

特性：

ListView、GridView 的升级版控件

- 你想要控制其显示的方式，请通过布局管理器LayoutManager
- 你想要控制Item间的间隔（可绘制），请通过ItemDecoration
- 你想要控制Item增删的动画，请通过ItemAnimator

使用方法：

1、添加依赖

```
compile 'com.android.support:recyclerview-v7:25.1.0'
```

2、通过布局中的ID获取RecyclerView对象

```
1 mRecyclerView = findViewById(R.id.id_recyclerview);
```

```
1 3、设置布局管理器
```

```
LinearLayoutManager manager = new LinearLayoutManager(this);
```

```
mRlv.setLayoutManager(manager);
```

注意：如果有使用网络获取数据那么第四步就是通过网络获取数据转为JAVABean对象

4、创建适配器

```
MAdapter adapter = new MAdapter(getApplicationContext());
```

设置Adapter数据适配器

```
mRlv.setAdapter(adapter);
```

5、创建袋子 自定义类继承 RecyclerView.ViewHolder

```
class MViewHolder extends RecyclerView.ViewHolder
```

用来获取Item中的控件对象

如：

```
public TextView mTextView;
```

```
public MViewHolder(View itemView) {  
    super(itemView);  
    mTextView = (TextView) itemView.findViewById(R.id.img_text);  
}
```

6、创建数据适配器Adapter 自定义类继承 RecyclerView.Adapter<MViewHolder>

```
class MAdapter extends RecyclerView.Adapter<MViewHolder> 泛型为：袋子类名
```

重写三个方法：

一、onCreateViewHolder(ViewGroup parent, int viewType) 获取袋子

```
View view = mInflater.inflate(R.layout.view_rc_item, parent, false);
```

```
MViewHolder holder = new MViewHolder(view);
```

```
return holder;
```

二、onBindViewHolder(MViewHolder holder, int position) 为Item子控件设置数据

```
holder.mTextView.setText(myData.get(position).getName()+"最新的");
```

三、getItemCount() 获取Item的条目总数

开发步骤详细说明：

1、解析布局获取RecyclerView对象

2、设置布局管理器

线性布局管理器

```
LayoutManager layout = new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, true);
```

```
LayoutManager layout = new LinearLayoutManager(this);
```

设置布局样式

```
mRecyclerView.setLayoutManager(layout);
```

网格布局

```
LayoutManager layout = new GridLayoutManager(this, 2);
```

```
LayoutManager layout = new GridLayoutManager(this, 2, GridLayoutManager.VERTICAL, false);
```

设置布局样式

```
mRecyclerView.setLayoutManager(layout);
```

流式布局管理器

```
LayoutManager layout = new StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.VERTICAL);
```

设置布局样式

```
mRecyclerView.setLayoutManager(layout);
```

3、通过网络获取数据并解析数据

开子线程获取数据，拿到数据后开主线程设置数据适配器

4、创建RecyclerView.ViewHolder

找到item的控件对象 itemView.findViewById(R.id.lv_tv);

5、创建数据适配器 RecyclerView.Adapter<ViewHolder> 泛型为Holder类

1、通过打气筒找到item布局，并创建Holder对象，返回Holder对象

2、设置数据到Item控件

3、返回Item条目总数

6、在解析到数据之后设置数据适配器

```
runOnUiThread(new Runnable() {  
    @Override  
    public void run() {  
        创建适配器对象  
        RecyclerView.Adapter<ViewHolder> adapter = new RecyclerView.Adapter<ViewHolder>() {  
            设置适配器  
            @Override  
            public int getItemCount() {  
                return data.size();  
            }  
            @Override  
            public ViewHolder onCreateViewHolder(Context context, int viewType) {  
                创建item布局  
                LayoutInflater inflater = LayoutInflater.from(context);  
                View view = inflater.inflate(R.layout.item, null);  
                ViewHolder holder = new ViewHolder(view);  
                return holder;  
            }  
            @Override  
            public void onBindViewHolder(ViewHolder holder, int position) {  
                设置数据到item  
                holder.itemView.setText(data.get(position));  
            }  
        };  
        mRecyclerView.setAdapter(adapter);  
    }  
});
```

