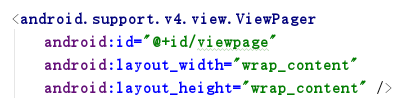
**ViewPage**

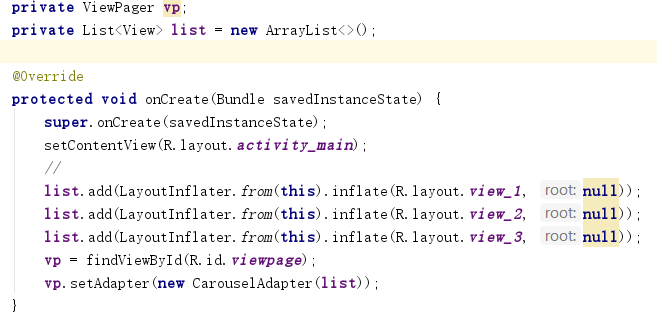
参考：<https://blog.csdn.net/harvic880925/article/details/38453725>

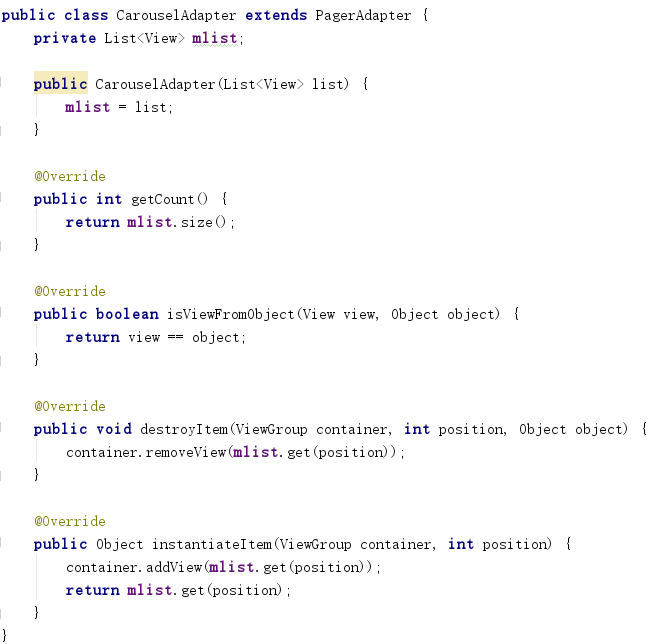
ViewPage的功能是使一组视图在屏幕上可以滑动的展示。ViewPag也需要PageAdapter的配合才能使用。

**基础使用**

布局文件如下，因为ViewPage来自于支持库，所以添加到布局时要用全称：







ViewPager默认加载当前屏幕上的列表项，以及左右相 邻页面的数据，因此响应迅速。如有需要，也可调用setOffscreenPageLimit(int)方法，定制预加载相邻页面的数目。

**适配器**

* **PageAdapter**

使用PageAdapter必须重写，getCount，instantiateItem，destroyItem，isViewFromObject四个函数：

**Int getCount()：**

返回当前有效视图的个数。

**Object instantiateItem(ViewGroup container, int position)：**

PagerAdapter创建指定位置的列表项视图（并不是立即创建视图），然后将其添加给ViewGroup视图容器。这里返回的对象一般是这个position位置上的View，即刚刚创建的视图。

**void destroyItem(ViewGroup container, int position, Object object):**

销毁ViewGroup中指定位置的视图。

**abstract boolean isViewFromObject(View view, Object object):**

其中object参数是instantiateItem方法返回的对象，这里需要判断view这个对象，与instaantiateItem方法创建的view对象是否是同一个对象，如果是则返回true，否则返回false。

根据官方文档解释，AdapterView通过一系列函数调用来完成一个更新的步骤，AdapterView不会管理每一个视图，而是将每一个视图与一个键值key联系起来，这个键值key代表唯一一个页面。这个key指的就是instantiateItem方法返回的对象。也就是说instantiateItem方法会向ViewGroup中添加一个View，然后创建一个与这个View唯一对应的key，这个key可以是任意值，可以是你在这个方法中添加的View对象，也可以是这个View对象的position，然后将添加的在这个View和这个Key传递给isViewFromObject方法中，在isViewFromObject方法中再次判断这个key是否是这个View唯一对应的那个key，如果是则返回true，否则返回false。在上面的例子中，直接将添加的这个view作为唯一对应的key，所以在isViewFromObject方法中直接做了view==object的对比即可。

**void startUpdate(ViewGroup container)**

开始调用显示页面

**void finishUpdate(ViewGroup container)：**

当显示界面加载完成时调用该方法

**int [getItemPosition](http://developer.android.com/reference/android/support/v4/view/PagerAdapter.html" \l "getItemPosition(java.lang.Object))([Object](http://developer.android.com/reference/java/lang/Object.html) object)**

如果item位置没有发生改变则返回POSITION\_UNCHANGED如果发生了改变则返回POSITION\_NONE

**CharSequence getPageTitle(int position)**

设置当前位置上的ViewPager的标题

**float getPageWidth(int position)**

返回position位置测量的页面的宽高比例范围是(0.f-1.f]

**void notifyDataSetChanged()**

调用该方法更新数据。pageradapter支持数据集合的改变，数据集合的改变必须要在主线程里面执行，然后还要调用notifyDataSetChanged方法，数据集合的改变包括页面的添加删除和修改位置。

**void registerDataSetObserver(DataSetObserver observer)**

注册观察者

**void unregisterDataSetObserver(DataSetObserver observer)**

在回调函数中通过adapter的数据变化注销观察者对象

**void restoreState(Parcelable state, ClassLoader loader)**

恢复与adapter关联的所有页面的状态实例，并且该页面是通过saveState()方法保存的。

**parcelable saveState()**

保存于adapter关联的所有的页面，直到调用restoreState方法时被恢复

**void setPrimaryItem(ViewGroup container, int position, Object object)**

Called to inform the adapter of which item is currently considered to be the "primary", that is the one show to the user as the current page.

* **FragmentStatePagerAdapter**

用到的时候在分析

* **FragmentPagerAdapter**

用到的时候在分析

**其它**

* **切换效果**

参考：https://blog.csdn.net/u012702547/article/details/52334161