



河北师范大学软件学院
Software College of Hebei Normal University

SmartRefreshLayout



智能设备教研室



ViewHolder作用



SmartRefreshLayout简介



基本用法





回顾

- ListView配合自定义Adapter显示数据
 - 布局中设置ListView控件
 - 设计Item子布局
 - 自定义Adapter并重写getView()等方法
 - 加载子布局页面及控件
 - 给控件绑定数据



不使用ViewHolder的情况

```
public View getView(int position, View convertView, ViewGroup parent) {  
    // 加载布局  
    convertView = LayoutInflater.from(context).inflate(R.layout.layout_item, root: null);  
    ImageView headImg = convertView.findViewById(R.id.header);  
    TextView nick = convertView.findViewById(R.id.nick);  
    TextView content = convertView.findViewById(R.id.content);  
    // 绑定数据  
    Map item = (Map) list.get(position);  
    if("boy".equals(item.get("sex").toString())) { //boy  
        // ...  
        headImg.setImageResource(R.mipmap.girl);  
    }  
    nick.setText(item.get("nick").toString());  
    content.setText(item.get("content").toString());  
    // 返回item对象  
    return convertView;  
}
```

View重复创建，效率低下，如何提高性能？



不使用ViewHolder的情况

//

加载布局

```
if(null == convertView) {  
    convertView = LayoutInflater.from(context).inflate(R.layout.layout_item, root: null);  
}  
ImageView headImg = convertView.findViewById(R.id.header);  
TextView nick = convertView.findViewById(R.id.nick);  
TextView content = convertView.findViewById(R.id.content);
```

是否从根本上提升了性能呢？



ViewHolder的作用

➤ 缓存View，避免重复创建。

```
public View getView(int position, View convertView, ViewGroup parent) {  
    ViewHolder holder = null;  
    if(null == convertView){  
        // 加载布局  
        convertView = LayoutInflater.from(context).inflate(R.layout.layout_item, root: null);  
        // 关联ViewHolder  
        holder = new ViewHolder();  
        holder.headerImg = convertView.findViewById(R.id.header_img);  
        holder.nickName = convertView.findViewById(R.id.nick_name);  
        holder.content = convertView.findViewById(R.id.content);  
        // 缓存ViewHolder对象  
        convertView.setTag(holder);  
    } else {  
        // 获取缓存的ViewHolder对象  
        holder = (ViewHolder) convertView.getTag();  
        // 给缓存的ViewHolder对象绑定数据  
        Map item = (Map) list.get(position);  
        if("boy".equals(item.get("sex").toString())) { //boy  
            holder.headerImg.setImageResource(R.mipmap.boy);  
        } else {  
            holder.headerImg.setImageResource(R.mipmap.girl);  
        }  
        holder.nickName.setText(item.get("nick").toString());  
        holder.content.setText(item.get("content").toString());  
        // 返回item对象  
        return convertView;  
    }  
}
```



