

- **ViewHolder**是用来保存视图引用的类，无论是**ListView**亦或是**RecyclerView**。只不过在**ListView**中，**ViewHolder**需要自己来定义，且这只是一种推荐的使用方式，不使用当然也可以，这不是必须的。只不过不使用**ViewHolder**的话，**ListView**每次**getView**的时候都会调用**findViewById(int)**，这将导致**ListView**性能展示迟缓。而在**RecyclerView**中使用**RecyclerView.ViewHolder**则变成了必须，尽管实现起来稍显复杂，但它却解决了**ListView**面临的上述不使用自定义**ViewHolder**时所面临的问题。
- 我们知道**ListView**只能在垂直方向上滚动，**Android API**没有提供**ListView**在水平方向上面滚动的支持。或许有多种方式实现水平滑动，但是请相信我，**ListView**并不是设计来做这件事情的。但是**RecyclerView**相较于**ListView**，在滚动上面的功能扩展了许多。它可以支持多种类型列表的展示要求，主要如下：
 1. **LinearLayoutManager**，可以支持水平和竖直方向上滚动的列表。
 2. **StaggeredGridLayoutManager**，可以支持交叉网格风格的列表，类似于瀑布流或者**Pinterest**。
 3. **GridLayoutManager**，支持网格展示，可以水平或者竖直滚动，如展示图片的画廊。
- 列表动画是一个全新的、拥有无限可能的维度。起初的**Android API**中，删除或添加**item**时，**item**是无法产生动画效果的。后面随着**Android**的进化，**Google**的**Chat Hasse**推荐使用**ViewPropertyAnimator**属性动画来实现上述需求。相比较于**ListView**，**RecyclerView.ItemAnimator**则被提供用于在**RecyclerView**添加、删除或移动**item**时处理动画效果。同时，如果你比较懒，不想自定义**ItemAnimator**，你还可以使用**DefaultItemAnimator**。
- **ListView**的**Adapter**中，**getView**是最重要的方法，它将视图跟**position**绑定起来，是所有神奇的事情发生的地方。同时我们也能够通过**registerDataObserver**在**Adapter**中注册一个观察者。**RecyclerView**也有这个特性，**RecyclerView.AdapterDataObserver**就是这个观察者。**ListView**有三个**Adapter**的默认实现，分别是**ArrayAdapter**、**CursorAdapter**和**SimpleCursorAdapter**。然而，**RecyclerView**的**Adapter**则拥有除了内置的内**DB**游标和**ArrayList**的支持之外的所有功能。**RecyclerView.Adapter**的实现的，我们必须采取措施将数据提供给**Adapter**，正如**BaseAdapter**对**ListView**所做的那样。
- 在**ListView**中如果我们想要在**item**之间添加间隔符，我们只需要在布局文件中对**ListView**添加如下属性即可：

```
android:divider="@android:color/transparent" android:dividerHeight="5dp" * ListView通过
```

AdapterView.OnItemClickListener接口来探测点击事件。而**RecyclerView**则通过**RecyclerView.OnItemTouchListener**接口来探测触摸事件。它虽然增加了实现的难度，但是却给予开发人员拦截触摸事件更多的控制权限。* **ListView**可以设置选择模式，并添加**MultiChoiceModeListener**，如下所示：

```
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE_MODAL);
listView.setMultiChoiceModeListener(new MultiChoiceModeListener() {
    public boolean onCreateActionMode(ActionMode mode, Menu menu) { ... }
    public void onItemCheckedStateChanged(ActionMode mode, int position,
        long id, boolean checked) { ... }
        public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
            switch (item.getItemId()) {
                case R.id.menu_item_delete_crime:
                    CrimeAdapter adapter = (CrimeAdapter) getListAdapter();
                    CrimeLab crimeLab = CrimeLab.get(getActivity());
                    for (int i = adapter.getCount() - 1; i >= 0; i--) {
                        if (getListView().isItemChecked(i)) {
                            crimeLab.deleteCrime(adapter.getItem(i));
                        }
                    }
                    mode.finish();
                    adapter.notifyDataSetChanged();
                    return true;
                default:
                    return false;
            }
        }
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) { ... }
    public void onDestroyActionMode(ActionMode mode) { ... }
});
```

而**RecyclerView**则没有此功能。