RecyclerView

一、简介

这个是谷歌官方出的控件，使我们可以非常简单的做出列表装的一个控件，当然recyclerview的功能不止这些，它还可以做出瀑布流的效果，这是一个非常强大的控件，内部自带viewholder可以使我们非常简单的完成许多操作，正在一步一步取代listview这个控件，当然它也有一些小的缺点，那就是谷歌官方并没有直接给我写出它的点击事件的接口，但是这并难不倒我们，我们可以自己写一个回调的接口，实现点击事件，在这里我不仅要为大家介绍recyclerview的item的点击事件，还要为大家介绍，一个item中局部的点击事件，还有添加header、footer，还有添加不同类别的item的布局。可以说彻底的读懂了这篇文章，我们对recyclerview就有了一个新的认识了。

二、涉及到的知识点

- item的点击事件
- item里面内容的点击事件
- 为recycler view添加header和footer
- 为item添加不同的布局

三、实现代码

```
/**
 * Created by linSir on 16/7/24.管理地址界面的适配器
 */
public class ManageAddressAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {//在这里我们要

    public OnTitleClickListener mListener;
    private List<AllAddress> mList;  //用户列表

    public ManageAddressAdapter() {
        mList = new ArrayList<>();
    }

    public void setList(List<AllAddress> list) {//从外界传入一个list
        mList.clear();
        mList.addAll(list);
        notifyDataSetChanged();
    }


    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {//重写方法，用上
        return new ManageAddressViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.
    }

    @Override public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {//在这里我们
        ManageAddressViewHolder mHolder = (ManageAddressViewHolder) holder;
        mHolder.userName.setText(mList.get(position).getShipName());
        mHolder.userTel.setText(mList.get(position).getPhone());

        String address = mList.get(position).getProvince() + " " + mList.get(position).getCity()
                + " " + mList.get(position).getArea() + " " + mList.get(position).getDetail();
        mHolder.address.setText(address);

        mHolder._default.setOnClickListener(new ClickListener(String.valueOf(position)));//在这里我们设

    }

    @Override public int getItemCount() {
        return mList.size();
    }


    public static class ManageAddressViewHolder extends RecyclerView.ViewHolder {

        private TextView userName;
        private TextView userTel;
        private TextView address;
        private TextView _default;

        public ManageAddressViewHolder(View itemView) {
            super(itemView);
            userName = (TextView) itemView.findViewById(R.id.manage_userName);
            userTel = (TextView) itemView.findViewById(R.id.manage_userTel);
            address = (TextView) itemView.findViewById(R.id.manage_userAddress);
            _default = (TextView) itemView.findViewById(R.id.manage_default);


        }
    }


    public class ClickListener implements View.OnClickListener {//在这里我们重写了点击事件
        private String id;

        public ClickListener(String id) {
            this.id = id;
        }

        @Override public void onClick(View view) {
            if (mListener != null) {
                mListener.onTitleClick(id);
            }
        }
    }


    public void setOnTitleClickListener(OnTitleClickListener listener) {//自己写了一个方法，用上我们的接口
        mListener = listener;
    }


    public interface OnTitleClickListener {//自己写了一个点击事件的接口
        void onTitleClick(String id);
    }


}
```

