1. build.gradle中引入

implementation **'com.android.support:recyclerview-v7:26+'**

1. 类似控件 ListView、GridView
2. 区别与ListView, GridView

整体上看RecyclerView架构，提供了一种插拔式的体验，高度的解耦，异常的灵活，通过设置它提供的不同LayoutManager，ItemDecoration , ItemAnimator实现令人瞠目的效果。

--你想要控制其显示的方式，请通过布局管理器LayoutManager

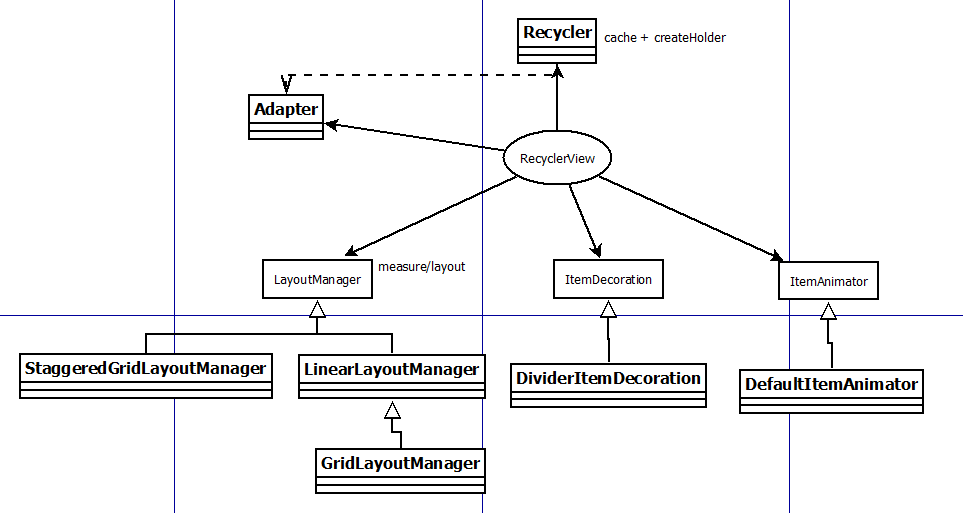
--你想要控制Item间的间隔（可绘制），请通过ItemDecoration

--你想要控制Item增删的动画，请通过ItemAnimator

--你想要控制点击、长按事件，请自己写（擦，这点尼玛。）

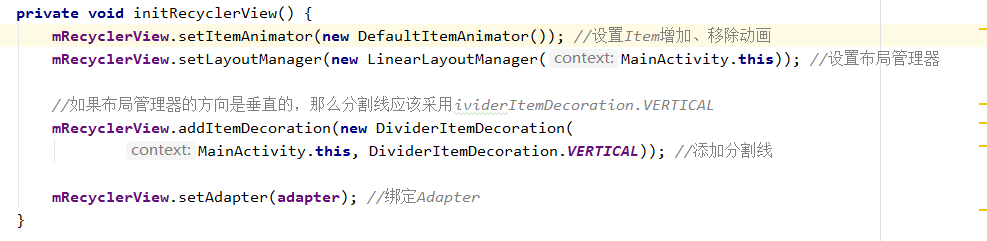
比起ListView, GridView的Adapter来说， RecyclerView多了LayoutManager， ItemDecoration， ItemAnimator； 另外比起ListView, GridView来说，RecyclerView没有设置点击事件，长按事件的API方法.

1. RecyclerView架构



5. 使用方式

RecyclerView只管回收与复用View，显示效果定制自由



6. RecyclerView.Adapter介绍

Adapter的作用就是为RecyclerView进行数据与Item View的绑定。

public abstract static class Adapter<VH extends ViewHolder>

此处泛型VH继承自ViewHolder； 创建Adapter如果不指定泛型，默认ViewHolder；

那么重写Adapter的方法的时候VH就是ViewHolder类型：

public abstract VH onCreateViewHolder(ViewGroup parent, int viewType); // VH == ViewHolder

public abstract void onBindViewHolder(VH holder, int position); // VH == ViewHolder

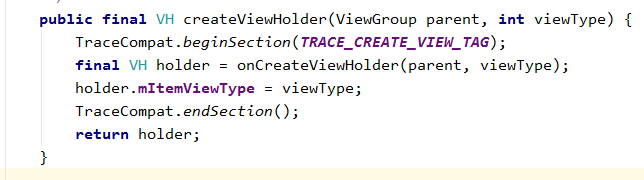
可以在重写Adapter的时候制定泛型类，该泛型类继承自ViewHolder，则

public abstract VH onCreateViewHolder(ViewGroup parent, int viewType); // VH == ViewHolder子类