ViewHolder的作用

- 通常出现在适配器中。
- 缓存数据，配合ListView等容器控件使用。
- 快速设置值，从而提升性能。



● ViewHolder作用

● SmartRefreshLayout简介

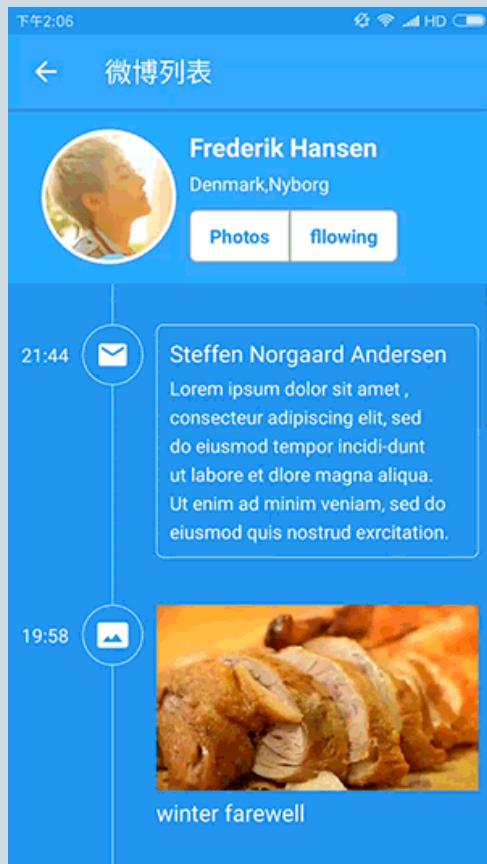


● 基本用法





SmartRefreshLayout简介





SmartRefreshLayout简介

- 智能刷新控件
- 功能
 - 下拉刷新
 - 上拉加载更多
- 支持嵌套
- 智能控制



SmartRefreshLayout简介

- 特点：对下拉刷新功能进行系统的拆分、组合
 - RefreshLayout 下拉的基本功能，包括布局测量、滑动事件处理、参数设定等等
 - RefreshContent 对不同内容的统一封装，包括判断是否可滚动、回弹判断、智能识别
 - RefreshHeader 下拉头部的实现和显示
 - RefreshFooter 上拉底部的实现和显示

参考：<https://segmentfault.com/a/1190000010066071>















SmartRefreshLayout简介

- 4种下拉刷新样式(`com.scwang.smartrefresh.layout.header`)
 - BezierRadarHeader
 - ClassicsHeader
 - FalsifyHeader
 - TwoLevelHeader
- 3种上拉加载样式(`com.scwang.smartrefresh.layout.footer`)
 - BallPulseFooter
 - ClassicsFooter
 - FalsifyFooter



更多下拉刷新方式

➤ SmartRefreshHeader

-  BezierCircleHeader.java
-  DeliveryHeader.java
-  DropboxHeader.java
-  FlyRefreshHeader.java
-  FunGameBattleCityHeader.java
-  FunGameHitBlockHeader.java
-  MaterialHeader.java
-  PhoenixHeader.java
-  StoreHouseHeader.java
-  TaurusHeader.java
-  WaterDropHeader.java
-  WaveSwipeHeader.java



● ViewHolder作用

● SmartRefreshLayout简介

● 基本用法





基本用法

➤ 导入依赖

- implementation 'com.scwang.smartrefresh:
SmartRefreshLayout:1.1.0-alpha-12'

➤ 在XML布局中添加智能刷新控件

- com.scwang.smartrefresh.layout.SmartRefreshLayout

➤ 设计刷新头、刷新尾的样式

➤ 添加事件监听器

- setOnLoadMoreListener
- setOnRefreshListener



基本用法

➤ 在XML布局中添加智能刷新控件

```
<com.scwang.smartrefresh.layout.SmartRefreshLayout  
    android:id="@+id/refreshLayout"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:srlEnableFooterFollowWhenLoadFinished="true"  
    app:srlEnableScrollContentWhenLoaded="true">
```

.....

```
</com.scwang.smartrefresh.layout.SmartRefreshLayout>
```




基本用法

➤ 设置刷新样式（方法一：全局方式）

```
public class App extends Application {  
    static { //static 代码段可以防止内存泄露  
        //设置全局的Header构建器  
        SmartRefreshLayout.setDefaultRefreshHeaderCreator(  
            new DefaultRefreshHeaderCreator() {  
                public RefreshHeader createRefreshHeader(Context context,  
                                                         RefreshLayout layout) {  
                    //全局设置主题颜色  
                    layout.setPrimaryColorsId(R.color.colorPrimary, android.R.color.white);  
                    return new ClassicsHeader(context);  
                }  
            });  
    }  
};
```



基本用法

➤ 设置刷新样式（方法一：全局方式）

//设置全局的Footer构建器

```
SmartRefreshLayout.setDefaultRefreshFooterCreator(  
    new DefaultRefreshFooterCreator() {  
        RefreshFooter createRefreshFooter(Context context,  
                                           RefreshLayout layout) {  
            //指定为经典Footer，默认是 BallPulseFooter  
            return new ClassicsFooter(context).setDrawableSize(20);  
        }  
    });  
}
```



