

2.4.9 ListView的数据更新问题

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

本节引言：

我们前面已经学习了ListView的一些基本用法咧，但是细心的你可能发现了，我们的数据 一开始定义好的，都是静态的，但是实际开发中，我们的数据往往都是动态变化的，比如 我增删该了某一列，那么列表显示的数据也应该进行同步的更新，那么本节我们就来探讨下 ListView数据更新的问题，包括全部更新，以及更新其中的一项，那



么开始本节内容！~

1.先写个正常的demo先

好的，先写个正常的Demo先，等下我们再慢慢调：

entity类：[Data.java](#)：

```
/**
 * Created by Jay on 2015/9/21 0021.
 */
public class Data {
    private int imgId;
    private String content;

    public Data() {}

    public Data(int imgId, String content) {
        this.imgId = imgId;
        this.content = content;
    }

    public int getImgId() {
        return imgId;
    }

    public String getContent() {
        return content;
    }
}
```

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)

```
    }

    public void setImgId(int imgId) {
        this.imgId = imgId;
    }

    public void setContent(String content) {
        this.content = content;
    }
}
```

Activity布局以及列表项布局：

activity_main.xml：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/list_one"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

item_list.xml：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/img_icon"
        android:layout_width="56dp"
        android:layout_height="56dp" />

    <TextView
        android:id="@+id/txt_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="10dp"
        android:textSize="18sp" />

</LinearLayout>
```

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...

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 AutoCompleteTextVie...

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

自定义BaseAdapter的实现：MyAdapter.java：

```
/**
 * Created by Jay on 2015/9/21 0021.
 */
public class MyAdapter extends BaseAdapter {

    private Context mContext;
    private LinkedList<Data> mData;

    public MyAdapter() {}

    public MyAdapter(LinkedList<Data> mData, Context mContext)
    {
        this.mData = mData;
        this.mContext = mContext;
    }

    @Override
    public int getCount() {
        return mData.size();
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @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, parent, false);
            holder = new ViewHolder();
            holder.img_icon = (ImageView) convertView.findViewById(R.id.img_icon);
            holder.txt_content = (TextView) convertView.findViewById(R.id.txt_content);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }
        holder.img_icon.setImageResource(mData.get(position).getImgId());
        holder.txt_content.setText(mData.get(position).getConte
```

[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JavaScript...](#)[7.5.3 Android 4.4后WebVie...](#)[7.5.4 WebView文件下载](#)[7.5.5 WebView缓存问题](#)[7.5.6 WebView处理网页返...](#)[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Socke...](#)[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](#)

```

nt());

    return convertView;
}

private class ViewHolder{
    ImageView img_icon;
    TextView txt_content;
}
}

```

MainActivity.java的编写：

```

public class MainActivity extends AppCompatActivity {

    private ListView list_one;
    private MyAdapter mAdapter = null;
    private List<Data> mData = null;
    private Context mContext = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mContext = MainActivity.this;
        bindViews();
        mData = new LinkedList<Data>();
        mAdapter = new MyAdapter((LinkedList<Data>) mData, mContext);
        list_one.setAdapter(mAdapter);
    }

    private void bindViews(){
        list_one = (ListView) findViewById(R.id.list_one);
    }
}

```

可以运行，运行后发现我们的页面并没有任何的数据，白茫茫的一片，这样的用户体验并不好，我们可以通过调用ListView的一个**setEmptyView(View)**的方法，当ListView数据为空的时候，显示一个对应的View，另外发现这个方法很奇葩，动态添加的View，竟然无效，只能在ListView所在的布局文件中添加当ListView无数据时，想显示的View，另外用这个setEmptyView设置后的View，加载的时候竟然不会显示出来，好灵异....比如这里的是没有数据时显示一个没有数据的TextView，部分代码如下：

```

<TextView
    android:id="@+id/txt_empty"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

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...

```
android:layout_gravity="center"
android:textSize="15pt"
android:textColor="#000000"/>
```

```
txt_empty = (TextView) findViewById(R.id.txt_empty);
txt_empty.setText("暂无数据~");
list_one.setEmptyView(txt_empty);
```

当然除了这种方法外我们还可以定义一个与ListView一样大小位置的布局，然后设置，`android:visibility="gone"`，在Java代码中对mData集合的size进行判断，如果`==0`，说明没数据，让这个布局显示出来，当有数据的时候让这个布局隐藏~

2.添加一条记录

好的，我们弄个添加按钮，没按一次添加一条记录哈~

运行效果图：



代码实现

在我们自定义的BaseAdapter中定义一个方法，方法内容如下：

```
public void add(Data data) {
    if (mData == null) {
        mData = new LinkedList<>();
    }
    mData.add(data);
    notifyDataSetChanged();
}
```

