**安卓新闻客户端**

**——**174340-周泓镔

目录

[1设计目标 2](#_Toc498891983)

[1.1 实验背景 2](#_Toc498891984)

[1.2 实验目的 2](#_Toc498891985)

[2对所需技术的分析 3](#_Toc498891986)

[2.1 LinearLayout 3](#_Toc498891987)

[2.2 ListView 3](#_Toc498891988)

[2.3 XListView 3](#_Toc498891989)

[2.4 WebView 3](#_Toc498891990)

[2.5布局嵌套 3](#_Toc498891991)

[2.6 GSON 4](#_Toc498891992)

[2.7从网络获取图片 4](#_Toc498891993)

[3设计与实现（源码及注释） 5](#_Toc498891994)

[3.1主活动的java文件”MainActivity” 5](#_Toc498891995)

[3.2主活动对应的xml布局文件”main\_layout”: 7](#_Toc498891996)

[3.3 XListView中的每条条目对应的xml布局文件”listview\_layout”: 8](#_Toc498891997)

[4演示截图（可正常运行的安卓程序） 10](#_Toc498891998)

[5 备注信息（如开发环境的版本、apk文件大小等） 12](#_Toc498891999)

[5.1 开发环境版本 12](#_Toc498892000)

[5.2 apk大小 12](#_Toc498892001)

# 1设计目标

## 1.1 实验背景

随着时代的发展，人们对新闻的获得方法在变得越来越方便，从手机上获得新闻已经是一种必不可少的方法。此次的课程设计就是设计一个简单的安卓手机的新闻客户端app。

## 1.2 实验目的

1. 了解XListView或其他增强型ListView的使用方法，实现上拉刷新，下拉加载更多的功能
2. 了解JSON数据格式
3. 如何使用GSON或者Jackson库，对JSON格式的数据进行解析，并且序列化成业务对象
4. 构建业务对象
5. 使用Adapter类，细致化实现ListView的数据绑定
6. 了解WebView在呈现新闻等html page的方法，了解WebView中支持Javascript的使用方法
7. 加强对布局文件美观效果的逐步追求，了解UI细节处理需要投入更多的精力，需要全面理解LinearLayout,RelativeLayout等多种布局的综合应用

# 2对所需技术的分析

## 2.1 LinearLayout

LinearLayout又称作线性布局，是一种非常常用的布局。在此项目中的布局基本上都是线性布局。正如它的名字描述的一样，这个布局会将它所包含的控件在线性方向上依次排列，它使用方便，可以轻易的对控件进行较复杂的布局。

## 2.2 ListView

由于手机屏幕空间比较有限，能够一次能够性在屏幕上显示的内容并不多，当我们的程序中有大量的数据需要展示的时候，就可以借助ListView来实现。ListView允许用户通过手指上下滑动的方式将屏幕外的数据滚动到屏幕内，同时屏幕上原有的数据则会滚动出屏幕。在新闻客户端中，新闻可以说是无穷多的，所以可以借助ListView来实现用户对新闻的浏览。

## 2.3 XListView

这是一个第三方的框架，它是基于ListView而做的。它在ListView的基础上，实现了上拉加载和下拉刷新的功能。

## 2.4 WebView

现在是互联网是时代，是个app都需要联网，因此难免会遇到需要在应用程序里展示一些网页的需求。如果需求里又明确要求加载和显示网页不可以使用浏览器，这时WebView就可以派上用场了。但在这次项目里，WebView主要用来显示用HTML写的新闻详情页。

## 2.5布局嵌套

单纯的使用一个布局是不可能写出复杂、美观的界面的，这时候就需要使用嵌套的布局来解决问题。在此次布局中，总共使用了两层的LinearLayout嵌套布局来实现ListView和ImageView的布局和ListView内图片和文字的布局

## 2.6 GSON

在此次项目中，因为数据是通过JSON格式输入的，所以使用了Google提供的GSON来解析JSON数据。它的方便之处在于，可以将JSON格式的字符串自动映射成一个对象，即使是解析JSON数组也只需要一条语句，使用方便。

## 2.7从网络获取图片

在此项目中，从网络获取图片的方法是，先新建一个子线程，在子线程中使用URL类的openConnection（）方法建立连接，然后用HttpURLConnection类的getInputStream方法的到流，再从流中读出数据，构建成Bitmap对象，然后通过Handler的对象把图片传到主线中，然后在主线程中使用图片。

# 3设计与实现（源码及注释）

## 3.1主活动的java文件”MainActivity”

Package …

Import …

**public class** MainActivity **extends** AppCompatActivity **implements** XListView.IXListViewListener{  
 **private** XListView **listView**;  
 **private** List<IData> **idatas**;  
 **private** List<IData> **idatasPart** = **new** ArrayList<>();  
 **private** IDataAdapter **adapter**;  
 **private** Handler **mHandler**;  
 **private int start** = 0;  
 **private int refreshCnt** = 0;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***main\_layout***);  
  
 ImageView bigImage = (ImageView) findViewById(R.id.***bigimage***);  
 bigImage.setImageResource(R.drawable.***pp***);  
  
