



链滴

Android 教程 2020 - RecyclerView 使用入门

作者: [RustFisher](#)

原文链接: <https://ld246.com/article/1580696144715>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

本文介绍 RecyclerView 的使用入门。这里给出一种比较常见的使用方式。

[Android教程2020 - 系列总览](#)

[本文链接](#)

想必读者朋友对列表的表现形式已经不再陌生。手机上有联系人列表，文件列表，短信列表等等。本文讲述的是在 Android 开发中用 RecyclerView 来实现列表效果。

使用步骤

引入 RecyclerView

在 app 的 build.gradle 文件中添加引用。我们使用的是 androidx 包。

```
dependencies {  
    // ...  
    implementation 'androidx.recyclerview:recyclerview:1.1.0'  
}
```

数据准备

先确定一下要显示什么样的数据。是用户信息，联系人，或是文件。这里以字符为例。

写代码前，我们先考虑一下需求，就是要如何显示，怎么显示数据。日常工作中，一般会有 UI 效果。本文中的美术设计就由我们自己发挥了。

比如显示 a 97。

ViewHolder 与 layout

现在要显示的数据已经确定了。来设计 UI 表现。layout 与 ViewHolder 紧密相关。在设计 Adapter 类前，先写 ViewHolder 类会比较好。

新建定义 item（列表子项）的布局文件 [item_letter.xml](#)。

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal">  
  
    <TextView  
        android:id="@+id/tv1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:textColor="#000000" />  
  
    <TextView  
        android:id="@+id/tv2"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"
```

```
    android:textColor="#000000" />
```

```
</LinearLayout>
```

这里使用内部类，把 ViewHolder 类写在 activity 类里面。

```
private class VH extends RecyclerView.ViewHolder {
    TextView tv1;
    TextView tv2;

    public VH(@NonNull View itemView) {
        super(itemView);
        tv1 = itemView.findViewById(R.id.tv1);
        tv2 = itemView.findViewById(R.id.tv2);
    }
}
```

从上面可以看出，ViewHolder 和 item 的 layout 是紧密相关的。

layout 里面的 id 写的比较简单。实际项目中可以给一些更有意义的 id 命名。

在 activity 的 layout 中，添加 RecyclerView。

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/re_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

设计 Adapter

设计一个适配器继承自 `RecyclerView.Adapter<VH>`。这里的 VH 是我们上面写好的 ViewHolder。`LetterAdapter` 中持有它自己的数据列表。需要实现 3 个方法。

`onCreateViewHolder` 方法，要求返回 VH 对象。这里就是创建 VH 对象并返回。

而 VH 构造器要求传入一个 View，我们利用 `LayoutInflater` 创建一个 view 给它。当然，创建的根据是前面设计好的 `item_letter`。

`onBindViewHolder` 是把数据交给对应的 VH 来显示。

`getItemCount` 方法要求返回数据的数量。

```
private class LetterAdapter extends RecyclerView.Adapter<VH> {

    private List<Character> dataList;

    public LetterAdapter(List<Character> dataList) {
        this.dataList = dataList;
    }

    @NonNull
    @Override
    public VH onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new VH(LayoutInflater.from(parent.getContext()).inflate(R.layout.item_letter, parent, false));
    }
}
```

```

    }

    @Override
    public void onBindViewHolder(@NonNull VH holder, int position) {
        Character c = dataList.get(position);
        holder.tv1.setText(c.toString());
        holder.tv2.setText(String.valueOf(Integer.valueOf(c)));
    }

    @Override
    public int getItemCount() {
        return dataList.size();
    }
}

```

粗心大意的时候，可能会忘记初始化 adapter 里的 dataList。报一个 null pointer 异常。

设置 RecyclerView

在 Activity 的 onCreate 方法里进行了初始化。

RecyclerView 需要 2 项设置，适配器与 LayoutManager。适配器就是我们上面定的那个。

LayoutManager 这里用 **LinearLayoutManager**，指定为垂直方向，这样我们会得到一个上下滑动的表。

```

List<Character> characterList = new ArrayList<>();
for (char c = 'a'; c <= 'z'; c++) {
    characterList.add(c);
}

mLetterAdapter = new LetterAdapter(characterList);
RecyclerView letterReView = findViewById(R.id.re_view);
letterReView.setAdapter(mLetterAdapter);
letterReView.setLayoutManager(new LinearLayoutManager(this, RecyclerView.VERTICAL,
also));

```

观察运行结果

运行到手机或者模拟器上，打开这个 activity。有的朋友就发现，为啥屏幕里一个子项 item 就占满整个屏幕？

因为我们前面设置的 item 占满屏幕。回到 `item_letter.xml`，看一下根 layout 的设置。

把 `layout_height="match_parent"` 里的设置改为 `wrap_content`。

重新编译运行一下再看看结果。

也可以给根 layout 设置固定的高度。具体视美术设计和需求而定。

小结

到这里我们掌握了 RecyclerView 的基本使用方式。

接下来还要了解如何自定义它的表现形式。例如点击或长按 item 的反应；item 之间添加分割线；增

删/改数据；使用瀑布流样式等等。

RecyclerView 使用了[适配器模式](#)。

可以参考官方的[RecyclerView文档](#)。

工程放这里：<https://github.com/AnRFDev/Tutorial2020>

相关阅读

[RecyclerView - 使用入门](#)

[RecyclerView点击事件 - 如何设置点击事件](#)

[RecyclerView示例 - 实际使用](#)

[RecyclerView获取滑动距离](#)

[RecyclerView显示多种item](#)