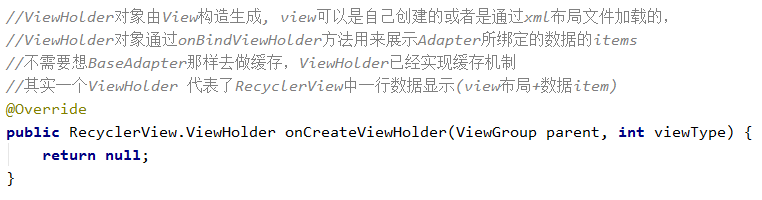
public abstract void onBindViewHolder(VH holder, int position); // VH == ViewHolder子类

RecyclerView.Adapter相关方法如下:

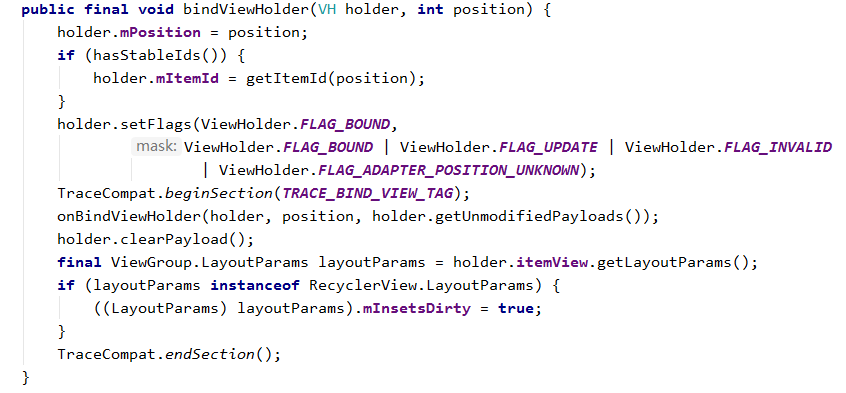
1).不可重写



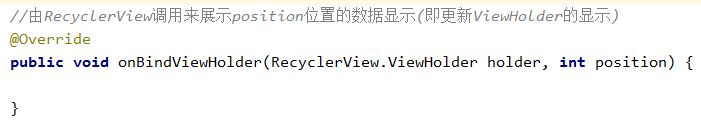
2).需要重写



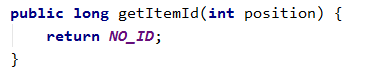
3).不可重写

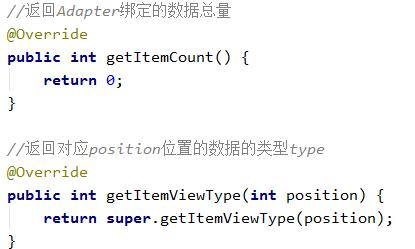


4).需重写

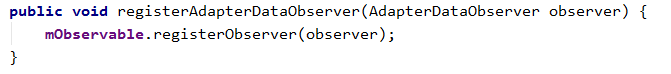


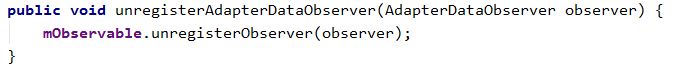
5).其他重要函数

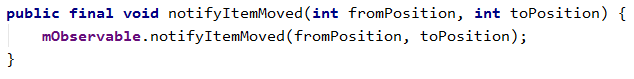




C:\Users\DELL\AppData\Local\Temp\1528440794(1).png



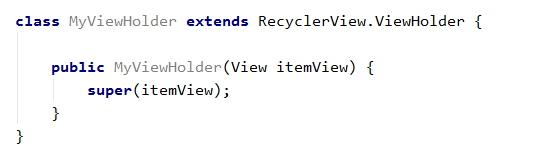




等等…

7. RecyclerView.ViewHolder 解释

ViewHolder本身是抽象类，意味着不能通过new实例化，因此需要创建ViewHolder的子类。



或者在onCreateViewHolder中通过

RecyclerView.ViewHolder holder = new RecyclerView.ViewHolder(view){}; 方式创建对象

8. ItemDecoration 分割线装饰

public abstract static class ItemDecoration {

public void onDraw(Canvas c, RecyclerView parent, State state) {

onDraw(c, parent);

}

public void onDrawOver(Canvas c, RecyclerView parent, State state) {

onDrawOver(c, parent);

}

public void getItemOffsets(Rect outRect, View view, RecyclerView parent, State state) {

getItemOffsets(outRect, ((LayoutParams)view.getLayoutParams()).getViewLayoutPosition(),

parent);