 String foodJson = *getJson*(MainActivity.**this**, **"get\_data.json"**);  
 Gson gson = **new** Gson();  
 *//将json内容赋入idatas中* **idatas** = gson.fromJson(foodJson, **new** TypeToken<List<IData>>() {}.getType());  
 *//将第一页要显示的内容赋入idatasPart中* geneItems();  
 **adapter** = **new** IDataAdapter(MainActivity.**this**, R.layout.***listview\_layout***, **idatasPart**);  
 **listView** = (XListView) findViewById(R.id.***xlist\_view***);  
 **listView**.setPullLoadEnable(**true**);*//上拉加载开* **listView**.setPullRefreshEnable(**true**); *//下拉刷新开* **listView**.setAdapter(**adapter**);  
 **listView**.setXListViewListener(**this**);*//实现接口* **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> adapterView, View view, **int** position, **long** id) {  
 IData idata = **idatasPart**.get(position);  
 WebViewl.*actionStart*(MainActivity.**this**);  
 } *//点击列表后的反应* });  
 **mHandler** = **new** Handler();  
  
 }  
  
 *//取出json文件内容并转化为String类型返回* **public static** String getJson(Context context, String fileName) {  
 StringBuilder stringBUilder = **new** StringBuilder();  
 AssetManager assetManager = context.getAssets();  
 **try** {  
 BufferedReader bufferedReader = **new** BufferedReader(**new** InputStreamReader(assetManager.open(fileName),**"utf-8"**));  
 String line;  
 **while** ((line = bufferedReader.readLine()) != **null**) {  
 stringBUilder.append(line);  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **return** stringBUilder.toString();  
 }  
  
 *//给idatasPart里加内容* **private void** geneItems() {  
 **for** (**int** i = 0; i<4&&**start**<**idatas**.size(); ++i) {  
 **idatasPart**.add(**idatas**.get(**start**++));  
 }  
 }  
  
 *//更新画面* **private void** onLoad() {  
 **listView**.stopRefresh();  
 **listView**.stopLoadMore();  
 **listView**.setRefreshTime(**"刚刚"**);  
 }  
  
 @Override  
 *//下拉刷新的动作* **public void** onRefresh() {  
 **mHandler**.postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 IData idataf = **new** IData();  
 idataf.setTitle(**"我是新的新闻 "**+(++**refreshCnt**));  
 idataf.setImageUrl(**idatas**.get(0).getImageUrl());  
 **idatasPart**.add(0, idataf);  
 **adapter**.notifyDataSetChanged();  
 onLoad();  
 }  
 }, 2000);  
 }  
  
 @Override  
 *//上拉加载的动作* **public void** onLoadMore() {  
 **mHandler**.postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 geneItems();  
 **adapter**.notifyDataSetChanged();  
 onLoad();  
 }  
 }, 2000);  
 }  
}

## 3.2主活动对应的xml布局文件”main\_layout”:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**LinearLayout  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="5"**>  
 <**ImageView  
 android:id="@+id/bigimage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="9"  
 android:orientation="vertical"**>  
  
 <**com.example.adminstration.homework.XListView  
 android:id="@+id/xlist\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:cacheColorHint="#00000000"**>  
 </**com.example.adminstration.homework.XListView**>  
  
 </**LinearLayout**>  
  
</**LinearLayout**>

## 3.3 XListView中的每条条目对应的xml布局文件”listview\_layout”:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**ImageView  
 android:layout\_gravity="center\_vertical"  
 android:id="@+id/list\_image"  
 android:layout\_width="70dp"  
 android:layout\_height="70dp"  
 android:paddingTop="5dp"  
 android:paddingBottom="5dp"  
 android:paddingLeft="3dp"  
 android:paddingRight="5dp"** />  
 <**TextView  
 android:layout\_gravity="center\_vertical"  
 android:id="@+id/list\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:paddingLeft="3dp"** />  
</**LinearLayout**>

# 4演示截图（可正常运行的安卓程序）



图1 初始界面 图2 上拉加载



图3 下拉加载 图4 下拉加载后

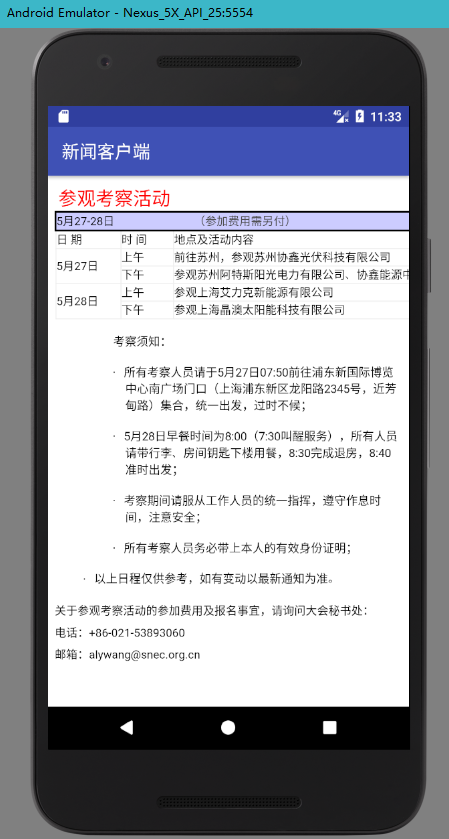


图5 新闻详情页

# 5 备注信息（如开发环境的版本、apk文件大小等）

## 5.1 开发环境版本

Android SDK：API 25

IDE：Android Studio 3.0

## 5.2 apk大小

1854KB