通过以上的代码，我们已经为recyclerView里面的item里面的textview添加成功了点击事件，我们只需要在调用它的界面实现这个接口，然后重写点击事件的方法，就可以实现这个textview的点击事件了，下面我们一起看一下代码：

```
/**
 * Created by linSir on 16/7/24.管理地址界面
 */
public class ManageAddressActivity extends AppCompatActivity implements ManageAddressAdapter.OnTitleC

    private ManageAddressAdapter mAdapter;
    private List<AllAddress> list;
    @BindView(R.id.manage_address_recyclerView) RecyclerView mRecyclerView;

    @Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manage);
        ButterKnife.bind(this);
        list = new ArrayList<AllAddress>();
        mAdapter = new ManageAddressAdapter();
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this);
        mAdapter.setOnTitleClickListener(this);//声明一下
        mRecyclerView.setAdapter(mAdapter);
        mRecyclerView.setLayoutManager(linearLayoutManager);//这里千万不要了为recyclerview设置布局


    }

    @OnClick(R.id.back_manage_address)
    public void back() {
        finish();
    }

    @OnClick(R.id.manage_add_address)
    public void add() {
        Intent intent = new Intent(ManageAddressActivity.this, EditAddress.class);
        startActivity(intent);
    }

    @Override
    protected void onResume() {
        super.onResume();
        final HttpResultListener<List<AllAddress>> listener;

        listener = new HttpResultListener<List<AllAddress>>() {
            @Override
            public void onSuccess(List<AllAddress> allAddresses) {
                Toast.makeText(ManageAddressActivity.this, "获取收货成功", Toast.LENGTH_SHORT).show();
                mAdapter.setList(allAddresses);
                list = allAddresses;

            }

            @Override
            public void onError(Throwable e) {
                Log.i("lin", "----lin----> 获取收货地址失败 " + e.toString());
            }
        };
        ApiService5.getInstance().allAddress(listener, 1);
    }

    @Override public void onTitleClick(String id) {//这里便是我们重写了的点击事件
        Log.i("lin", "----lin----> onTitleClick 的 id " + id);
    }
}
```

