



Android ViewHolder 篇

工程师 李洪江

— 在 5.0 之前，ListView 官方推荐写法：

1 convertView 复用 View

2 Tag 保持 ViewHolder 对象。

2 ViewHolder 的子 View 复用。在创建 ViewHolder 对象时候，调试 findViewById 一次。

3 缓存数据复用。

```
@Override
public View getView(int position, View convertView, ViewGroup parent)
{
    ViewHolder viewHolder = null;
    if (null == convertView)
    {
        viewHolder = new ViewHolder();
        LayoutInflater mInflater = LayoutInflater.from(mContext);
        convertView = mInflater.inflate(R.layout.item_marker_item, null);

        viewHolder.name = (TextView) convertView.findViewById(R.id.name);
        viewHolder.description = (TextView) convertView
            .findViewById(R.id.description);
        viewHolder.createTime = (TextView) convertView
            .findViewById(R.id.createTime);

        convertView.setTag(viewHolder);
    }
    else
    {
        viewHolder = (ViewHolder) convertView.getTag();
    }

    // set item values to the viewHolder:
    MarkerItem markerItem = getItem(position);
    if (null != markerItem)
    {
        viewHolder.name.setText(markerItem.getName());
        viewHolder.description.setText(markerItem.getDescription());
        viewHolder.createTime.setText(markerItem.getCreateDate());
    }
    return convertView;
}

private static class ViewHolder
{
    TextView name;
    TextView description;
    TextView createTime;
}
```



二 在 5.0 之后，官方推出了 RecyclerView 升级版。

1 明确了 ViewHolder 写法。

2 创建和复用 ViewHolder 分开。

```
public class SupetRecyclerViewHolder extends ViewHolder {
    private View mWholeView;

    public SupetRecyclerViewHolder(View itemView) {
        super(itemView);
        this.mWholeView = itemView;
    }

    public View getWholeView() {
        return this.mWholeView;
    }
}
```

目前个人觉得统一写法：

```
public abstract class ViewHolder {
    public Context context;
    public View mWholeView;

    public ViewHolder(ViewGroup viewGroup) {
        this.context = viewGroup.getContext();
        this.mWholeView = LayoutInflater.from(context)
            .inflate(getLayoutId(), viewGroup, false);
        this.mWholeView.setTag(this);
        assignViews();
        setListeners();
    }

    public View getWholeView() {
        return mWholeView;
    }

    public abstract void setListeners();
    public abstract void assignViews();
    public abstract @LayoutRes int getLayoutId();
}
```