

2.4.5 ListView简单实用

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

本节引言：

本节我们来继续学习没有讲完的UI控件部分，回顾上一节，我们介绍了Adapter适配器的概念，然后学习了三个最简单的适配器的使用：ArrayAdapter，SimpleAdapter和SimpleCursorAdapter，而本节给大家讲解的是第一个 需搭配Adapter使用的UI控件：ListView，不过在版本中被RecyclerView这个新控件替换掉了！列表作为最常用的控件之一，还是有必要好好学习的，本节以一个初学者的角度来学习 ListView，ListView的属性，以及BaseAdapter简单定义，至于ListView优化这些，我们一步步来~莫急！

1.自定义BaseAdapter，然后绑定ListView的最简单例子

先看看我们要实现的效果图：

- 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)



一个很简单的ListView，自己写下Item，然后加载点数据这样~ 下面贴下关键代码：

Animal.java:

```
/**
 * Created by Jay on 2015/9/18 0018.
 */
public class Animal {
    private String aName;
    private String aSpeak;
    private int aIcon;

    public Animal() {
    }

    public Animal(String aName, String aSpeak, int aIcon) {
        this.aName = aName;
        this.aSpeak = aSpeak;
        this.aIcon = aIcon;
    }

    public String getaName() {
        return aName;
    }

    public String getaSpeak() {
```

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 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

```

        return aSpeak;
    }

    public int getaIcon() {
        return aIcon;
    }

    public void setaName(String aName) {
        this.aName = aName;
    }

    public void setaSpeak(String aSpeak) {
        this.aSpeak = aSpeak;
    }

    public void setaIcon(int aIcon) {
        this.aIcon = aIcon;
    }
}

```

AnimalAdapter.java : 自定义的BaseAdapter :

```

/**
 * Created by Jay on 2015/9/18 0018.
 */
public class AnimalAdapter extends BaseAdapter {

    private LinkedList<Animal> mData;
    private Context mContext;

    public AnimalAdapter(LinkedList<Animal> mData, Context mCon
text) {
        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, ViewGro

```

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基本概述

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...

7.5.4 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

```

up parent) {
    convertView = LayoutInflater.from(mContext).inflate(R.l
ayout.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).geta
Icon());
    txt_aName.setText(mData.get(position).getName());
    txt_aSpeak.setText(mData.get(position).getaSpeak());
    return convertView;
}
}

```

最后是MainActivity.java：

```

public class MainActivity extends AppCompatActivity {

    private List<Animal> mData = null;
    private Context mContext;
    private AnimalAdapter mAdapter = null;
    private ListView list_animal;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mContext = MainActivity.this;
        list_animal = (ListView) findViewById(R.id.list_animal)
;

        mData = new LinkedList<Animal>();
        mData.add(new Animal("狗说", "你是狗么?", R.mipmap.ic_ic
on_dog));
        mData.add(new Animal("牛说", "你是牛么?", R.mipmap.ic_ic
on_cow));
        mData.add(new Animal("鸭说", "你是鸭么?", R.mipmap.ic_ic
on_duck));
        mData.add(new Animal("鱼说", "你是鱼么?", R.mipmap.ic_ic
on_fish));
        mData.add(new Animal("马说", "你是马么?", R.mipmap.ic_ic
on_horse));
        mAdapter = new AnimalAdapter((LinkedList<Animal>) mData
, mContext);
        list_animal.setAdapter(mAdapter);
    }

}

```

好的，自定义BaseAdapter以及完成数据绑定就是这么简单~

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...

