

2.4.6 BaseAdapter优化

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

本节引言：

上一节中我们学习了如何来使用一个ListView以及自定义一个简单的BaseAdapter，我们从代码中可以看出比较重要的两个方法：getCount()和getView()，界面上有多少列就会调用多少次getView，这个时候可能看出一些端倪，每次都是新inflate一个View，都要进行这个XML的解析，这样会很浪费资源，当然，几十列或者几百列的列表并不能体现什么问题，但假如更多或者布局更加复杂？所以学习ListView的优化很重要，而本节针对的是BaseAdapter的优化，优化的两点有，复用convertView 以及使用ViewHolder重用组件，不用每次都findViewById，我们具体通过代码来体会吧！

1.复用convertView：

上面也说了，界面上有多少个Item，那么getView方法就会被调用多少次！我们来看看上一节我们写的getView()部分的代码：

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    convertView = LayoutInflater.from(mContext).inflate(R.layout.item_list_animal, parent, false);

    ImageView img_icon = (ImageView) convertView.findViewById(R.id.img_icon);

    TextView txt_aName = (TextView) convertView.findViewById(R.id.txt_aName);

    TextView txt_aSpeak = (TextView) convertView.findViewById(R.id.txt_aSpeak);

    img_icon.setBackgroundResource(mData.get(position).getIcon());
    txt_aName.setText(mData.get(position).getName());
    txt_aSpeak.setText(mData.get(position).getSpeak());

    return convertView;
}
```



反馈

- 1.0 Android基础入门教程
 - 1.0.1 2015年最新Android基...
 - 1.1 背景相关与系统架构分析
 - 1.2 开发环境搭建
 - 1.2.1 使用Eclipse + ADT + S...
 - 1.2.2 使用Android Studio开...
 - 1.3 SDK更新不了问题解决
 - 1.4 Genymotion模拟器安装
 - 1.5.1 Git使用教程之本地仓...
 - 1.5.2 Git之使用GitHub搭建...
 - 1.6 .9(九妹)图片怎么玩
 - 1.7 界面原型设计
 - 1.8 工程相关解析(各种文件...
 - 1.9 Android程序签名打包
 - 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
 - 2.2.1 LinearLayout(线性布局)
 - 2.2.2 RelativeLayout(相对布...
 - 2.2.3 TableLayout(表格布局)
 - 2.2.4 FrameLayout(帧布局)
 - 2.2.5 GridLayout(网格布局)
 - 2.2.6 AbsoluteLayout(绝对...
 - 2.3.1 TextView(文本框)详解
 - 2.3.2 EditText(输入框)详解
 - 2.3.3 Button(按钮)与ImageB...
 - 2.3.4 ImageView(图像视图)
 - 2.3.5.RadioButton(单选按钮...
 - 2.3.6 开关按钮ToggleButton...
 - 2.3.7 ProgressBar(进度条)
 - 2.3.8 SeekBar(拖动条)
 - 2.3.9 RatingBar(星级评分条)
 - 2.4.1 ScrollView(滚动条)

是吧，inflate()每次都要加载一次xml，其实这个convertView是系统提供给我们的可供服用的View 的缓存对象，那就坐下判断咯，修改下，优化后的代码：

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if(convertView == null){
        convertView = LayoutInflater.from(mContext).inflate(R.layout.item_list_animal,parent,false);
    }

    ImageView img_icon = (ImageView) convertView.findViewById(R.id.img_icon);
    TextView txt_aName = (TextView) convertView.findViewById(R.id.txt_aName);
    TextView txt_aSpeak = (TextView) convertView.findViewById(R.id.txt_aSpeak);

    img_icon.setBackgroundResource(mData.get(position).getIcon());
    txt_aName.setText(mData.get(position).getName());
    txt_aSpeak.setText(mData.get(position).getSpeak());
    return convertView;
}
```

2.ViewHolder重用组件

嘿嘿，getView()会被调用多次，那么findViewById不一样得调用多次，而我们的ListView的Item 一般都是一样的布局，我们可以对这里在优化下，我们可以自己定义一个ViewHolder类来对这一部分 进行性能优化！修改后的代码如下：