以上的代码，我们实现了，为recyclerView的一个item中的textview添加的点击事件，下面我要为大家介绍一下，如何为recyclerView中item添加不同的布局文件，并且如何为item整个添加点击事件：

```
/**
 * Created by lin_sir on 2016/7/7.部分商品的展示,的适配器
 */
public class BuyerRecyclerAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {

    public static final int FOOTER_TYPE = 0;//最后一个的类型
    public static final int HAS_IMG_TYPE = 1;//有图片的类型

    private List<FamousPageModel> dataList;

    private ProgressBar mProgress;
    private TextView mNoMore;

    public BuyerRecyclerAdapter() {
        dataList = new ArrayList<>();
    }

    public void addData(List<FamousPageModel> list) {
        dataList.addAll(list);
        notifyDataSetChanged();
    }

    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        if (viewType == FOOTER_TYPE) {
            return new FooterItemViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout
        } else {
            return new BuyerItemViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.
        }

    }

    @Override
    public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {
        int type = getItemViewType(position);
        if (type == FOOTER_TYPE) {
            bindFooterView((FooterItemViewHolder) holder);
        } else {
            bindView((BuyerItemViewHolder) holder, dataList.get(position));
        }
    }

    @Override
    public int getItemViewType(int position) {
        if (position + 1 == getItemCount()) {
            return FOOTER_TYPE;
        } else {
            FamousPageModel news = dataList.get(position);
            return HAS_IMG_TYPE;
        }
    }

    private void bindView(BuyerItemViewHolder holder, FamousPageModel data) {

        String productName = data.getProductName();
        String[] products = productName.split(" ");
        if (products.length != 1) {
            productName = products[1] + " 等商品";
        }

        String count = data.getNum();
        int sum = 0;
        String[] counts = count.split(" ");
        if (counts.length == 2) {
            sum = Integer.parseInt(counts[0]) + Integer.parseInt(counts[1]);
            count = String.valueOf(sum);
        }

        if (counts.length == 3) {
            sum = Integer.parseInt(counts[0]) + Integer.parseInt(counts[1]) + Integer.parseInt(counts
            count = String.valueOf(sum);
        }

        holder.count.setText(count);
        holder.goods_name.setText(productName);
        holder.price.setText(data.getTotal());
        if (data.getUser() != null) {
            holder.user_name.setText(data.getUser().getName());

        }

        String imgUrl = data.getPicture();

        if (imgUrl != null) {
            ImageUtil.requestImg(BaseApplication.get().getAppContext(), imgUrl, holder.img);
        }

//          Picasso.with(BaseApplication.get().getAppContext()).load(data.getPicture()).into(holder.img
    }


    @Override
    public int getItemCount() {
        return dataList == null ? 0 : dataList.size() + 1;
    }

    public static class BuyerItemViewHolder extends RecyclerView.ViewHolder {
```

```
    public static class BuyerItemViewHolder extends RecyclerView.ViewHolder {

        private ImageView img;
        private TextView price;
        private TextView goods_name;
        private TextView count;
        private TextView user_name;


        public BuyerItemViewHolder(View itemView) {

            super(itemView);

            img = (ImageView) itemView.findViewById(R.id.iv_user_item_buyer2);
            price = (TextView) itemView.findViewById(R.id.tv_product_price);
            goods_name = (TextView) itemView.findViewById(R.id.tv_product_name);
            count = (TextView) itemView.findViewById(R.id.tv_product_number);
            user_name = (TextView) itemView.findViewById(R.id.tv_user_name);

        }
    }


    /**
     * 刷新列表
     */
    public void refreshList(List<FamousPageModel> list) {
        dataList.clear();
        dataList.addAll(list);
        notifyDataSetChanged();
    }

    /**
     * 加载更多
     */
    public void loadMoreList(List<FamousPageModel> list) {
        dataList.addAll(list);
        notifyDataSetChanged();
    }

    /**
     * 显示没有更多
     */
    public void showNoMore() {
        if (getItemCount() > 0) {
            if (mProgress != null && mNoMore != null) {
                mNoMore.setVisibility(View.VISIBLE);
                mProgress.setVisibility(View.GONE);
            }
        }
    }


    /**
     * 显示加载更多
     */
    public void showLoadMore() {
        if (mProgress != null && mNoMore != null) {
            mProgress.setVisibility(View.VISIBLE);
            mNoMore.setVisibility(View.GONE);
        }
    }

    private void bindFooterView(FooterItemViewHolder viewHolder) {
        mProgress = viewHolder.mProgress;
        mNoMore = viewHolder.tvNoMore;
    }


    public static class FooterItemViewHolder extends RecyclerView.ViewHolder {

        private ProgressBar mProgress;
        private TextView tvNoMore;

        public FooterItemViewHolder(View itemView) {
            super(itemView);
            mProgress = (ProgressBar) itemView.findViewById(R.id.pb_footer_load_more);
            tvNoMore = (TextView) itemView.findViewById(R.id.tv_footer_no_more);
        }
    }


    /**
    /**
     * 获取点击的 item,如果是最后一个,则返回 null
     */
    public FamousPageModel getClickItem(int position) {
        if (position < dataList.size()) {
            return dataList.get(position);
        } else {
            return null;
        }
    }

}
```