基本用法

➤ 设置刷新样式（方法二：XML方式）

```
<LinearLayout .....>
    <com.scwang.smartrefresh.layout.SmartRefreshLayout .....>
        <com.scwang.smartrefresh.header.BezierCircleHeader
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <ListView ...../>
        <com.scwang.smartrefresh.layout.footer.ClassicsFooter
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:srlClassicsSpinnerStyle="Translate" />
    </com.scwang.smartrefresh.layout.SmartRefreshLayout>
</LinearLayout>
```

注意：app属性，需添加命名空间xmlns:app=<http://schemas.android.com/apk/res-auto>



基本用法

➤ 设置刷新样式（方法三：Java方式）

//设置 Header 为 贝塞尔雷达 样式

```
refreshLayout.setRefreshHeader(new BezierCircleHeader(this))  
                .setHorizontalFadingEdgeEnabled(true);
```

//设置 Footer 为 球脉冲 样式

```
refreshLayout.setRefreshFooter(new BallPulseFooter(this)  
                .setSpinnerStyle(SpinnerStyle.Scale));
```

注意：三种样式优先级

Java方式 > XML方式 > 全局方式



自定义刷新头样式

```
<com.scwang.smartrefresh.layout.SmartRefreshLayout  
.....>
```

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:src="@mipmap/img"/>
```



```
<ListView .....>  
</com.scwang.smartrefresh.layout.SmartRefreshLayout>
```



自定义刷新头样式

```
<com.scwang.smartrefresh.layout.SmartRefreshLayout  
.....>
```

```
<pl.droidsonroids.gif.GifImageView  
    android:layout_width="match_parent"  
    android:layout_height="150dp"  
    android:scaleType="centerCrop"  
    android:src="@mipmap/eat"/>
```



动态
效果

注意：加载gif格式的动态图片，需添加依赖

implementation 'pl.droidsonroids.gif:android-gif-drawable:1.2.3'

```
</com.scwang.smartrefresh.layout.SmartRefreshLayout>
```



附录：SmartRefreshLayout主要属性方法

- `setPrimaryColorsId(R.color.colorPrimary, android.R.color.white);`//设置主题颜色
- `setDragRate(0.5f);`//显示手指下拉高度
- `setReboundDuration(300);`//回弹动画时长（毫秒）
- `setHeaderHeight(100);`//Header标准高度（显示下拉高度 \geq 标准高度 触发刷新）
- `setHeaderMaxDragRate(2);`//最大Header显示下拉高度
- `setEnabledRefresh(true);`//是否启用下拉刷新功能
- `setEnabledLoadMore(false);`//是否启用上拉加载功能



附录： SmartRefreshLayout主要属性方法

- `finishRefresh();`//结束刷新
 - `finishLoadMore();`//结束加载
 - `finishRefresh(3000);`//延迟3000毫秒后结束刷新
 - `finishLoadMore(3000);`//延迟3000毫秒后结束加载
 - `finishRefresh(false);`//结束刷新（刷新失败）
 - `finishLoadMore(false);`//结束加载（加载失败）
 - `finishLoadMoreWithNoMoreData();`//完成加载并标记没有更多数据
- 1.0.4
- `autoRefresh(400),autoLoadMore(400);`//延迟400ms后刷新、加载



附录： SmartRefreshHeader主要属性方法

