

8.RecycleView

8.1.RecyclerView的多级缓存机制,每一级缓存具体作用是什么,分别在什么场景下会用到哪些缓存

<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/>

<https://www.wanandroid.com/wenda/show/14222>

<https://blog.csdn.net/u013700502/article/details/105058771>

<https://juejin.cn/post/6854573221702795277#heading-9>

Scrap、Cache、 ViewCacheExtension 、 RecycledViewPool

Scrap缓存用在RecyclerView布局时，布局完成之后就会清空

添加到Cache缓存和RecyclerViewPool缓存的item，他们的View必须已经从RecyclerView中detached或removed

一级缓存：mAttachedScrap 和 mChangedScrap 二级缓存：mCachedViews 三级缓存：ViewCacheExtension
四级缓存：RecycledViewPool 然后说怎么用，就是先从1级找，然后2级...然后4级，找不到 create ViewHolder。

<https://www.jianshu.com/p/467ae8a7ca6e>

mAttachedScrap/mChangedScrap

屏幕内缓存

RecyclerView 的滑动场景来说，新卡位的复用以及旧卡位的回收机制，不会涉及到 mChangedScrap 和 mAttachedScrap

notifyItemChanged/rangeChange，此时如果Holder发生了改变那么就放入changeScrap中，反之放入到 AttachScrap。

mCachedViews

当列表滑动出了屏幕时，ViewHolder会被缓存在 mCachedViews，其大小由mViewCacheMax决定，默认 DEFAULT_CACHE_SIZE为2，可通过RecyclerView.setItemViewCacheSize()动态设置。

ViewCacheExtension

可以自己实现ViewCacheExtension类实现自定义缓存，可通过RecyclerView.setViewCacheExtension()设置。

缓存池

ViewHolder在首先会缓存在 mCachedViews 中，当超过了个数（比如默认为2），就会添加到 RecycledViewPool 中。RecycledViewPool 会根据每个ViewType把ViewHolder分别存储在不同的列表中，每个 ViewType最多缓存DEFAULT_MAX_SCRAP = 5 个ViewHolder

8.2.RecyclerView的滑动回收复用机制

<https://www.jianshu.com/p/467ae8a7ca6e>

RecyclerView 滑动的场景触发的回收复用机制工作时，并不需要四级缓存都参与的。

1.RecyclerView 向下滑动操作的日志，第三行5个卡位的显示都是重新创建的 ViewHolder

新一行5个卡位和复用不可能用到刚移出屏幕的5个卡位，因为先复用再回收，新一行的5个卡位先去目前的 mCachedViews 和 ViewPool 的缓存中寻找复用，没有就重新创建，然后移出屏幕的那行的5个卡位再回收缓存到 mCachedViews 和 ViewPool 里面，

2.RecyclerView 再次向上滑动重新显示第一行的5个卡位时，只有后面3个卡位触发了 onBindViewHolder() 方法，重新绑定数据。

滑动场景下涉及到的回收和复用的结构体是 mCachedViews 和 ViewPool，前者默认大小为2，后者为5。所以，当第三行显示出来后，第一行的5个卡位被回收，回收时先缓存在 mCachedViews，满了再移出旧的到 ViewPool 里，所有5个卡位有2个缓存在 mCachedViews 里，3个缓存在 ViewPool，所以最新的两个卡位是0、1，会放在 mCachedViews 里，而2、3、4的卡位则放在 ViewPool 里。

3.而至于为什么会创建了17个 ViewHolder，那是因为再第四行的卡位要显示出来时，ViewPool 里只有3个缓存，而第四行的卡位又用不了 mCachedViews 里的2个缓存，因为这两个缓存的是6、7卡位的 ViewHolder，所以需要再重新创建2个 ViewHodler 来给第四行最后的两个卡位使用。

8.3.RecyclerView的刷新回收复用机制

notifyXxx后会RecyclerView会进行两次布局，一次预布局，一次实际布局，然后执行动画操作

dispatchLayoutStep1

查找改变holder，并保存在mChangedScrap中；其他未改变的保存到mAttachedScrap中（mChangedScrap 保存的holder信息只有预布局时才会被复用）

dispatchLayoutStep2

此步骤会创建一个新的holder并执行绑定数据，充当改变位置的holder，其他位置holder从mAttachedScrap中获取

8.4.RecyclerView 为什么要预布局

<https://juejin.cn/post/6890288761783975950#heading-0>

why

这种负责执行动画的View在原布局或新布局中不存在的动画，就是预测动画。

因为RecyclerView 要执行预测动画。比如有A,B,C三个itemView，其中A和B被加载到屏幕上，这时候删除B后，按照最终效果我们会看到C移动到B的位置；因为我们只知道 C 最终的位置，但是不知道 C 的起始位置在哪里(即C还未被加载)。

用户有 A、B、C 三个 item，A，B 刚好显示在屏幕中，这个时候，用户把 B 删除了，那么最终 C 会显示在 B 原来的位置

因为我们只知道 C 最终的位置，但是不知道 C 的起始位置在哪里，无法确定 C 应该从哪里滑动过来。

在其他 LayoutManager 中，它可能是从侧面或者是其他地方滑动过来的。

what

当 Adapter 发生变化的时候，RecyclerView 会让 LayoutManager 进行两次布局。

第一次，预布局,为动画前的表项先执行一次pre-layout，根据 Adapter 的 notify 信息，我们知道哪些 item 即将变化了,将不可见的表项 3 也加载到布局中，形成一张布局快照（1、2、3）。

第二次，实际布局，也就是变化完成之后的布局同样形成一张布局快照（1、3）。

这样只要比较前后布局的变化，就能得出应该执行什么动画了，就称为预测动画。

8.5.ListView 与 RecyclerView区别

1.布局效果

ListView 的布局比较单一，只有一个纵向效果；RecyclerView 的布局效果丰富，可以在 LayoutManager 中设置：线性布局（纵向，横向），表格布局，瀑布流布局

2.局部刷新

RecyclerView中可以实现局部刷新，例如：notifyItemChanged()；

如果要在ListView实现局部刷新，依然是可以实现的，当一个item数据刷新时，我们可以在Adapter中，实现一个notifyItemChanged()方法，在方法里面通过这个 item 的 position，刷新这个item的数据

3.缓存区别

ListView有两级缓存，在屏幕与非屏幕内。RecyclerView比ListView多两级缓存 ListView缓存View。RecyclerView缓存RecyclerView.ViewHolder

8.6.RecyclerView性能优化

1.数据处理与视图加载分离

简单来说就是在onBindViewHolder()只设置UI显示，不做任何逻辑判断，需要的业务逻辑在得到javabean之前处理好，

2.布局优化

减少过度绘制 减少布局层级

3.设置RecyclerView.addOnScrollListener()来在滑动过程中停止加载的操作。
