# ListView和RecycleView总结

目录

[ListView和RecycleView总结 1](#_Toc42950521)

[一、 ListView的基本使用 2](#_Toc42950522)

[1.1使用ListView的一般步骤： 2](#_Toc42950523)

[1.2适配器的使用 2](#_Toc42950524)

[二、 RecycleView的基本使用 3](#_Toc42950525)

[三、 ListView和RecycleView的优化 5](#_Toc42950526)

## 一、ListView的基本使用

### 1.1使用ListView的一般步骤：

* 在布局文件中编写代码（xml）
* 添加ListView标签
* 在Activity中编写代码（java）

- 获取ListView对象

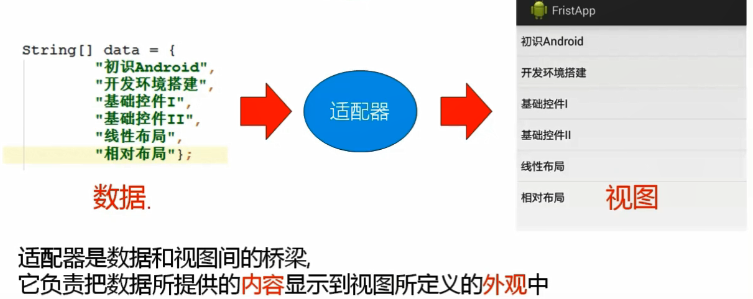
- 准备数据源

- 配置适配器

- 将适配器关联到ListView

### 1.2适配器的使用

适配器负责将数据和视图（ListView的子项）进行绑定。最终再把适配器配置给ListView完成视图展示。



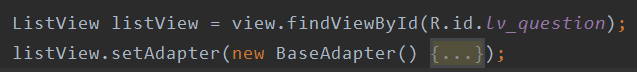
1.2.1适配器的种类

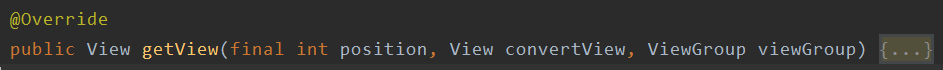


ArrayAdapter、SimpleAdapter、SimpleCursorAdapter都继承于BaseAdapter，是Android内置的适配器。

一般在使用ListView的时候，系统提供的适配器都很难满足需求了，需要自定义适配器。

自定义适配器的核心在于实现BaseAdapter的getView()方法。







getView方法返回一个View作为ListView的子项（item）。

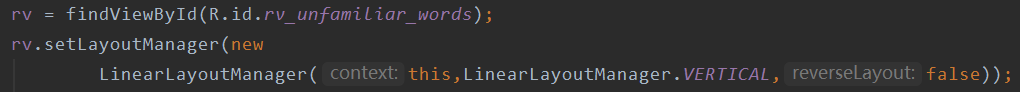
下面展示几个效果图：

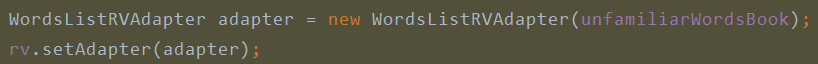
 

图1 图2

## RecycleView的基本使用

2.1简单使用：相较于ListView，RecycleView的使用步骤中多了一步setLayoutManager()。RecyclerView中默认带有三个布局管理器：LinearLayoutManager、GridLayoutManager、StaggeredGridLayoutManager，这三种布局管理器都支持横向和纵向排列以及反向滑动。





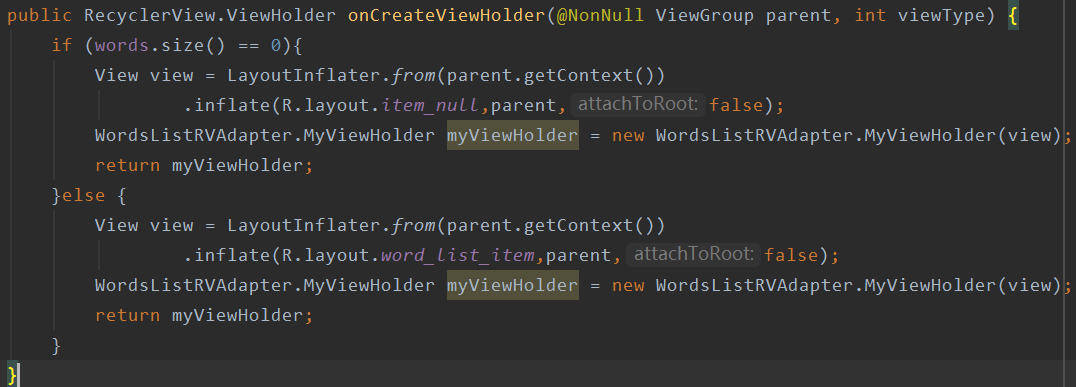
2.2 RecycleView的适配器

自定义适配器，继承RecyleView.Adapter。



构造方法，负责初始化适配器中所需的数据。

onCreateViewHolder，创建ViewHolder，并return。这里可以根据实际情况返回不同的ViewHolder，比如：当适配器构造方法中传入的list长度为0，返回无数据提示界面，不为0，则绑定子视图。如下：



onBindViewHolder，绑定数据，通过参数holder，配置控件属性。



getItemCount，控制RecycleView显示多少个子项。比如上面的情形，显示一个或者多个item。



最终效果图：

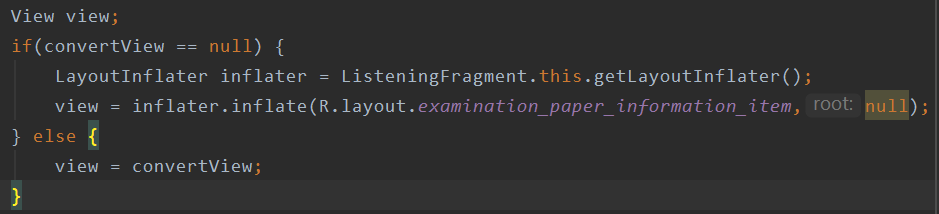
 

## ListView和RecycleView的优化

**3.1 ListView优化：**

布局复用。对于很多item使用相同布局的情况下适用：





复用机制：简单来说就是当一个item完全滑出屏幕外后，这个item的view并没有被销毁，而是存储到了一个mScrapView数组（废弃View）中，当新的View要显示的时候，如果mAcitivityView(屏幕上可见的View)中存在，直接复用；mAcitivityView不存在，mScrapView存在，间接复用，都不存在，inflater。

复用：mAcitivityView存在🡪直接复用

mAcitivityView不存在&& mScrapView存在🡪间接复用

mAcitivityView && mScrapView都不存在🡪inflater view。

**3.2 RecycleView优化**

1、数据处理于视图绑定分离

由于RecycleView的bindViewHolder方法是在UI线程进行的，因此不能再此方法中进行耗时操作，否则会影响滑动流畅性。这个方法中，应该只是负责将数据set到视图中，而不进行业务处理。

2、布局优化，减少布局层级。

3、xml布局换动态布局。

4、设置item的高度固定。

RecycleView.setHasFixedSize(true);避免requestLayout浪费资源。

5、减少itemView的监听器创建。