```
header.setTextTimeMarginTop(10); //设置时间文字的上边距 (dp单位)
header.setTextTimeMarginTopPx(10); //同上-像素单位 (1.1.0版本删除)
header.setEnableLastTime(true); //是否显示时间
header.setFinishDuration(500); //设置刷新完成显示的停留时间 (设为0可以关闭停留功能)
header.setDrawableSize(20); //同时设置箭头和图片的大小 (dp单位)
header.setDrawableArrowSize(20); //设置箭头的大小 (dp单位)
header.setDrawableProgressSize(20); //设置图片的大小 (dp单位)
header.setDrawableMarginRight(20); //设置图片和箭头和文字的间距 (dp单位)
header.setDrawableSizePx(20); //同上-像素单位
header.setArrowBitmap(bitmap); //设置箭头位图 (1.1.0版本删除)
header.setArrowDrawable(drawable); //设置箭头图片
header.setArrowResource(R.drawable.ic_arrow); //设置箭头资源
header.setProgressBitmap(bitmap); //设置图片位图 (1.1.0版本删除)
header.setProgressDrawable(drawable); //设置图片
header.setProgressResource(R.drawable.ic_progress); //设置图片资源
```



附录： SmartRefreshLayout主要属性方法

```
header.setTextTimeMarginTop(10);//设置时间文字的上边距（dp单位）
header.setTextTimeMarginTopPx(10);//同上-像素单位（1.1.0版本删除）
header.setEnableLastTime(true);//是否显示时间
header.setFinishDuration(500);//设置刷新完成显示的停留时间（设为0可以关闭停留功能）
header.setDrawableSize(20);//同时设置箭头和图片的大小（dp单位）
header.setDrawableArrowSize(20);//设置箭头的大小（dp单位）
header.setDrawableProgressSize(20);//设置图片的大小（dp单位）
header.setDrawableMarginRight(20);//设置图片和箭头和文字的间距（dp单位）
header.setDrawableSizePx(20);//同上-像素单位
header.setArrowBitmap(bitmap);//设置箭头位图（1.1.0版本删除）
header.setArrowDrawable(drawable);//设置箭头图片
header.setArrowResource(R.drawable.ic_arrow);//设置箭头资源
header.setProgressBitmap(bitmap);//设置图片位图（1.1.0版本删除）
header.setProgressDrawable(drawable);//设置图片
header.setProgressResource(R.drawable.ic_progress);//设置图片资源
```



附录：Header属性

name	format	description
srlPrimaryColor	color	主题颜色
srlAccentColor	color	强调颜色
srlDrawableArrow	drawable	箭头图片
srlDrawableProgress	drawable	转动图片
srlClassicsSpinnerStyle	enum	变换样式：Translate(平行移动)、Scale（拉伸形变）、FixedBehind（固定在背后）
srlSpinnerStyle	enum	变换样式：srlClassicsSpinnerStyle的全部、FixedFront（固定在前面或全屏）
srlFinishDuration	int	动画结束时，显示完成状态停留的时间（毫秒）
srlEnableLastTime	boolean	是否显示上次更新时间（默认true）
srlDrawableMarginRight	dimension	图片相对右边文字的距离（默认20dp）
srlTextTimeMarginTop	dimension	更新时间相对上面标题的距离（默认2dp）
srlTextSizeTitle	dimension	标题文字大小（默认16sp）
srlTextSizeTime	dimension	时间文字大小（默认12sp）



附录：Header方法

name	format	description
setPrimaryColor	color	主题颜色
setAccentColor	color	强调颜色
setArrowDrawable	drawable	设置箭头图片
setProgressDrawable	drawable	设置转动图片
setArrowBitmap	bitmap	设置箭头图片(V1.1.0版本删除)
setProgressBitmap	bitmap	设置转动图片(V1.1.0版本删除)



附录：Header方法

setArrowResource	int	设置箭头图片
setProgressResource	int	设置转动图片
setSpinnerStyle	enum	变换样式：参考属性srlSpinnerStyle
setClassicsSpinnerStyle	enum	变换样式：参考属性srlClassicsSpinnerStyle
setFinishDuration	int	设置动画结束时，显示完成状态停留的时间（毫秒）
setEnabledLastTime	boolean	是否显示上次更新时间（默认true）
setTextSizeTitle	dimension	标题文字大小（默认16sp）
setTextSizeTime	dimension	时间文字大小（默认12sp）
setLastUpdateTime	string	手动设置更新时间，将不会自动更新时间



附录

➤ 更过属性和方法参考：

- https://github.com/scwang90/SmartRefreshLayout/blob/master/art/md_property.md



Thank You!

