

Flutter GridView 组件 以及动态 GridView

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 列表组件概述.....	1
二、Flutter GridView 组件的常用参数.....	1
三、Flutter GridView.count 实现网格布局.....	2
四、Flutter GridView.builder 实现网格布局.....	4
五、Vscode 调试 Flutter 项目.....	5

一、Flutter 列表组件概述

列表布局是我们项目开发中最常用的一种布局方式。Flutter 中我们可以通过 `ListView` 来定义列表项，支持垂直和水平方向展示。通过一个属性就可以控制列表的显示方向。列表有以下分类：

- 1、垂直列表
- 2、垂直图文列表
- 3、水平列表
- 4、动态列表
- 5、矩阵式列表（网格布局）

二、Flutter GridView 组件的常用参数

当数据量很大的时候用矩阵方式排列比较清晰。此时我们可以用网格列表组件 `GridView` 实现布局。

`GridView` 创建网格列表有多种方式，下面我们主要介绍两种。

- 1、可以通过 `GridView.count` 实现网格布局
- 2、通过 `GridView.builder` 实现网格布局

常用属性:

名称	类型	说明
<code>scrollDirection</code>	<code>Axis</code>	滚动方法
<code>padding</code>	<code>EdgeInsetsGeometry</code>	内边距
<code>resolve</code>	<code>bool</code>	组件反向排序
<code>crossAxisSpacing</code>	<code>double</code>	水平子 Widget 之间间距
<code>mainAxisSpacing</code>	<code>double</code>	垂直子 Widget 之间间距
<code>crossAxisCount</code>	<code>int</code>	一行的 Widget 数量
<code>childAspectRatio</code>	<code>double</code>	子 Widget 宽高比例
<code>children</code>		<code><Widget>[]</code>
<code>gridDelegate</code>	<code>SliverGridDelegateWithFixedCrossAxisCount</code> (常用) <code>SliverGridDelegateWithMaxCrossAxisExtent</code>	控制布局主要用在 <code>GridView.builder</code> 里面

三、Flutter `GridView.count` 实现网格布局

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

import 'res/listData.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('FlutterDemo')),
        body: LayoutContent(),
      ));
  }
}
```

```
}  
}  
  
class LayoutContent extends StatelessWidget {  
  
  List<Widget> _getListData() {  
    var tempList=listData.map((value){  
      return Container(  
        child: Column(  
          children: <Widget>[  
            Image.network(value["imageUrl"]),  
            SizedBox(height: 12),  
            Text(value["title"],textAlign: TextAlign.center,style:TextStyle(fontSize: 20)),  
          ],  
        ),  
        decoration: BoxDecoration(  
          border: Border.all(  
            color:Color.fromRGBO(230, 230, 230, 0.9),  
            width: 1.0  
          )  
        ),  
      );  
    });  
    // ('124124','124214')  
    return tempList.toList();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
    return GridView.count(  
      padding: EdgeInsets.all(20),  
      crossAxisCount: 2,  
      crossAxisSpacing: 20,  
      mainAxisSpacing: 20,  
      // childAspectRatio:0.7,  
      children: this._getListData(),  
    );  
  }  
}
```

四、Flutter GridView.builder 实现网格布局

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

import 'res/listData.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('FlutterDemo')),
        body: LayoutContent(),
      ));
  }
}

class LayoutContent extends StatelessWidget {

  Widget _getListData(context,index) {
    return Container(
      child: Column(
        children: <Widget>[
          Image.network(listData[index]["imageUrl"]),
          SizedBox(height: 12),
          Text(listData[index]["title"],textAlign: TextAlign.center,style:TextStyle(fontSize:
20)),
        ],
      ),
      decoration: BoxDecoration(
        border: Border.all(
          color:Color.fromRGBO(230, 230, 230, 0.9),
          width: 1.0
        )
      ),
    );
  }
}
```

```
}  
  
@override  
Widget build(BuildContext context) {  
  // TODO: implement build  
  return GridView.builder(  
    itemCount: listData.length,  
    gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
      //横轴元素个数  
      crossAxisCount: 2,  
      //纵轴间距  
      mainAxisSpacing: 20.0,  
      //横轴间距  
      crossAxisSpacing: 10.0,  
      //子组件宽高长度比例  
      childAspectRatio: 1.0  
    ),  
    itemBuilder: this._getListData,  
  );  
}
```

五、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

湖北众猿腾网络科技有限公司

nox_adb.exe connect 127.0.0.1:62001