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder = null;
    if(convertView == null){
        convertView = LayoutInflater.from(mContext).inflate(R.layout.item_list_animal,parent,false);
        holder = new ViewHolder();
        holder.img_icon = (ImageView) convertView.findViewById(R.id.img_icon);
        holder.txt_aName = (TextView) convertView.findViewById(R.id.txt_aName);
        holder.txt_aSpeak = (TextView) convertView.findViewById(R.id.txt_aSpeak);
        convertView.setTag(holder); //将Holder存储到convertView
    }
    return convertView;
}
```

2.4.2 Date & Time组件(上)

2.4.3 Date & Time组件(下)

2.4.4 Adapter基础讲解

2.4.5 ListView简单实用

2.4.6 BaseAdapter优化

2.4.7 ListView的焦点问题

2.4.8 ListView之checkbox错...

2.4.9 ListView的数据更新问题

2.5.0 构建一个可复用的自定义...

2.5.1 ListView Item多布局的...

2.5.2 GridView(网格视图)的...

2.5.3 Spinner(列表选项框)...

2.5.4 AutoCompleteTextView...

2.5.5 ExpandableListView(...

2.5.6 ViewPager(翻转视图)...

2.5.7 Toast(吐司)的基本使用

2.5.8 Notification(状态栏通...

2.5.9 AlertDialog(对话框)详解

2.6.0 其他几种常用对话框基...

2.6.1 PopupWindow(悬浮框)...

2.6.2 菜单(Menu)

2.6.3 ViewPager的简单使用

2.6.4 DrawerLayout(官方侧...

3.1.1 基于监听的事件处理机制

3.2 基于回调的事件处理机制

3.3 Handler消息传递机制浅析

3.4 TouchListener PK OnTo...

3.5 监听EditText的内容变化

3.6 响应系统设置的事件(Co...

3.7 AsyncTask异步任务

3.8 Gestures(手势)

4.1.1 Activity初学乍练

4.1.2 Activity初窥门径

4.1.3 Activity登堂入室

4.2.1 Service初涉

4.2.2 Service进阶

4.2.3 Service精通

4.3.1 BroadcastReceiver牛...

4.3.2 BroadcastReceiver庖...

4.4.1 ContentProvider初探

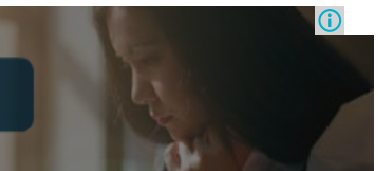
```
        }else{
            holder = (ViewHolder) convertView.getTag();
        }
        holder.img_icon.setBackgroundResource(mData.get(position).getIcon());
        holder.txt_aName.setText(mData.get(position).getName());
        holder.txt_aSpeak.setText(mData.get(position).getSpeak());
        return convertView;
    }

