会占据当前屏幕的所有可见区域。

PageView 的使用

PageView 的使用有三种方式：

1. 使用默认的构造函数
2. 使用 PageView.builder
3. 使用 PageView.custom

1.使用默认的构造函数，给 children 属性赋值

代码所在位置

`flutter_widget_demo/lib/pageview/PageViewDefaultWidget.da`

使用方法

使用默认构造函数写 PageView，只适用于那些只有少量 子Widget 的 PageView。

demo 如下：

```

import 'package:flutter/material.dart';

void main() => runApp(PageViewDefaultWidget());

class PageViewDefaultWidget extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Test',
      home: Scaffold(
        appBar:
AppBar(title: new Text('Flutter 可滚动Widget -- PageView')),
        body: PageView(
          onPageChanged: (index){
            print('current page $index ');
          },
          children: <Widget>[
            ListTile(title: Text('Title0')),
            ListTile(title: Text('Title1')),
            ListTile(title: Text('Title2')),
            ListTile(title: Text('Title3')),
            ListTile(title: Text('Title4')),
          ],
        )),
    );
  }
}

```

运行效果为：



6:21

....



Debug

Flutter 可滚动Widget -- PageView

Title2

可以左右滑动切换页面。

2.使用 `PageView.builder`,可用于和数据绑定实现大量或无限的列表

代码所在位置

`flutter_widget_demo/lib/pageview/PageViewBuilderWidget.da`

使用方法

`PageView.builder` 可以和数据绑定，用于构建大量或无限的列表。而且只会构建那些实际可见的 子Widget。

`PageView.builder` 的定义如下：

```
PageView.builder({
  Key key,
  this.scrollDirection = Axis.horizontal,
  this.reverse = false,
  PageController controller,
  this.physics,
  this.pageSnapping = true,
  this.onPageChanged,
  @required IndexedWidgetBuilder itemBuilder,
  int itemCount,
  this.dragStartBehavior =
DragStartBehavior.down,
})
```

多了和 `ListView.builder` 类似的 `itemCount` 和 `itemBuilder` 属性，用法也是一样的：

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

void main() => runApp(PageViewBuilderWidget(
    items: List<String>.generate(10000, (i) =>
    "Item $i"),
    ));

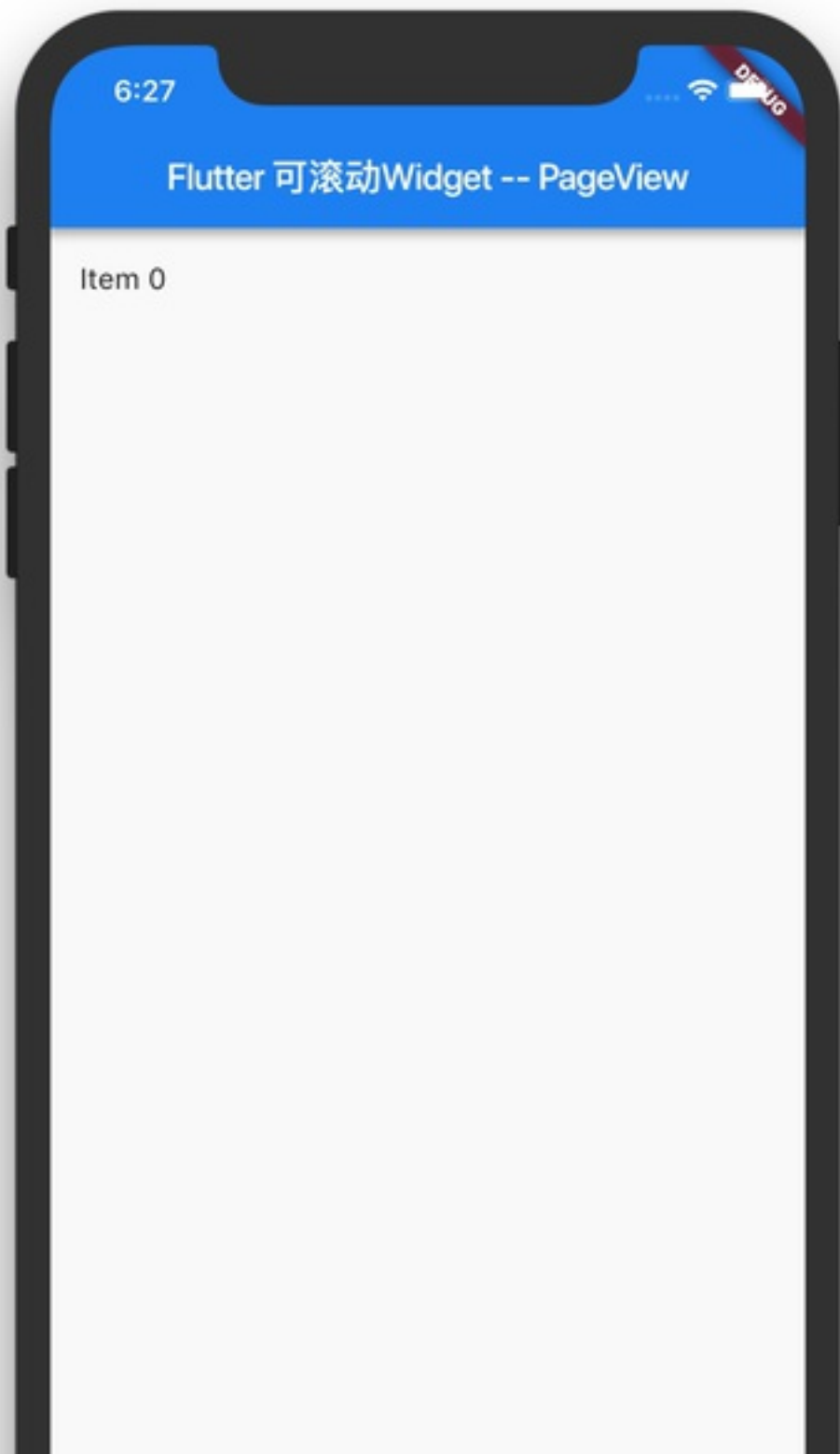
class PageViewBuilderWidget extends
StatelessWidget {
    final List<String> items;