在这里我们首先重写了getItemViewType，这里面我目前只写了两种类型，那就是带有图片的类型，和不带有图片的类型，并且我让不带图片的类型出现在了最后面，当然大家可以随意的设置，我们只需要根据它们的特征为他们分好类，然后根据不同的类型加载不同的布局文件即可。

```
/**
 * Created by lin_sir on 2016/7/7. buyer界面
 */
public class FragmentBuyer extends Fragment implements SwipeRefreshLayout.OnRefreshListener {

    private static int CURRENT_PAGE = 1;                                //获取需要请求的页号
    private RecyclerView recyclerView;                                  //recyclerView
    private BuyerRecyclerAdapter madapter;                              //适配器
    private List<FamousPageModel> mlist;                                //一个装载数据的集合
    private LinearLayoutManager linearLayoutManager;                    //linearLayoutManger
    private HttpResultListener<List<FamousPageModel>> listener;         //数据请求的回调接口
    private SwipeRefreshLayout refreshLayout;                           //下拉刷新控件
    private boolean isLoadMore;                                         //是否加载更多

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle
        View view = inflater.inflate(R.layout.fragment_buyer, container, false);
        initviews(view);
        initListener();
        refreshData();
        ButterKnife.bind(this, view);
        return view;
    }

    private void initListener() {
        listener = new HttpResultListener<List<FamousPageModel>>() {
            @Override
            public void onSuccess(List<FamousPageModel> list) {
                Log.i("lin", "----lin---->  refresh  success");
                refreshLayout.setRefreshing(false);
                if (isLoadMore) {
                    mlist = list;
                    madapter.loadMoreList(mlist);
                    madapter.notifyDataSetChanged();
                    madapter.showNoMore();
                } else {
                    mlist = list;
                    madapter.refreshList(list);
                    madapter.notifyDataSetChanged();
                }
            }

            @Override
            public void onError(Throwable e) {
                Log.i("lin", "----lin---->   refresh  error  " + e.toString());
                refreshLayout.setRefreshing(false);
                madapter.showNoMore();
            }
        };
    }

    private void initviews(View view) {

        mlist = new ArrayList<>();

        refreshLayout = (SwipeRefreshLayout) view.findViewById(R.id.refresh_news);
        recyclerView = (RecyclerView) view.findViewById(R.id.rv_buyer);

        refreshLayout.setColorSchemeResources(R.color.blue_500, R.color.purple_500, R.color.green_500
        refreshLayout.setOnRefreshListener(this);
        linearLayoutManager = new LinearLayoutManager(getActivity());
        linearLayoutManager.setOrientation(OrientationHelper.VERTICAL);

        madapter = new BuyerRecyclerAdapter();
        madapter.addData(mlist);

        recyclerView.setLayoutManager(linearLayoutManager);
        recyclerView.setAdapter(madapter);
        recyclerView.addOnScrollListener(new OnRecyclerScrollListener());
        recyclerView.addOnItemTouchListener(new RecyclerItemClickListener(getActivity(), onItemClickL
    }

    /**
     * 刷新时,默认请求第一页的数据
     */
    private void refreshData() {
        refreshLayout.setRefreshing(true);
        isLoadMore = false;
        CURRENT_PAGE = 1;
        requestData(1);
    }

    /**
     * 加载更多
     */
    private void loadMoreData() {
        refreshLayout.setRefreshing(false);// 加载更多与刷新不能同时存在
        isLoadMore = true;
        requestData(++CURRENT_PAGE);
    }

    public class OnRecyclerScrollListener extends RecyclerView.OnScrollListener {

        int lastVisibleItem = 0;

        @Override
        public void onScrollStateChanged(RecyclerView recyclerView, int newState) {
            super.onScrollStateChanged(recyclerView, newState);

            if (madapter != null && newState == RecyclerView.SCROLL_STATE_IDLE && lastVisibleItem + 1
                loadMoreData();
            }
        }

        @Override
        public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
            super.onScrolled(recyclerView, dx, dy);
            lastVisibleItem = linearLayoutManager.findLastVisibleItemPosition();
        }
    }

    /**
     * 请求数据
     */
    private void requestData(int page) {
        madapter.showLoadMore();
        ApiService2.getInstance().famous(listener, page);
    }

    @Override
    public void onRefresh() {
        refreshData();
    }

    private RecyclerItemClickListener.OnItemClickListener onItemClickListener = new RecyclerItemClick
        @Override
        public void onItemClick(View view, int position) {
            Toast.makeText(getActivity(), "您点击的是:    " + position, Toast.LENGTH_SHORT).show();

        }
    };


    @OnClick(R.id.release_distance_buyer)
    public void release() {
        Intent intent = new Intent(getActivity(), ReleaseOrderActivity.class);
        startActivity(intent);


    }

    @Override public void onDestroy() {
        super.onDestroy();
        }

}
```

以上便是根据我最近做的项目粗略的整理了一下，recyclerview的简单用法，在这里我也用到了，下拉刷新，上拉加载更多，等等等等的方法，写这些也是为了让自己在以后遇到同样的需求的时候，可以很快的写出，如果大家也有这些小问题的可以和我一起交流。