Demo57-简单的自定义View（2）

中：在构造方法里初始化view对象，然后addView对象，构造成一个新的View，同时还可以重写系统回调的API

RefreshListView.java

**import** android.content.Context;  
**import** android.media.MediaPlayer;  
**import** android.os.SystemClock;  
**import** android.util.AttributeSet;  
**import** android.util.Log;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.view.animation.Animation;  
**import** android.view.animation.RotateAnimation;  
**import** android.widget.AbsListView;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.ListView;  
**import** android.widget.ProgressBar;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.example.newbeijing.R;  
**import** com.example.newbeijing.utils.MiToast;  
  
**public class** RefreshListView **extends** ListView {  
  
 **private int headerViewHeight**;  
 **private** LinearLayout **headerView**;  
 **private** View **ll\_pull\_down\_refresh**;  
 **private** ImageView **iv\_arrow**;  
 **private** ProgressBar **pb\_status**;  
 **private** TextView **tv\_status**;  
 **private** TextView **tv\_time**;  
 **private int headHeight**;

*/\*\*  
 \* 下拉刷新  
 \*/* **public static final int *PULL\_DOWN\_REFRESH*** = 0;  
  
 */\*\*  
 \* 手松刷新  
 \*/* **public static final int *RELEASE\_REFRESH*** = 1;  
  
  
 */\*\*  
 \* 正在刷新  
 \*/* **public static final int *REFRESHING*** = 2;  
  
  
 */\*\*  
 \* 当前状态  
 \*/* **private int currentStatus** = ***PULL\_DOWN\_REFRESH***;  
  
 **private** Animation **upAnimation**;  
 **private** Animation **downAnimation**;  
 **private** View **footerView**;  
 **private int footerViewHeight**;  
 **private boolean isLoadMore** = **false**;  
 **private** View **topview**;  
 **private int listViewOnScreenY**=-1;  
  
  
 **public** RefreshListView(Context context, AttributeSet attrs, **int** defStyle) {  
 **super**(context, attrs, defStyle);  
 initHeaderView(context);  
 initFooterView(context);  
  
 initAnimation();  
 }  
  
  
 **public** RefreshListView(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 initHeaderView(context);  
 initFooterView(context);  
  
  
 initAnimation();  
 }  
  
 **public** RefreshListView(Context context) {  
 **super**(context);  
 initHeaderView(context);  
 initFooterView(context);  
  
 initAnimation(); *//初始化两个动画，前面奔溃就是因为没初始化view* }  
  
 */\*\*  
 \*    \* 本次自定义view的唯一目的，就是为了加一个新的view  
 \*    \* @param context  
 \*     
 \*/* **public void** initHeaderView(Context context) {  
  
 **headerView** = (LinearLayout) View.*inflate*(context, R.layout.***refresh\_header***, **null**);  
  
 *//下拉刷新控件* **ll\_pull\_down\_refresh** = **headerView**.findViewById(R.id.***ll\_pull\_down\_refresh***);  
 **iv\_arrow** = (ImageView) **headerView**.findViewById(R.id.***iv\_arrow***);  
 **pb\_status** = (ProgressBar) **headerView**.findViewById(R.id.***pb\_status***);  
 **tv\_status** = (TextView) **headerView**.findViewById(R.id.***tv\_status***);  
 **tv\_time** = (TextView) **headerView**.findViewById(R.id.***tv\_time***);  
 *//tv\_more=(TextView) headerView.findViewById(R.id.tv\_more);  
  
  
 //先测量才能得到高，参数值随便设* **ll\_pull\_down\_refresh**.measure(0, 0);  
 *//得到高* **headHeight** = **ll\_pull\_down\_refresh**.getMeasuredHeight();  
 *//默认隐藏  
 //ll\_pull\_down\_refresh.setPadding(0,-控件高,0,0);就会完全隐藏，0就是全显，height就是两倍高* **ll\_pull\_down\_refresh**.setPadding(0, -**headHeight**, 0, 0);  
  
  
 *//添加到内部listView的头部，才能显示* addHeaderView(**headerView**);  
  
  
  
 }  
  
  
 *//加载更多的vie加在尾巴* **private void** initFooterView(Context context) {  
  
 **footerView** = View.*inflate*(context, R.layout.***refresh\_footer***, **null**);  
 **footerView**.measure(0, 0);  
 **footerViewHeight** = **footerView**.getMeasuredHeight();  
 **footerView**.setPadding(0, -**footerViewHeight**, 0, 0);  
  
 *//本类是extends ListView的，所以this代表的就是一个listView，可以省略，listView添加尾巴View* **this**.addFooterView(**footerView**);  
  
 *//可以理解为控件本身监听自己  
 //监听ListView的滚动,这是加载更多布局的触发条件* **this**.setOnScrollListener(**new** MyOnScrollListener());  
 }  
  
