

# Android 教程 2020 - RecyclerView 实际使用

作者: [RustFisher](#)

原文链接: <https://ld246.com/article/1580866946194>

来源网站: [链滴](#)

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

示例，用RecyclerView的item做出一个列表。

[Android教程2020 - 系列总览](#)

[本文链接](#)

前面我们已经知道如何用RecyclerView显示一系列数据。这里我们做出一个具体的例子。尽量做好看一。

这个示例会分为以下几个步骤：

- 确定数据。这里用的是模拟数据。
- 设计UI和表现形式。
- 编写layout与适配器。

## 模拟数据

先模拟一个数据。新建一个类DataTest，它有4个属性。

```
public class DataTest {
    private String timezone;
    private int number;
    private int personCount;
    private int count;

    public DataTest(String timezone, int number, int personCount, int count) {
        this.timezone = timezone;
        this.number = number;
        this.personCount = personCount;
        this.count = count;
    }
    // getter setter...
}
```

## 设计 item 的布局

设计 UI，在一行里显示 4 个属性值。item 用的是item\_recy2.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="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/tv1"
        style="@style/RePage2Header"
        android:layout_marginEnd="@dimen/re_2_half_gap" />

    <TextView
        android:id="@+id/tv2"
        style="@style/RePage2Header"
```

```

        android:layout_marginStart="@dimen/re_2_half_gap"
        android:layout_marginEnd="@dimen/re_2_half_gap" />

<TextView
    android:id="@+id/tv3"
    style="@style/RePage2Header"
    android:layout_marginStart="@dimen/re_2_half_gap"
    android:layout_marginEnd="@dimen/re_2_half_gap" />

<TextView
    android:id="@+id/tv4"
    style="@style/RePage2Header"
    android:layout_marginStart="@dimen/re_2_half_gap" />

</LinearLayout>

```

## 准备颜色，尺寸等资源

相关的 style 和颜色，尺寸配置文件，在 `res/values` 目录下。

style 文件 `style.xml`。

```

<style name="RePage2Header">
    <item name="android:layout_width">0dp</item>
    <item name="android:layout_height">40dp</item>
    <item name="android:layout_weight">2</item>
    <item name="android:textColor">#ffffff</item>
    <item name="android:gravity">center</item>
    <item name="android:background">@color/rePage2Item</item>
</style>

```

我们给 layout 里的每个 TextView 都设置了 layout\_width 为 0dp。是为了使用 layout\_weight 属

。让它们 4 个 TextView 按比例分割父 View 的宽度。

颜色配置文件 `color.xml`，添加如下颜色设置。

```

<color name="rePage2Item">#082941</color>

```

尺寸配置 `dimens.xml`。

```

<dimen name="re_2_gap">4dp</dimen>
<dimen name="re_2_half_gap">2dp</dimen>

```

## 设计 ViewHolder

资源文件和 layout 准备妥当，开始写对应的 viewHolder。这里也是把 VH 类和 Adapter 类放在 activity 类里面。

```

private class VH extends RecyclerView.ViewHolder {
    TextView tv1;
    TextView tv2;
    TextView tv3;
}

```

```

    TextView tv4;

    public VH(@NonNull View itemView) {
        super(itemView);
        tv1 = itemView.findViewById(R.id.tv1);
        tv2 = itemView.findViewById(R.id.tv2);
        tv3 = itemView.findViewById(R.id.tv3);
        tv4 = itemView.findViewById(R.id.tv4);
    }
}

```

## 设计 Adapter 类

适配器 Adapter 类。

```

private class Adapter extends RecyclerView.Adapter<VH> {

    private List<DataTest> dataList = new ArrayList<>();

    public Adapter() {
    }

    public void setDataList(List<DataTest> dataList) {
        this.dataList = dataList;
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public VH onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new VH(LayoutInflater.from(parent.getContext()).inflate(R.layout.item_recy2, parent, false));
    }

