



中山大學
SUN YAT-SEN UNIVERSITY

《手机平台应用开发 (与 Google 共建) 实验》

实验九：网络防问&Web 服务开发 实验报告

学 院 名 称 : 数据科学与计算机学院

专 业 : 软件工程 (计应)

学 生 姓 名 : 张凯鑫

学 号 : 14331362

班 级 : 周三上午 4-5 节、周五下午 7-8 节

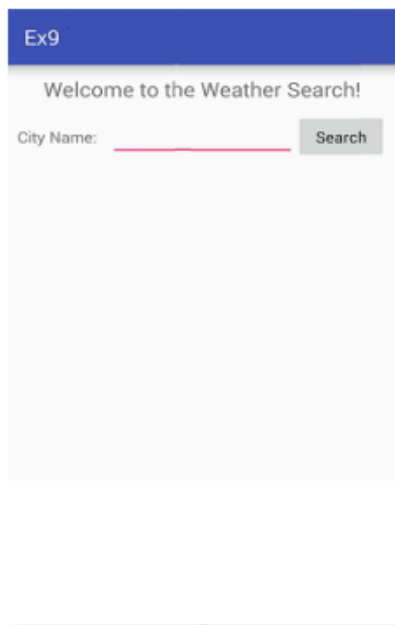
【实验目的】

1. 熟练使用 HttpURLConnection 访问 Webservice;
2. 熟悉使用多线程以及 Handler 更新 UI;
3. 熟悉使用 XmlPullParser 解析 xml 文档数据;
4. 了解 RecyclerView 控件的使用;

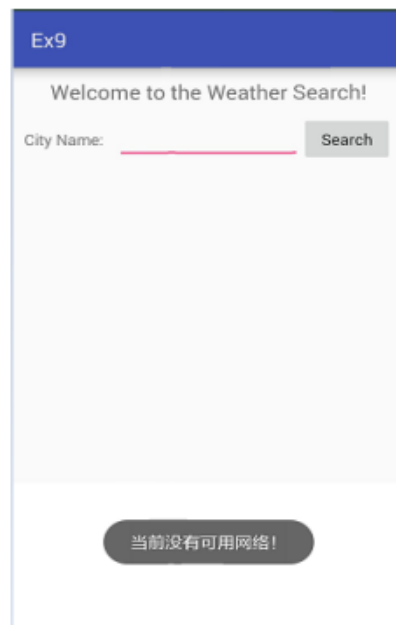
【实验内容】

实现一个简单的天气查询应用，具体要求如下：

1) 该界面为应用启动之后的主界面



2) 点击 Search 按钮后（若网络不可用）



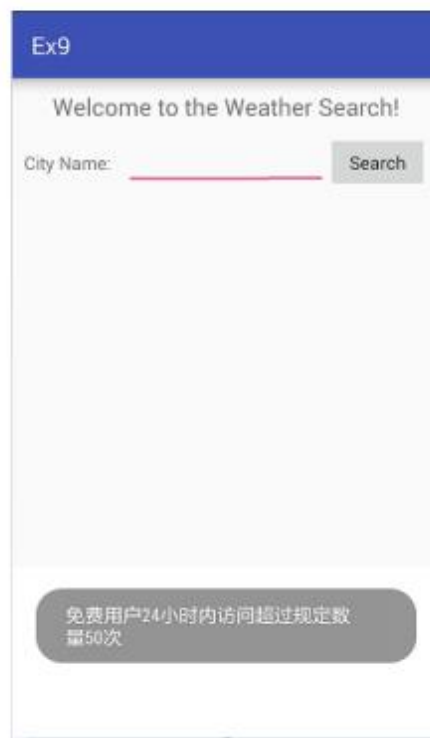
3) 输入城市名 Search（网络可用且城市名正确） 4) 输入城市名 Search（城市名不正确）