    PageViewBuilderWidget({Key key, @required
this.items}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Test',
            home: Scaffold(
                appBar: AppBar(title: new Text('Flutter 可
滚动Widget -- PageView')),
                body: PageView.builder(
                    onPageChanged: (index) {
                        print('current page $index ');
                    },
                    itemCount: items.length,
                    itemBuilder: (context, index) {
                        return ListTile(
                            title: Text('${items[index]}'),
                        );
                    },
                ),
            ),
        );
    }
}
```

```
}  
  }  
);
```

运行效果如下：





可以左右滑动切换页面。

3.使用 PageView.custom

代码所在位置

flutter_widget_demo/lib/pageview/PageViewCustomWidget.dart

使用方法

PageView.custom 的定义如下：

```
PageView.custom({  
  Key key,  
  this.scrollDirection = Axis.horizontal,  
  this.reverse = false,  
  PageController controller,  
  this.physics,  
  this.pageSnapping = true,  
  this.onPageChanged,  
  @required this.childrenDelegate,  
  this.dragStartBehavior =  
  DragStartBehavior.down,  
  })
```

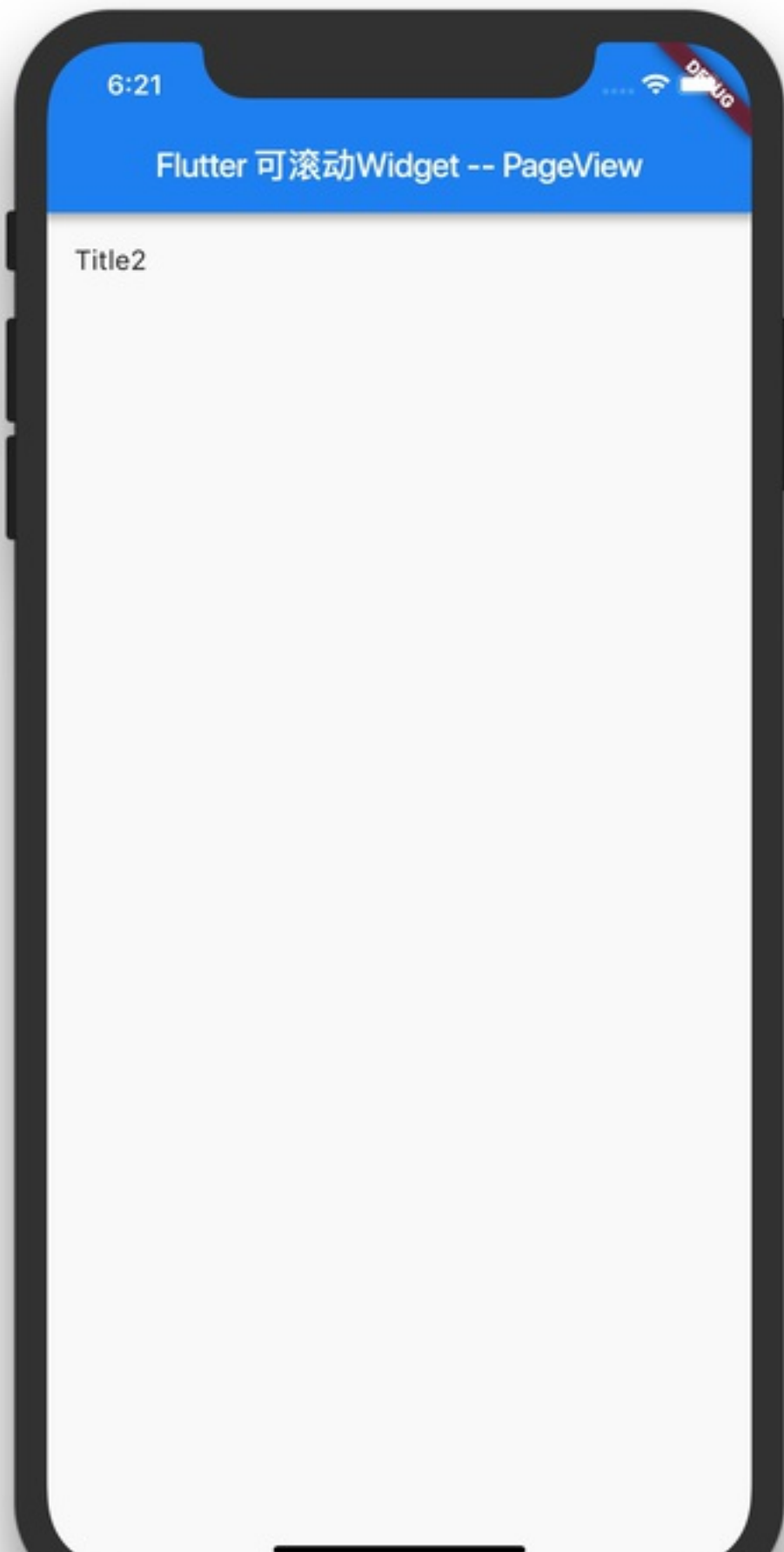
增加了 childrenDelegate 的属性，类型为 SliverChildDelegate，具有定制 子Widget 的能力，和 ListView.custom 里的一样，所以用法也一样：

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

void main() => runApp(PageViewCustomWidget());

class PageViewCustomWidget extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Test',
      home: Scaffold(
        appBar: AppBar(title: new Text('Flutter
可滚动Widget -- PageView')),
