**RecyclerView**

参考：<http://www.jianshu.com/p/ff6082c0867e?mType=Group>

<https://blog.csdn.net/qq_23012315/article/details/50807224>

<https://zhooker.github.io/2017/08/14/%E5%85%B3%E4%BA%8ERecyclerview%E7%9A%84%E7%BC%93%E5%AD%98%E6%9C%BA%E5%88%B6%E7%9A%84%E7%90%86%E8%A7%A3/>

1. RecyclerView和ListView对比

<https://www.jianshu.com/p/f592f3715ae2>

<https://www.jianshu.com/p/cd8244d1c19a>

<https://zhuanlan.zhihu.com/p/23339185>

<https://blog.csdn.net/harvic880925/article/details/84866486>

1. 缓存机制不同



1. 布局效果
2. Listview只能线性布局；
3. Recyclerview有线性布局、网格布局和瀑布流布局；（Recyclerview通过LayoutManager实现布局）
4. 空数据的处理：
5. ListView通过setEmptyView() 来处理Adapter中数据为空的情况；
6. Recyclerview没有相应的api，需要用户自己处理数据为空的情况。
7. HeaderView与 FooterView的处理
8. ListView中可通过addHeaderView() 与 addFooterView()来添加头部item与底部item，可用来实现下拉刷新或者上拉加载的情况；且这两个API不影响Adapter；
9. RecyclerView中并没有这两个API，所以当需要在RecyclerView添加头部item或者底部item时，可在Adapter中自己编写，根据ViewHolder的Type与View自定义Header，Footter与普通的item，但会影响Adapter的数据，如position，添加了Header与Footter后，实际的position将大于数据的position；
10. 刷新
11. ListView通常用notifyDataSetChanged() 来刷新数据，但其是全局刷新的（每个item的数据都会重新加载一遍），这样就会非常消耗资源； 但ListView依然可以实现局部刷新，当一个item数据刷新时，我们可以在Adapter中，实现一个onItemChanged()方法，在方法里面获取到这个item的position（可以通过getFirstVisiblePosition()），然后调用getView()方法来刷新这个item的数据；
12. RecyclerView中可实现局部刷新，例如：notifyItemChanged()，notifyDataInserted()，notifyItemMoved()
13. 动画效果
14. ListView并没有实现动画效果，但可以在Adapter自己实现item的动画效果；
15. RecyclerView通过相应的接口实现自定义的动画效果（RecyclerView.ItemAnimator类），然后调用RecyclerView.setItemAnimator() (默认的有SimpleItemAnimator与DefaultItemAnimator）；
16. 点击事件
17. 在ListView中有onItemClickListener(), onItemLongClickListener(), onItemSelectedListener(), 但是添加HeaderView与FooterView后就不一样了，因为HeaderView与FooterView都会算进position中，这时会发现position会出现变化，可能会抛出数组越界，为了解决这个问题，我们在getItemId()方法（在该方法中HeaderView与FooterView返回的值是-1）中通过返回id来标志对应的item，而不是通过position来标记；但是我们可以在Adapter中针对每个item写在getView()中会比较合适；
18. RecyclerView中提供了唯一一个API：addOnItemTouchListener()，监听item的触摸事件；我们可以通过RecyclerView的addOnItemTouchListener()加上系统提供的Gesture Detector来实现像ListView那样监听某个item某个操作方法；
19. 嵌套滚动机制
20. 在事件分发机制中，Touch事件在进行分发的时候，由父View向子View传递，一旦子View消费这个事件的话，那么接下来的事件分发的时候，父View将不接受，由子View进行处理；但是与Android的事件分发机制不同，嵌套滚动机制（Nested Scrolling）可以弥补这个不足，能让子View与父View同时处理这个Touch事件，主要实现在于NestedScrollingChild与NestedScrollingParent这两个接口；而在RecyclerView中，实现的是NestedScrollingChild，所以能实现嵌套滚动机制；
21. ListView就没有实现嵌套滚动机制；
22. RecyclerView的使用
23. RecyclerView必须添加布局（RecyclerView.setLayoutManager()）和adaper（RecyclerView.setAdapter()）
24. 获取第一个可见的Item的位置  
    LinearLayoutManager linearLayoutManager = new LinearLayoutManager(mContext);  
    recyclerView.setLayoutManager(layoutManager);  
    layoutManager.findFirstVisibleItemPosition()
25. 获取第position位置的Item  
    View view = layoutManager.getChildAt(position - layoutManager.findFirstVisibleItemPosition());
26. RecyclerView的item焦点争夺问题解决方法：
27. RecyclerView的父view的属性设置为：

android:focusable="true"

android:focusableInTouchMode="true"

参考：<https://blog.csdn.net/android_yyf/article/details/79924669>

1. RecyclerView设置ViewGroup的属性为：

android:descendantFocusability="beforeDescendants"

参考：<https://www.jianshu.com/p/4e73c3129872>

1. RecyclerView.OnScrollListener

SCROLL\_STATE\_DRAGGING： 手指按住屏幕拖动  
SCROLL\_STATE\_SETTLING： 手指快速在屏幕滑一下后的惯性滑动  
SCROLL\_STATE\_IDLE： 屏幕处于静止状态