  
 **private float startY** = -1;  
  
 @Override  
 **public boolean** onTouchEvent(MotionEvent ev) {  
  
 **switch** (ev.getAction()) {  
  
 **case** MotionEvent.***ACTION\_DOWN***:  
 *//1、记录起始坐标* **startY** = ev.getY();  
 **break**;  
  
  
 **case** MotionEvent.***ACTION\_MOVE***:  
 **if** (**startY** == -1) {  
 **startY** = ev.getY();  
 }  
  
  
  
  
 *//判断顶部轮播图是否显示，只有完全显示，才会有下拉刷新  
 //解决瞬间跳转的Bug* **boolean** isDisplayTopNews=isDisplayTopNews();  
 **if**(!isDisplayTopNews){  
 *//就是不完全显示，即加载更多的情景，就不需要重新下拉刷新，一个break，程序在这里出去，此方法下面代码不执行* **break**;  
 }  
  
  
  
  
  
  
 *//如果正在刷新，就没必要继续刷新* **if** (**currentStatus** == ***REFRESHING***) {  
 **break**;  
 }  
  
  
 *//2、新的坐标* **float** endY = ev.getY();  
 **float** distance = endY - **startY**;  
 **if** (distance > 0) {  
 *//下拉* **int** paddingTop = (**int**) (-**headHeight** + distance);  
  
  
 **if** (paddingTop < 0 && **currentStatus** != ***PULL\_DOWN\_REFRESH***) {  
 *//下拉刷新状态* **currentStatus** = ***PULL\_DOWN\_REFRESH***;  
 *//更新状态* refreshViewState();  
  
 } **else if** (paddingTop > 0 && **currentStatus** != ***RELEASE\_REFRESH***) {  
 *//手松刷新状态* **currentStatus** = ***RELEASE\_REFRESH***;  
 *//更新状态* refreshViewState();  
  
 }  
  
  
 **ll\_pull\_down\_refresh**.setPadding(0, paddingTop, 0, 0);  
 *// GET="yes";* }  
 **break**;  
  
 **case** MotionEvent.***ACTION\_UP***:  
 **startY** = -1;  
 *//手松离开后的操作* **if** (**currentStatus** == ***PULL\_DOWN\_REFRESH***) {  
*// View.setPadding(0,-控件高，0,0);//完全隐藏* **ll\_pull\_down\_refresh**.setPadding(0, -**headHeight**, 0, 0);  
 } **else if** (**currentStatus** == ***RELEASE\_REFRESH***) {  
 *//设置状态为正在刷新* **currentStatus** = ***REFRESHING***;  
  
 refreshViewState();  
  
*// View.setPadding(0,0，0,0);//完全显示* **ll\_pull\_down\_refresh**.setPadding(0, 0, 0, 0);  
 *//回调接口，判空是为了怕用户没设置下拉刷新，一null，一调用就奔溃* **if** (**mOnRefreshListener** != **null**) {  
 **mOnRefreshListener**.onPullDownRefresh();  
 }  
 }  
 **break**;  
  
  
 }  
  
 **return super**.onTouchEvent(ev);  
 }  
  
  
  
 *//判断是否完全显示顶部轮播图  
 //ListView的Y坐标小于等于轮播图的Y坐标，顶部轮播图完全显示* **private boolean** isDisplayTopNews() {  
  
 **if** (**topview**!=**null**){  
 *//得到ListView的坐标* **int**[] location=**new int**[2];  
 **if** (**listViewOnScreenY**==-1){  
 **this**.getLocationOnScreen(location);  
 **listViewOnScreenY**=location[1];  
 }  
 *//得到轮播图的坐标* **topview**.getLocationOnScreen(location);*//因为location是临时量，我们直接覆盖赋值，循环利用* **int** topviewOnScreenY=location[1];  
  
 **return listViewOnScreenY**<=topviewOnScreenY;  
 }**else** {  
 **return true**;  
 }  
  
 }  
  
  
  
  
 **private void** refreshViewState() {  
  
 **switch** (**currentStatus**) {  
 **case *PULL\_DOWN\_REFRESH***: *//下拉刷新状态* **iv\_arrow**.startAnimation(**downAnimation**);  
 **tv\_status**.setText(**"下拉刷新..."**);  
 **break**;  
  
 **case *RELEASE\_REFRESH***: *//手松刷新状态* **iv\_arrow**.startAnimation(**upAnimation**);  
 **tv\_status**.setText(**"手松刷新..."**);  
 **break**;  
  
 **case *REFRESHING***: *//正在刷新状态* **iv\_arrow**.clearAnimation();  
 **iv\_arrow**.setVisibility(***GONE***);  
 **tv\_status**.setText(**"正在刷新..."**);  
 **pb\_status**.setVisibility(***VISIBLE***);  
 **break**;  
 }  
 }  
  
  
 **private void** initAnimation() {  
  
 **upAnimation** = **new** RotateAnimation(0, -180, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 **upAnimation**.setDuration(500);  
 **upAnimation**.setFillAfter(**true**);  
  
