Step 2: 连接网络，查询指定城市的天气信息（JSON格式），把结果显示在文本域中

目录

[目标： 1](#_Toc449454132)

[步骤： 2](#_Toc449454133)

[1、为APP增加网络访问权限 2](#_Toc449454134)

[2、新建内部类，用于访问网络获取数据 2](#_Toc449454135)

[3、编写Go按钮按下时的处理代码 3](#_Toc449454136)

[4、编写HttpTools 3](#_Toc449454137)

[5、添加showError方法 5](#_Toc449454138)

[执行流程说明 5](#_Toc449454139)

## 目标：

连接网络，查询指定城市的天气信息（JSON格式），把结果显示在文本域中。

连接网络和获取数据的方法请参见:“5 案例 - 获取天气预报信息.pptx”



## 步骤：

### 1、为APP增加网络访问权限

打开AndroidManifest.xml文件， 添加如下行：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="cn.edu.uoh.cs.weatherforecast"**>   <**uses-permission android:name="android.permission.INTERNET"** />   … |

### 2、新建内部类，用于访问网络获取数据

在Android中不能在主线程中访问网络，必须使用线程或异步任务，为此在MainActivity中建立类NetTask，用于访问网络。NetTask继承AsyncTask(Android 中的异步抽象类)，具体代码如下：

|  |
| --- |
| **class** NetTask **extends** AsyncTask<String, Void, String> {  IOException **exception** = **null**;  @Override  **protected** String doInBackground(String... params) {  **try** {  *// 获取调用execute方法时传入的参数：城市名* String cityName = params[0];  *// 获取天气信息，得到的是Json字符串* String json = HttpTools.*getWeatherInfo*(cityName);  **return** json;  } **catch** (IOException e) {  *// 如果有异常，记录* **exception** = e;  **return null**;  }  }   @Override  **protected void** onPostExecute(String result) {  **if** (**exception** == **null**) {  *// 显示获取的信息* TextView txtInfo = (TextView) findViewById(R.id.***txtInfo***);  txtInfo.setText(result);  } **else** {  *// 如果有异常，显示错误消息* showError(**"获取天气信息失败："** + **exception**.getMessage());  }  } }; |

类AsyncTask<String, Void, String>，中<String, Void, String>的分别表示，输入参数是String类型，无任务的进度信息，结果类型是String

### 3、编写Go按钮按下时的处理代码

编写方法用于响应Go按钮按下事件

1、编写方法

|  |
| --- |
| **public void** fetchWeatherInfo(View view) {  *// 取得输入框中的城市名称* EditText edtCity = (EditText) findViewById(R.id.***edtCity***);  String cityName = edtCity.getText().toString();  *// 调用异步任务类，访问网络，获取数据* NetTask task = **new** NetTask();  task.execute(cityName); } |

2、修改content\_main.xml, 为按钮设置单击方法

|  |
| --- |
| <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Go"  android:id="@+id/btnExec"  android:onClick="fetchWeatherInfo"** /> |

### 4、编写HttpTools

编写获取天气预报的类

|  |
| --- |
| */\*\*  \* 获取天气预报信息的类，  \* <p/>  \* Created by yan on 2016/4/12.  \*/* **public class** HttpTools {  */\*\*  \* 获取天气预报信息的方法，调用此方法可以获取指定城市的天气信息。  \* 用这个地区获取：http://wthrcdn.etouch.cn/weather\_mini?city=城市名  \*  \** ***@param cityName*** *城市名称  \** ***@return*** *获取到的字符串JSON格式。  \*/* **public static** String getWeatherInfo(String cityName) **throws** IOException {  *// 获取天气的URL* URL url = **new** URL(**"http://wthrcdn.etouch.cn/weather\_mini?city="** + cityName);  URLConnection urlc = url.openConnection();  InputStream is = urlc.getInputStream();  *// 检查获取的结果是否是被压缩过的* **if** (**"gzip"**.equals(urlc.getContentEncoding())) {  *// 如果是用GZIPInputStream包裹节压缩* is = **new** GZIPInputStream(is);  }  String json = *readToEnd*(is);  **return** json;  }   */\*\*  \* 从流中读取所有内容，直到流结束，并转换为String格式。  \** ***@param is*** *输入流  \** ***@return*** *读取到字符串  \** ***@throws*** *IOException  \*/* **private static** String readToEnd(InputStream is) **throws** IOException {  **final int** len = 1024;  *// 定义读取的缓冲区* **byte** data[] = **new byte**[len];  *// 用ByteArrayOutputStream把byte[] 转换为String* ByteArrayOutputStream bos = **new** ByteArrayOutputStream();  **while** (**true**) {  **int** rlen = is.read(data);  **if** (rlen == -1) {  **break**;  }  bos.write(data, 0, rlen);  }  **return** bos.toString(**"UTF-8"**);  } } |

### 5、添加showError方法

编写方法用于响应Go按钮按下事件

|  |
| --- |
| */\*\*  \* 显示错误信息提示框  \*  \** ***@param msg*** *显示的错误消息  \*/* **public void** showError(String msg) {  **new** AlertDialog.Builder(**this**)  .setTitle(**"出错了"**)  .setMessage(msg)  .setPositiveButton(**"确定"**,  **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialoginterface, **int** i) {  }  }).show(); } |

## 执行流程说明

1、用户输入城市名称，单击Go按钮，进入方法fetchWeatherInfo

2、在方法fetchWeatherInfo中，获取城市名称cityName，创建异步任务（NetTask task = new NetTask()），并执行（task.execute(cityName)）

3、在异步任务类中，调用json = HttpTools.getWeatherInfo(cityName)，获取天气预报数据，当方法doInBackground执行完成后，系统自动调用方法onPostExecute，在方法onPostExecute中更新UI。