然后布局自己加个按钮，然后设置下事件，代码如下：

```
private Button btn_add;
```

```
btn_add = (Button) findViewById(R.id.btn_add);
btn_add.setOnClickListener(this);

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btn_add:
            mAdapter.add(new Data(R.mipmap.ic_icon_qitao, "给猪
            哥跪了~~~ x " + flag));
            flag++;
            break;
    }
}
```

嘿嘿，成了，添加数据就这么简单~，如果你想插入到特定位置，也行，我们Adapter类里，再另外 写一个方法：

```
//往特定位置，添加一个元素
public void add(int position, Data data) {
    if (mData == null) {
        mData = new LinkedList<>();
    }
    mData.add(position, data);
    notifyDataSetChanged();
}
```

然后加个按钮，写个事件：

```
private Button btn_add2;

btn_add2 = (Button) findViewById(R.id.btn_add2);
btn_add2.setOnClickListener(this);

case R.id.btn_add2:
    //position从0开始算的
    mAdapter.add(4, new Data(R.mipmap.ic_icon_qitao, "给猪哥跪了~~~ x
    " + flag));
    break;
```

运行效果图：



可以看到我们的第九项插入到了第五个位置~

3.删除某一项

同样的，我们写两个方法，一个直接删对象，一个根据游标来删：

```
public void remove(Data data) {  
    if(mData != null) {  
        mData.remove(data);  
    }  
    notifyDataSetChanged();  
}  
  
public void remove(int position) {  
    if(mData != null) {  
        mData.remove(position);  
    }  
    notifyDataSetChanged();  
}
```

然后加两个Button，调用下这两个方法：

```
case R.id.btn_remove:  
    mAdapter.remove(mData_5);  
    break;  
case R.id.btn_remove2:  
    mAdapter.remove(2);  
    break;
```

运行效果图：



从图中我们可以看到，第五项被移除了，然后点击游标删除数据，一直删的是第三项！

4.移除所有的记录：

这个更加简单，直接调用clear方法即可！方法代码如下：

```
public void clear() {  
    if (mData != null) {  
        mData.clear();  
    }  
    notifyDataSetChanged();  
}
```

5.更新某一个记录

细心的你应该发现了，进行了数据修改操作后，都会调用一个
notifyDataSetChanged(); **一开始我以为：**

notifyDataSetChanged()会把界面上现实的item都重绘一次，这样会影响ui性能吧，如果数据量 很大，但是我改变一项就要重新绘制所有的item，这肯定不合理是吧！于是乎，我用了一个傻办法 来修改某个Item中控件的值，我在Java代码中写了这样一段代码：

```
private void updateListItem(int postion, Data mData) {  
    int visiblePosition = list_one.getFirstVisiblePosition();  
    View v = list_one.getChildAt(postion - visiblePosition);  
    ImageView img = (ImageView) v.findViewById(R.id.img_icon);  
    TextView tv = (TextView) v.findViewById(R.id.txt_content);  
    img.setImageResource(mData.getImgId());  
}
```



```
tv.setText(mData.getContent());  
}
```

后来和群里的朋友讨论了下，发现自己错了：

notifyDataSetChanged()方法会判断是否需要重新渲染，如果当前item没有必要重新渲染 是不会重新渲染的，如果某个Item的状态发生改变，都会导致View的重绘，而重绘的并不是 所有的Item，而是View状态发生变化的那个Item！所以我们直接notifyDataSetChanged()方法 即可，当然知道多一个上面的方法也没什么~

代码下载：

[ListViewDemo3.zip](#)

本节小结：

好的，本节跟大家讲述了ListView中数据更新的实现，当然不止ListView，其他的Adapter 类控件都可以调用这些方法来完成数据更新~就说这么多吧~谢谢

← 2.4.8 ListView之checkbox错位问题解决


2.5.0 构建一个可复用的自定义BaseAdapter →

大家都在玩手机而我却用手机学英语

邮箱订阅《每日英语》

每天坚持五分钟, 随时随地学英语！

立即行动



反馈

在线实例

- HTML 实例
- CSS 实例
- JavaScript 实例
- Ajax 实例
- jQuery 实例
- XML 实例
- Java 实例

字符集&工具

- HTML 字符集设置
- HTML ASCII 字符集
- HTML ISO-8859-1
- HTML 实体符号
- HTML 拾色器
- JSON 格式化工具

最新更新

- PHP接收 json，并...
- Foundation CSS ...
- Foundation CSS ...
- Foundation 图标...
- Foundation 网格...
- Foundation 块状...
- Foundation 网格...

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

关注微信



Copyright © 2013-2015 菜鸟教程
runoob.com All Rights Reserved.
备案号：闽ICP备15012807号-1

