Demo57-简单的自定义View（1）

易：extends xxxView，重写某个系统监听回调API

NoScrollViewPager.java

**import** android.content.Context;  
**import** android.support.v4.view.ViewPager;  
**import** android.util.AttributeSet;  
**import** android.view.MotionEvent;  
  
*/\*\*  
 \* Created by Administrator on 2018/8/12.  
 \* 自定义不可滑动的ViewPager  
 \*/***public class** NoScrollViewPager **extends** ViewPager {  
 */\*\*  
 \* 在代码中实例化的时候调用改方法  
 \** ***@param context*** *\*/* **public** NoScrollViewPager(Context context) {  
 **super**(context);  
 }  
  
 */\*\*  
 \* 在布局文件中使用该类的时候，要实例化该类（因为不是static），用该构造方法  
 \* 该构造方法不能少，少的话就会奔溃\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* 这是系统的规定！！！！非常重要！！！  
 \** ***@param context*** *\** ***@param attrs*** *\*/* **public** NoScrollViewPager(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 }  
  
  
  
  
 */\*\*  
 \* 重写触摸事件，消费掉  
 \* 这是核心代码！！！  
 \* ViewPager的父类是ViewGroup,自己在onTouchEvent retrun true，就会消费掉触摸事件  
 \* 消费后，就不能传递到view,我们在view层就不可以滑动了  
 \** ***@param ev*** *\** ***@return*** *\*/* @Override  
 **public boolean** onTouchEvent(MotionEvent ev) {  
 **return true**;  
 }  
  
  
 @Override  
 **public boolean** onInterceptTouchEvent(MotionEvent ev) {  
 **return false**;  
 }  
  
  
}

HorizonTalScrollViewPager .java

**import** android.content.Context;  
**import** android.support.v4.view.ViewPager;  
**import** android.util.AttributeSet;  
**import** android.view.MotionEvent;  
  
*/\*\*  
 \* Created by Administrator on 2018/8/18.  
 \*/***public class** HorizonTalScrollViewPager **extends** ViewPager {  
 **public** HorizonTalScrollViewPager(Context context) {  
 **super**(context);  
 }  
  
 **public** HorizonTalScrollViewPager(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 }  
  
  
 *//起始坐标* **private float startX**;  
 **private float startY**;  
  
  
  
 @Override  
 **public boolean** dispatchTouchEvent(MotionEvent ev) {  
 *//请求父层视图不用拦截当前控件的事件,即滑动权给当前控件  
 //getParent().requestDisallowInterceptTouchEvent(true);* **switch** (ev.getAction()){  
  
 **case** MotionEvent.***ACTION\_DOWN***:  
 **startX**=ev.getX();  
 **startY**=ev.getY();  
 **break**;  
 **case** MotionEvent.***ACTION\_MOVE***:  
 *//新的坐标* **float** endX=ev.getX();  
 **float** endY=ev.getY();  
  
 *//计算偏移量* **float** distanceX=endX-**startX**;  
 **float** distanceY=endY-**startY**;  
  
 *//判断滑动方向 ,水平滑动,true的滑动权就给子控件，false就给父类滑动* **if** (Math.*abs*(distanceX)>Math.*abs*(distanceY)){  
 *//第一个页面* **if** (getCurrentItem()==0&&distanceX>0){  
 getParent().requestDisallowInterceptTouchEvent(**false**);  
 }  
 **else if** (getCurrentItem()==getAdapter().getCount()-1&&distanceX<0){  
 getParent().requestDisallowInterceptTouchEvent(**false**);  
 } **else** { getParent().requestDisallowInterceptTouchEvent(**true**);}  
 }  
 *//上下滑动* **else** {  
 getParent().requestDisallowInterceptTouchEvent(**false**);  
 }  
 **break**;  
 **case** MotionEvent.***ACTION\_UP***:  
 **break**;  
 **default**:**break**;  
  
 }  
 **return super**.dispatchTouchEvent(ev);  
 }  
}