Android 滚动监听篇

工程师 李洪江

一 View 的滚动监听

从 API 1 开始存在该方法，未公开接口和 set 方法。自从 API 23 开始才公开。

```
protected void onScrollChanged(int l, int t, int oldl, int oldt) {
    notifySubtreeAccessibilityStateChangedIfNeeded();

    if (AccessibilityManager.getInstance(mContext).isEnabled()) {
        postSendViewScrolledAccessibilityEventCallback();
    }

    mBackgroundSizeChanged = true;
    if (mForegroundInfo != null) {
        mForegroundInfo.mBoundsChanged = true;
    }

    final AttachInfo ai = mAttachInfo;
    if (ai != null) {
        ai.mViewScrollChanged = true;
    }

    if (mListenerInfo != null && mListenerInfo.mOnScrollChangeListener
        mListenerInfo.mOnScrollChangeListener.onScrollChange( v: this, l
    }
}
```

二 ScrollView 滚动监听

1 API 23 可以设置监听器.

```
@RequiresApi(api = Build.VERSION_CODES.M)
public NoScrollView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    setOnScrollChangeListener(new OnScrollChangeListener() {
        @Override
        public void onScrollChange(View v, int scrollX,
                                   int scrollY, int oldScrollX, int oldScrollY) {

        }
    });
}
```

2 兼容方法使用 ViewTreeObserver

```
getViewTreeObserver().addOnScrollChangedListener(
        new ViewTreeObserver.OnScrollChangedListener() {
    @Override
    public void onScrollChanged() {

    }
});
```

3 兼容方法，重写 onSizeChanged

```
@Override
protected void onScrollChanged(int l, int t, int oldl, int oldt) {
    super.onScrollChanged(l, t, oldl, oldt);
    if (mOnScrollChanged != null)
        mOnScrollChanged.onScroll(l, t, oldl, oldt);
}

public void setOnScrollChanged(OnScrollChanged onScrollChanged){
    this.mOnScrollChanged = onScrollChanged;
}
public interface OnScrollChanged{
    void onScroll(int l, int t, int oldl, int oldt);
}
```

```
public interface OnScrollListener{
    void onScroll(int scrollY);
}

@Override
protected void onScrollChanged(int l, int t, int oldl, int oldt) {
    super.onScrollChanged(l, t, oldl, oldt);
    if(listener != null){
        listener.onScroll(t);
    }
}
```

4 ScrollView 内部是通过 Scroller 来实现的，当滑动的时候会去调用 computeScroll()方法，从而达到监听的效果。

```
// 滑动距离监听器
public interface OnScrollListener{
    //在滑动的时候调用，scrollY为已滑动的距离
    void onScroll(int scrollY);
}

@Override
public void computeScroll() {
    super.computeScroll();
    if(listener!=null){
        listener.onScroll(getScrollY());
    }
}
```

三 AbsListView 滚动监听——AbsListView.OnScrollListener 是个内部接口

子类 ListView，GridView。

只能添加一个监听器，需要自己扩展。

```
 */
public void setOnScrollListener(OnScrollListener l) {
    mOnScrollListener = l;
    invokeOnItemScrollListener();
}

/**
 * Notify our scroll listener (if there is one) of a ch
 */
void invokeOnItemScrollListener() {
    if (mFastScroll != null) {
        mFastScroll.onScroll(mFirstPosition, getChildCo
    }
    if (mOnScrollListener != null) {
        mOnScrollListener.onScroll( view: this, mFirstPo
    }
    onScrollChanged( l: 0,  t: 0,  oldl: 0,  oldt: 0); // dummy
}
```

类代码

```
public interface OnScrollListener {

    public static int SCROLL_STATE_IDLE = 0;
    public static int SCROLL_STATE_TOUCH_SCROLL = 1;
    public static int SCROLL_STATE_FLING = 2;

    public void onScrollStateChanged(AbsListView view, int scrollState);

    public void onScroll(
            AbsListView view,
            int firstVisibleItem,
            int visibleItemCount,
            int totalItemCount);
}
```

四 RecyclerView 的滚动监听——RecyclerView.OnScrollListener 是个抽象内部类

类代码

```
public abstract static class OnScrollListener {

    public void onScrollStateChanged(RecyclerView recyclerView, int newState){}

    public void onScrolled(RecyclerView recyclerView, int dx, int dy){}
}
```

可以添加多个滚动监听器

```
public void addOnScrollChangedListener(OnScrollChangedListener listener) {
    checkIsAlive();

    if (mOnScrollChangedListeners == null) {
        mOnScrollChangedListeners = new CopyOnWriteArray<OnScrollChangedListener>()
    }

    mOnScrollChangedListeners.add(listener);
}
```

五 ViewPager 滚动监听——ViewPager.OnPageChangeListener 是个接口

类代码

```
public interface OnPageChangeListener {
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels);
    public void onPageSelected(int position);
    public void onPageScrollStateChanged(int state);
}
```

原生支持一个

```
public void setOnPageChangeListener(OnPageChangeListener listener) {
    mOnPageChangeListener = listener;
}
```

兼容库多个

```
/**
 * Set a listener that will be invoked whenever the page changes or is
 * scrolled. See {@link OnPageChangeListener}.
 *
 * @param listener Listener to set
 *
 * @deprecated Use {@link #addOnPageChangeListener(OnPageChangeListene
 * and {@link #removeOnPageChangeListener(OnPageChangeListener)} inste
 */
@Deprecated
public void setOnPageChangeListener(OnPageChangeListener listener) {
    mOnPageChangeListener = listener;
}
```

```
/**
 * Add a listener that will be invoked whenever the page changes or is
 * scrolled. See {@link OnPageChangeListener}.
 *
 * <p>Components that add a listener should take care to remove it whe
 * Other components that take ownership of a view may call {@link #cle
 * to remove all attached listeners.</p>
 *
 * @param listener listener to add
 */
public void addOnPageChangeListener(OnPageChangeListener listener) {
    if (mOnPageChangeListeners == null) {
        mOnPageChangeListeners = new ArrayList<>();
    }
    mOnPageChangeListeners.add(listener);
}
```

六 例子

1 滚动方向判断

2 滚动距离判断