    static class ViewHolder{
        ImageView img_icon;
        TextView txt_aName;
        TextView txt_aSpeak;
    }
```

没错就是这么简单，你以后BaseAdapter照着这个模板写就对了，哈哈，另外这个修饰ViewHolder的 static，关于是否定义成静态，跟里面的对象数目是没有关系的，加静态是为了在多个地方使用这个Holder的时候，类只需加载一次，如果只是使用了一次，加不加也没所谓！——Berial(B神)原话~

本节小结：

好的，关于BaseAdapter的优化大概就上述的两种，非常简单，复用convertView以及自定义ViewHolder 减少findViewById()的调用~ 如果你有其他关于BaseAdapter优化的建议欢迎提出，谢谢~

[← 2.4.5 ListView简单实用](#)[2.4.7ListView的焦点问题 →](#)[4.4.2 ContentProvider再探...](#)[4.5.1 Intent的基本使用](#)[4.5.2 Intent之复杂数据的传递](#)[5.1 Fragment基本概念](#)[5.2.1 Fragment实例精讲—...](#)[5.2.2 Fragment实例精讲—...](#)[5.2.3 Fragment实例精讲—...](#)[5.2.4 Fragment实例精讲—...](#)[5.2.5 Fragment实例精讲—...](#)[6.1 数据存储与访问之——文...](#)[6.2 数据存储与访问之——S...](#)[6.3.1 数据存储与访问之——...](#)[6.3.2 数据存储与访问之——...](#)[7.1.1 Android网络编程要学...](#)[7.1.2 Android Http请求头与...](#)[7.1.3 Android HTTP请求方...](#)[7.1.4 Android HTTP请求方...](#)[7.2.1 Android XML数据解析](#)[7.2.2 Android JSON数据解析](#)[7.3.1 Android 文件上传](#)[7.3.2 Android 文件下载（1）](#)[7.3.3 Android 文件下载（2）](#)[7.4 Android 调用WebService](#)[7.5.1 WebView\(网页视图\)基...](#)[7.5.2 WebView和JavaScrip...](#)[7.5.3 Android 4.4后WebVie...](#)[7.6.1 Socket学习网络基础准备](#)[7.6.2 基于TCP协议的Socket...](#)[7.6.3 基于TCP协议的Socket...](#)[7.6.4 基于UDP协议的Socket...](#)[8.1.1 Android中的13种Draw...](#)[8.1.2 Android中的13种Draw...](#)[8.1.3 Android中的13种Draw...](#)[8.2.1 Bitmap\(位图\)全解析 P...](#)[8.2.2 Bitmap引起的OOM问题](#)[8.3.1 三个绘图工具类详解](#)[8.3.2 绘图类实战示例](#)

- 8.3.3 Paint API之—— Mask...
- 8.3.4 Paint API之—— Xferm...
- 8.3.5 Paint API之—— Xferm...
- 8.3.6 Paint API之—— Xferm...
- 8.3.7 Paint API之—— Xferm...
- 8.3.8 Paint API之—— Xferm...
- 8.3.9 Paint API之—— Color...
- 8.3.10 Paint API之—— Colo...
- 8.3.11 Paint API之—— Colo...
- 8.3.12 Paint API之—— Path...
- 8.3.13 Paint API之—— Sha...
- 8.3.14 Paint几个枚举/常量值...
- 8.3.15 Paint API之——Type...
- 8.3.16 Canvas API详解(Part 1)
- 8.3.17 Canvas API详解(Part...
- 8.3.18 Canvas API详解(Part...
- 8.4.1 Android动画合集之帧...
- 8.4.2 Android动画合集之补...
- 8.4.3 Android动画合集之属...
- 8.4.4 Android动画合集之属...
- 9.1 使用SoundPool播放音...
- 9.2 MediaPlayer播放音频与...
- 9.3 使用Camera拍照
- 9.4 使用MediaRecord录音
- 10.1 TelephonyManager(电...
- 10.2 SmsManager(短信管理...
- 10.3 AudioManager(音频管...
- 10.4 Vibrator(振动器)
- 10.5 AlarmManager(闹钟服务)
- 10.6 PowerManager(电源服...
- 10.7 WindowManager(窗口...
- 10.8 LayoutInflater(布局服务)
- 10.9 WallpaperManager(壁...
- 10.10 传感器专题(1)——相...
- 10.11 传感器专题(2)——方...
- 10.12 传感器专题(3)——加...
- 10.12 传感器专题(4)——其...
- 10.14 Android GPS初涉
- 11.0 《2015最新Android基...

<div>在线实例</div> <div><div>· HTML 实例</div><div>· CSS 实例</div><div>· JavaScript 实例</div><div>· Ajax 实例</div><div>· jQuery 实例</div><div>· XML 实例</div><div>· Java 实例</div></div>		<div>字符集&工具</div> <div><div>· HTML 字符集设置</div><div>· HTML ASCII 字符集</div><div>· HTML ISO-8859-1</div><div>· HTML 实体符号</div><div>· HTML 拾色器</div><div>· JSON 格式化工具</div></div>	<div>最新更新</div> <div><div>· PHP接收 json , 并...</div><div>· Foundation CSS ...</div><div>· Foundation CSS ...</div><div>· Foundation 图标...</div><div>· Foundation 网...</div><div>· Foundation 块...</div><div>· Foundation 网...</div></div>	<div>站点信息</div> <div><div>· 意见反馈</div><div>· 免责声明</div><div>· 关于我们</div><div>· 文章归档</div></div>
		<div>关注微信</div> <div></div> <div>Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1</div>		