        body: PageView.custom(
          onPageChanged: (index) {
            print('current page $index ');
          },
          childrenDelegate:
SliverChildListDelegate(<Widget>[
            ListTile(title: Text('Title0')),
            ListTile(title: Text('Title1')),
            ListTile(title: Text('Title2')),
            ListTile(title: Text('Title3')),
            ListTile(title: Text('Title4')),
          ]),
        ),
      );
  }
}
```


运行效果如下：



可以左右滑动切换页面。

PageView 的构造函数及参数说明

先看 PageView 的构造函数：

```
class PageView extends StatefulWidget {  
  PageView({  
    Key key,  
    this.scrollDirection = Axis.horizontal,  
    this.reverse = false,  
    PageController controller,  
    this.physics,  
    this.pageSnapping = true,  
    this.onPageChanged,  
    List<Widget> children = const <Widget>[],  
    this.dragStartBehavior =  
    DragStartBehavior.down,  
  }) :  
    ...  
}
```

参数名字

参数类型

意义

key	Key	Widget 的标识
scrollDirection	Axis	滑动的方向 默认为 Axis.vertical， 向可滑动
reverse	bool	控制 PageView 里列表 列顺序，是按照插入顺 还是按照插入顺序相反 排序。 默认为 false，就是按照 顺序排序，第一个插入 部 ，当 reverse 为 true 时 一个插入的会在底部
controller	PageController	PageController 可以控 制到哪一页，还有其他
physics	ScrollPhysics	设置 PageView 的滚动 应该使用 PageScrollPh 默认值为 false
pageSnapping	bool	设置为false以禁用页面 对自定义滚动行为很有
onPageChanged	ValueChanged<int>	当 PageView 当前页面 时候调用
children	List<Widget>	PageView 的列表项
semanticChildCount	int	提供语义信息的列表项 默认为 PageView 的 it 数量 确定处理拖动开始行为 式。 如果设置为 [DragStartBehavior.st

dragStartBehavior	DragStartBehavior	则在检测到拖动手势时 滚动拖动行为 如果设置为 [DragStartBehavior.d 它将在首次检测到向下 开始
-------------------	-------------------	---