    @Override
    public void onBindViewHolder(@NonNull VH holder, int position) {
        DataTest dataTest = dataList.get(position);
        holder.tv1.setText(dataTest.getTimezone());
        holder.tv2.setText(String.valueOf(dataTest.getNumber()));
        holder.tv3.setText(String.valueOf(dataTest.getPersonCount()));
        holder.tv4.setText(String.valueOf(dataTest.getCount()));
    }

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

```

## 配置 RecyclerView

设置 recyclerview。

```

private Adapter mAdapter = new Adapter();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.act_recy_2);
    RecyclerView recyclerView = findViewById(R.id.re_view);
    recyclerView.setLayoutManager(new LinearLayoutManager(this, RecyclerView.VERTICAL,
also));
    recyclerView.setAdapter(mAdapter);
    mAdapter.setDataList(genDataTestList());
    recyclerView.addItemDecoration(new RecyclerView.ItemDecoration() {
        @Override
        public void getItemOffsets(@NonNull Rect outRect, @NonNull View view, @NonNull
RecyclerView parent, @NonNull RecyclerView.State state) {
            super.getItemOffsets(outRect, view, parent, state);
            outRect.top = getResources().getDimensionPixelOffset(R.dimen.re_2_gap);
        }
    });
}

// 生成模拟数据
private List<DataTest> genDataTestList() {
    List<DataTest> list = new ArrayList<>();
    for (int i = 1; i <= 60; i++) {
        DataTest d = new DataTest("时区" + i, i, i, i);
        list.add(d);
    }
    return list;
}

```

在 onCreate 方法中配置 recyclerView。

`recyclerView.addItemDecoration`方法是给 item 设置间隔样式。

`getItemOffsets`可以设置子项的间距。这里给子项底部一个间距值。具体数值设置在 `dimen` 中。

`genDataTestList()`是生成模拟的数据。

## 运行效果

运行起来即可看到效果。

测试记录			
时区	序号	人员	数量
时区1	1	1	1
时区2	2	2	2
时区3	3	3	3
时区4	4	4	4
时区5	5	5	5
时区6	6	6	6
时区7	7	7	7
时区8	8	8	8
时区9	9	9	9
时区10	10	10	10
时区11	11	11	11
时区12	12	12	12
时区13	13	13	13

## 使用 include 修改 layout

看到效果图，有的朋友问：表头和 item 的结构是一样的，可以复用吗？

其实是可以的。我们可以在 layout 中使用 `include` 标签，把另一个 layout 文件“包含”进来。

复制 `act_recy_2.xml` 粘贴得到 `act_recy_2_include.xml`，把原来的表头的 `LinearLayout` 改成 `include`。

```
<include layout="@layout/item_recy2" />
```

给 `include` 设定 layout，即我们定义的 item 的布局 `item_recy2`。

当我们想添加个 `margin-top` 的时候，比如这样

```
<include  
    android:layout_marginTop="4dp"  
    layout="@layout/item_recy2" />
```

as 会弹出警告：

Layout parameter `layout_marginTop` ignored unless both `layout_width` and `layout_height` are also specified on `<include>` tag

也就是在 `include` 标签中，如果要设置其他属性，需要先设置 `layout_width` 和 `layout_height`。

修改一下，再加个 `id`，变成这样。

```
<include  
    android:id="@+id/header"  
    layout="@layout/item_recy2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="4dp" />
```

让 Activity 使用这个 layout。修改一下[RecyclerViewDemo2Act](#)。

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.act_recy_2_include);
    initHeader();
    // 配置RecyclerView的部分
}

private void initHeader() {
    View header = findViewById(R.id.header);
    TextView tv1 = header.findViewById(R.id.tv1);
    tv1.setText("时区");
    TextView tv2 = header.findViewById(R.id.tv2);
    tv2.setText("序号");
    TextView tv3 = header.findViewById(R.id.tv3);
    tv3.setText("人员");
    TextView tv4 = header.findViewById(R.id.tv4);
    tv4.setText("数量");
}
```

我们添加了一个方法[initHeader\(\)](#)。里面先把 header 找到，通过 header 找到它的子 view，也就是那 4 个 TextView。

分别设置文字即可。

运行看看效果，是不是和前面的一样。

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](#)