别问我拿示例的代码，刚开始学就会写出这些代码，我只是演示下流程，让大家熟悉 熟悉而已~另外，也是为下面的属性验证做准备~

2.表头表尾分割线的设置：

listview作为一个列表控件，他和普通的列表一样，可以自己设置表头与表尾：以及分割线，可供我们设置的属性如下：

footerDividersEnabled：是否在footerView(表尾)前绘制一个分隔条,默认为true

headerDividersEnabled:是否在headerView(表尾)前绘制一个分隔条,默认为true

divider:设置分隔条,可以用颜色分割,也可以用drawable资源分割

dividerHeight:设置分隔条的高度

翻遍了API发现并没有可以直接设置ListView表头或者表尾的属性，只能在Java中写代码 进行设置了，可供我们调用的方法如下：

addHeaderView(View v)：添加headView(表头),括号中的参数是一个View对象

addFooterView(View v)：添加footerView(表尾)，括号中的参数是一个View对象

addHeaderView(headView, null, false)：和前面的区别：设置Header是否可以被选中

addFooterView(View,view,false)：同上

对了，使用这个addHeaderView方法必须放在listview.setAdapter前面，否则会报错。

使用示例：

运行效果图：



代码实现：

先编写下表头与表尾的布局：

view_header.xml(表头),表尾一样，就不贴了：

```
<?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="vertical"
    android:gravity="center">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:textSize="18sp"
        android:text="表头"
        android:gravity="center"
        android:background="#43BBEB"
        android:textColor="#FFFFFF"/>
</LinearLayout>
```

MainActivity.java:

```
public class MainActivity extends AppCompatActivity implements
AdapterView.OnItemClickListener{

    private List<Animal> mData = null;
    private Context mContext;
    private AnimalAdapter mAdapter = null;
    private ListView list_animal;
    private LinearLayout ly_content;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mContext = MainActivity.this;
        list_animal = (ListView) findViewById(R.id.list_animal)
;

        //动态加载顶部View和底部View
        final LayoutInflater inflater = LayoutInflater.from(this);

        View headView = inflater.inflate(R.layout.view_header,
null, false);
        View footView = inflater.inflate(R.layout.view_footer,
null, false);

        mData = new LinkedList<Animal>();
        mData.add(new Animal("狗说", "你是狗么?", R.mipmap.ic_ic
on_dog));
        mData.add(new Animal("牛说", "你是牛么?", R.mipmap.ic_ic
```

```

on_cow));
        mData.add(new Animal("鸭说", "你是鸭么?", R.mipmap.ic_ic
on_duck));
        mData.add(new Animal("鱼说", "你是鱼么?", R.mipmap.ic_ic
on_fish));
        mData.add(new Animal("马说", "你是马么?", R.mipmap.ic_ic
on_horse));
        mAdapter = new AnimalAdapter((LinkedList<Animal>) mData
, mContext);
        //添加表头和表尾需要写在setAdapter方法调用之前!!!
        list_animal.addHeaderView(headView);
        list_animal.addFooterView(footView);

        list_animal.setAdapter(mAdapter);
        list_animal.setOnItemClickListener(this);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, i
nt position, long id) {
        Toast.makeText(mContext, "你点击了第" + position + "项", T
oast.LENGTH_SHORT).show();
    }
}

```

好的，代码还是比较简单的，从上面我们看出来一个要注意的问题，就是：

添加表头表尾后，我们发现positon是从表头开始算的，就是你添加的第一个数据本来的postion是 0，但是此时却变成了0，因为表头也算！！

3.列表从底部开始显示：stackFromBottom

如果你想让列表显示你列表的最下面的话，那么你可以使用这个属性，将stackFromBottom 属性设置为true即可，设置后的效果图如下：



4.设置点击颜色cacheColorHint

如果你为ListView设置了一个图片作为Background的话，当你拖动或者点击listView空白位置会发现 item都变成黑色了，这是时候我们可以通过这个**cacheColorHint**将颜色设置为透明:#00000000

5.隐藏滑动条

我们可以通过设置：`android:scrollbars="none"` 或者 `setVerticalScrollBarEnabled(true);` 解决这个问题！

本节小结：

好的，关于ListView的基本用法大概就这些，当然除了上述的这些属性外还有其他的，实际遇到再查查吧~这里知道如何去重写BaseAdapter和完成数据绑定就好，下节我们来教大家如何来优化这个BaseAdapter的编写~



最省心的支付SDK
3行代码实现APP/网页支付

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