}

}

在绘制RecyclerView的Items的过程中，如果通过addItemDecoration()方法添加decoration； 那么

(1) ItemDecoration的onDraw方法先于drawChildren

(2) ItemDecoration的onDrawOver在drawChildren之后，一般我们选择复写其中一个即可。

(3) getItemOffsets 可以通过outRect.set()为每个Item设置一定的偏移量，主要用于绘制Decorator。

Android提供了一些默认实现类，比如DividerItemDecoration(针对水平防线以及垂直方式做了适配)

getItemOffsets() : 设置ItemView的内嵌偏移长度（inset）

其实RecyclerView 中的 ItemView 外面会包裹着一个矩形（outRect）

内嵌偏移长度 是指：该矩形（outRect）与 ItemView的间隔



那么设置了outRect的left,top,right,bottom是将原有的itemview向内压缩一定距离。

通过这个方法设置的rect区域主要用来绘制decoration -------限制区域.

如果outRect.set(0, 0, 0, 0); 那么绘制的decoration 有可能会影响周边的view.

9. LayoutManager 抽象类

Android提供了3个默认实现：

1). LinearLayoutManager(线性，支持横向、纵向);

2). GridLayoutManager(网格布局管理器)**;**

3). StaggeredGridLayoutManager(瀑布式布局管理器) :既可以水平方向又可以垂直方向

mRecyclerView.setLayoutManager(new StaggeredGridLayoutManager(4, StaggeredGridLayoutManager.HORIZONTAL));

10. 瀑布流：StaggeredGridLayoutManager

item高度随机设置

int width = ((Activity) holder.tv.getContext()).getWindowManager().getDefaultDisplay().getWidth();

//高度随机----针对瀑布流

ViewGroup.LayoutParams params = holder.tv.getLayoutParams();

//设置图片的相对于屏幕的宽高比

params.width = width/RAW;

params.height = (int) (200 + Math.random() \* 200) ;

holder.tv.setLayoutParams(params);

11. ItemAnimator

当Adapter中的item发生变化时，该动画就会触发。

该抽象类可以被继承实现来自定义动画效果。

动画发生时RecyclerView负责管理维护items;

itemAnimator.dispatchAnimationFinished(ViewHolder)

recyclerview.animateAppearance()🡪itemAnimator.animateAppearance(ViewHolder, ItemHolderInfo, ItemHolderInfo)

recyclerview.animateChange()🡪itemAnimator.animateChange(ViewHolder, ViewHolder, ItemHolderInfo, ItemHolderInfo)

itemAnimator.animatePersistence(ViewHolder, ItemHolderInfo, ItemHolderInfo) //

recyclerview.animateDisappearance()🡪itemAnimator.animateDisappearance(ViewHolder, ItemHolderInfo, ItemHolderInfo)

系统默认实现类 DefaultItemAnimator

animateRemove() 动画结束后，子类必须调用dispatchRemoveFinished (ViewHolder); disappear item也会调用该方法 --- onRemoveFinished()

animateAdd() ---- dispatchAddFinished() -- onAddFinished()

animateMove() --- dispatchMoveFinished(ViewHolder) --- onMoveFinished()

animateChange() --- dispatchChangeFinished(ViewHolder, boolean) Adapter对象的notifyItemChanged(),notifyItemRangeChanged()会触发该方法 -- onChangeFinished()

dispatchRemoveStarting() --- onRemoveStarting()

dispatchAddStarting() --- onAddStarting()

dispatchMoveStarting() --- onMoveStarting()

dispatchChangeStarting() --- onChangeStarting()

runPendingAnimations()

第三方提供的item增删动画

<https://github.com/gabrielemariotti/RecyclerViewItemAnimators>

12.响应item的单击，长按等事件

RecyclerView本身没有提供对应的接口.

可以通过RecyclerView.Adapter实现类对外提供接口，外部传入接口类，RecyclerView.Adapter内部对item进行单击、长按处理的时候，通过回调通知外部。



13. ListView RecyclerView在实际使用中几乎都会与Imageview打交道，都会涉及网络图片的加载显示。

这里就涉及网络图片的三级缓存:

1). 内存缓存: 核心LruCache

2). 磁盘缓存:

3). 网络加载: 比如通过AsyncTask +HttpURLConnection进行下载显示

或者直接使用第三方开源框架,如Picasso、Glide、Fresco等.

14.问题1： RecyclerView 多条数据却只显示一条数据

解决方案：

（1）检查getItemCount()返回值

（2）检查 onBindViewHolder()执行次数情况

我个人情况是getItemCount() >1; onBindViewHolder()却只执行一次；

检查layout item的布局，发现item根布局的宽高属性都是match\_parent; 改成wrap\_content解决。

如果是垂直布局，把item的高改成wrap\_content就可以了。