RecyclerView设置Item的点击事件：

在自定义extends RecyclerView.Adapter<MyAdapter.Myholder> 类的

onBindViewHolder(Myholder holder, int position) 方法里设置点击事件

通过判断是否有点击事件然后设置点击事件

```
if(!holder.itemView.hasOnClickListeners()) {  
    holder.itemView.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            Toast.makeText(mContext, "你点了" + position, Toast.LENGTH_SHORT).show();  
        }  
    });  
}
```

添加头部设置下接刷新：

1、获取头布局

```
View headerView = LayoutInflater.from(mContext).inflate(R.layout.view_rv_news_head, (ViewGroup) mMain
```

2、获取头布局的控件

```
mVpNewsHeader = (ViewPager) headerView.findViewById(R.id.vp_news_header);  
mIndicator = (CirclePageIndicator) headerView.findViewById(R.id.vp_indicator);  
mTvNewsHeaderTitler = (TextView) headerView.findViewById(R.id.tv_news_head_title);
```

3、将头布局添加到RecyclerView

```
mRecyclerView.addHeaderView(headerView);
```

compile 'com.jcodecraeer:xrecyclerview:1.3.2' 三方依赖

// 设置可以加载更多

```
mRecyclerView.setLoadingMoreEnabled(true);
```

// 设置下拉刷新的样式

```
mRecyclerView.setRefreshProgressStyle(ProgressStyle.TriangleSkewSpin);  
mRecyclerView.setLoadingMoreProgressStyle(ProgressStyle.Pacman);  
mRecyclerView.setArrowImageView(R.drawable.common_listview_headview_red_arrow);
```

//下拉刷新

```
mRecyclerView.setLoadingListener(new XRecyclerView.LoadingListener() {  
    @Override  
    public void onRefresh() {  
        mHandler.postDelayed(new Runnable() {  
            @Override  
            public void run() {  
                Toast.makeText(mContext, "下拉刷新完成", Toast.LENGTH_SHORT).show();  
                mRecyclerView.refreshComplete();  
            }  
        }, 3000);  
    }  
});
```

//上拉加载更多

```
@Override  
public void onLoadMore() {  
    mHandler.postDelayed(new Runnable() {  
        @Override  
        public void run() {  
            if(!TextUtils.isEmpty(mMoreUrl)) {  
                mMoreUrl= Constants.HOST+mMoreUrl;  
                HttpUtils<NewsDetailBean> httpUtils = new HttpUtils<NewsDetailBean>(mMainActivity);  
                httpUtils.getData(mMoreUrl, NewsDetailBean.class, new HttpUtils.OnResultListener<NewsDetailBean>() {  
                    @Override  
                    public void onResult(NewsDetailBean b) {  
                        List<NewsDetailBean.DataBean.NewsBean> moreNews = b.getData().getNews();  
                        mNewsBeen.addAll(moreNews);  
                        // 更新UI  
                        if(mNewsAdapter!=null){  
                            mNewsAdapter.notifyDataSetChanged();  
                        }  
                        // mMoreUrl 重新赋值  
                        mMoreUrl=b.getData().getMore();  
                        Toast.makeText(mContext, "加载更多完成", Toast.LENGTH_SHORT).show();  
                        mRecyclerView.loadMoreComplete();  
                    }  
                });  
            }else{  
                Toast.makeText(mContext, "没有更多数据，好好休息一下", Toast.LENGTH_SHORT).show();  
                mRecyclerView.loadMoreComplete();  
            }  
        }  
    });  
}
```

```
}
```

```
}
```

```
}, 1000);
```

```
}
```

```
});
```