5) 快速点击按钮（二次查询<600ms）



6) 查询达到上限 50 次



【实验过程】

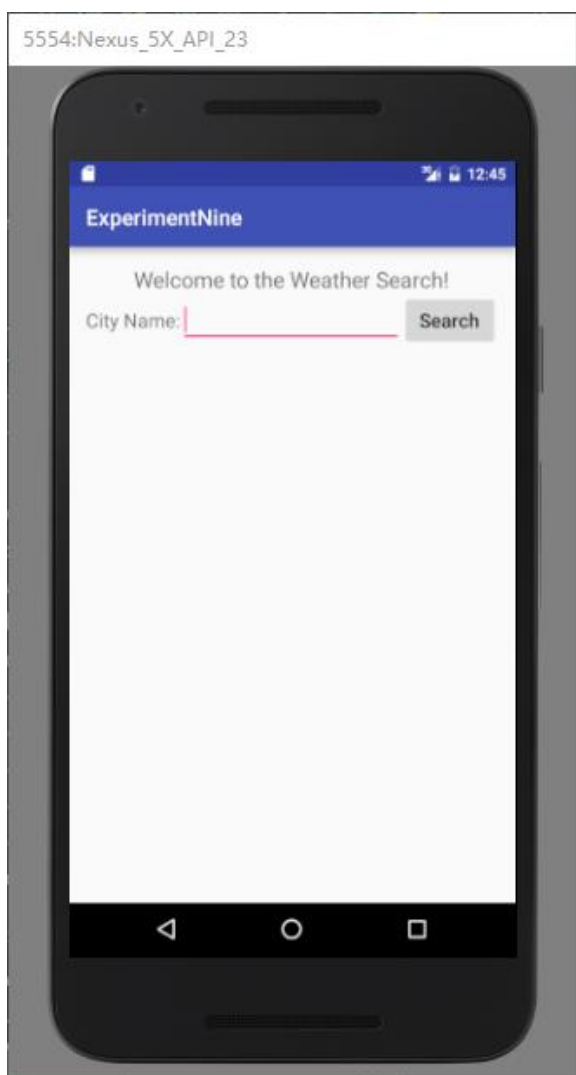
（以下过程为简单步骤介绍，具体实验效果与关键代码在“实验结果”中结合截图分析给出）

1. 创建新的 Android Studio 项目，命名为：ExperimentNine。
2. 按要求创建布局文件：activity_main.xml，添加控件 TextView、EditText、Button、ListView 等。界面控件的布局较复杂，总体为垂直方面上的 LinearLayout，包含 RelativeLayout 和 GridLayout 等。其中 RelativeLayout 用于布局输入的 EditText、Button 和少量查询结果的显示（查询地点、更新时间），GridLayout 用于布局气温的实况（温度、湿度、风力、温差、空气质量），ListView 显示各种指数，RecyclerView 用于显示近期五天的天气（后面再实现）。
3. 按要求创建布局文件：list_item.xml，作为 ListView 的布局。
4. 实现 MainActivity.java，主要是 Button 的点击事件处理。包括网络是否可访问的检查、URLConnection 访问 Webservice、多线程及 Handler 更新 UI、XmlPullParser 解析文档数据。
5. 导入 RecyclerView 的 jar 包，在布局文件中添加控件 RecyclerView，构造 Weather 类和 WeatherAdapter 类，在 MainActivity.java 文件中实现 RecyclerView 的添加和事件处理。
6. 运行并调整实验代码直到实验完成。