 **downAnimation** = **new** RotateAnimation(-180, -360, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 **downAnimation**.setDuration(500);  
 **downAnimation**.setFillAfter(**true**);  
  
 }  
  
  
 **public void** onRefreshFinish(**boolean** success) {  
  
*/\*  
 //下拉刷新  
  
 if (success) {  
 tv\_status.setText("正在刷新...");  
 currentStatus = PULL\_DOWN\_REFRESH;  
 pb\_status.setVisibility(GONE);  
 iv\_arrow.clearAnimation();  
 iv\_arrow.setVisibility(VISIBLE);  
 //隐藏下拉刷新控件  
 ll\_pull\_down\_refresh.setPadding(0, -headHeight, 0, 0);  
 // Toast.makeText(getContext(),"下拉刷新请求网络成功",Toast.LENGTH\_SHORT).show();  
 // MiToast.showShort(getContext(),"刷新成功");  
 } else {  
  
 tv\_status.setText("正在刷新...");  
 currentStatus = PULL\_DOWN\_REFRESH;  
 pb\_status.setVisibility(GONE);  
 iv\_arrow.clearAnimation();  
 iv\_arrow.setVisibility(VISIBLE);  
 //隐藏下拉刷新控件  
 //Toast.makeText(getContext(),"网络不太给力哦",Toast.LENGTH\_SHORT).show();  
 MiToast.showShort(getContext(), "网络不太给力哦");  
 ll\_pull\_down\_refresh.setPadding(0, -headHeight, 0, 0);  
  
 }\*/  
  
  
  
  
 //加载更多* **if** (**isLoadMore**){  
 **isLoadMore**=**false**;*//切换回默认状态  
  
 //隐藏加载的布局* **footerView**.setPadding(0,-**footerViewHeight**,0,0);  
  
 }  
 *//下拉刷新* **else** {  
 **tv\_status**.setText(**"正在刷新..."**);  
 **currentStatus**=***PULL\_DOWN\_REFRESH***;  
 **pb\_status**.setVisibility(***GONE***);  
 **iv\_arrow**.clearAnimation();  
 **iv\_arrow**.setVisibility(***VISIBLE***);  
 *//隐藏下拉刷新控件* **ll\_pull\_down\_refresh**.setPadding(0,-**headHeight**,0,0);  
  
 **if** (success==**true**){  
 Log.*d*(**"yes"**,**"yes"**);  
 }**else** {  
 *//Toast.makeText(getContext(),"网络不给力",Toast.LENGTH\_SHORT).show();* }}  
  
  
 }  
  
  
 */\*\*  
 \* 接口的使用原理，学懂了就会创建接口了，非常重要  
 \* 这是刷新的接口，主要来处理刷新时处理数据的动作  
 \*/* **public interface** OnRefreshListener {  
  
 */\*\*  
 \* 当下拉刷新的时候回调这个方法  
 \*/* **public void** onPullDownRefresh();  
  
 */\*\*  
 \* 当加载更多布局的时候回调此方法  
 \*/* **public void** onLoadMore();  
 }  
  
 **private** OnRefreshListener **mOnRefreshListener**;  
  
  
 */\*\*  
 \* 设置监听刷新，由外界来设置  
 \*/* **public void** setOnRefreshListener(OnRefreshListener l) {  
 **this**.**mOnRefreshListener** = l;  
  
 }  
  
  
 **class** MyOnScrollListener **implements** OnScrollListener {  
  
 @Override  
 **public void** onScrollStateChanged(AbsListView view, **int** scrollState) {  
 *//当最后静止或者是惯性滚动的时候* **if** (scrollState == ***SCROLL\_STATE\_IDLE*** || scrollState == ***SCROLL\_STATE\_FLING***) {  
 *//如果此时是最后一条可见* **if** (getLastVisiblePosition() >= getCount() - 1) {  
  
 *//1.加载更多布局* **footerView**.setPadding(8, 8, 8, 8); *//一般是setPading(0,0,0,0):此时是因为LinearLayout里面有padding  
 //2.状态改变* **isLoadMore** = **true**;  
 *//3.回调接口方法* **if** (**mOnRefreshListener** != **null**) {  
 **mOnRefreshListener**.onLoadMore();  
 }  
 }  
  
 }  
  
 }  
  
 @Override  
 **public void** onScroll(AbsListView view, **int** firstVisibleItem, **int** visibleItemCount, **int** totalItemCount) {  
  
 }  
  
 }  
  
  
 *//添加轮播图到外面的下拉刷新的那个布局，合成一个头* **public void** addTopNewsView(View topview) {  
 **if** (topview!=**null**){  
 *//判空是为了不让数据域空指针报错* **this**.**topview**=topview;  
 *//添加轮播图到外面的下拉刷新的那个布局，合成一个headerView就行了* **headerView**.addView(topview);  
 }  
  
 }  
  
}