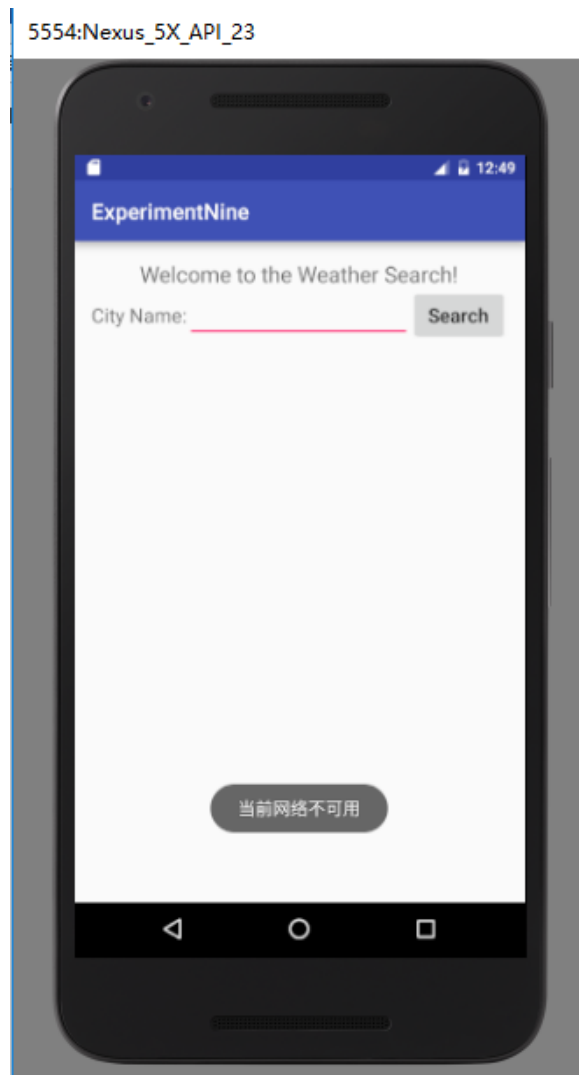
【实验结果】

实验效果图及部分代码分析如下（实验代码详见 lab9_code 文件夹）：

实验中，因为网站提供的免费账号不能查询，不得不注册一个用户，因此，免费用户的限制：“二次查询时间间隔小于 600ms”和“24 小时内访问量不超过 50 次”，无法得到实现，但实验代码中有所体现。



主界面



点击 Search 按键（网络不可用）

关键代码如下：

【“Search” 按键的事件处理】通过方法 `isNetworkAvailable(Activity activity)` 返回的 Boolean 值来判断当前网络是否可用，可用，则向网站发送请求；不可用，则用 Toast 提示当前网络不可用。具体如下：

```
Search.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isNetworkAvailable(MainActivity.this)) {
            postRequest();
        } else {
            Toast.makeText(MainActivity.this, "当前网络不可用", Toast.LENGTH_SHORT).show();
        }
    }
});
```

【用于判断网络是否可用】返回 true 或 false 表示可用或不可用。

```
public boolean isNetworkAvailable(Activity activity) {
    Context context = activity.getApplicationContext();
    ConnectivityManager connectivityManager =
    (ConnectivityManager)context.getSystemService(Context.CONNECTIVITY_SERVICE);
    if (connectivityManager == null) {
        return false;
    } else {
        NetworkInfo[] networkInfo = connectivityManager.getAllNetworkInfo();
        if (networkInfo != null && networkInfo.length > 0) {
            for (int i = 0; i < networkInfo.length; i++) {
                if (networkInfo[i].getState() == NetworkInfo.State.CONNECTED) {
                    return true;
                }
            }
        }
    }
    return false;
}
```

5554:Nexus_5X_API_23



访问城市不存在

5554:Nexus_5X_API_23



访问成功（默认为上海）

关键代码：

【使用多线程提交请求】在前文代码提及的“postRequest()”中实现，主要包含 HttpURLConnection 的连接、数据流写入与提交、返回值的读取，XmlPullParser 解析 xml 文件、Handler 更新 UI。

HttpURLConnection 的连接：

```
URL url = new URL(web);
connection = (HttpURLConnection) url.openConnection();
connection.setRequestMethod("POST");
connection.setConnectTimeout(8000);
connection.setReadTimeout(8000);
connection.setDoOutput(true);
```

数据流的写入与提交：

```
DataOutputStream out = new DataOutputStream(connection.getOutputStream());
String request = City.getText().toString();
request = URLEncoder.encode(request, "utf-8");
out.writeBytes("theCityCode="+ request + "&theUserID="+ "489bf01aa5624255ac630fb50b076520");
```

返回值的读取与转换成 String 形式、调用 Handler 更新 UI：

```
InputStream in = connection.getInputStream();
BufferedReader reader = new BufferedReader(new InputStreamReader(in));
StringBuilder response = new StringBuilder();
String line;
while ((line = reader.readLine()) != null) {
    response.append(line);
}
```

XmlPullParser 解析 xml 文件、Handler 更新 UI。

```
Message message = new Message();
message.what = UPDATE_CONTENT;
message.obj = parseXMLWithPull(response.toString());
handler.sendMessage(message);
```

【XmlPullParser 解析 xml 文件】即上文代码提及的“parseXMLWithPull()”方法

```
public ArrayList<String> parseXMLWithPull(String str_xml) {
    list = new ArrayList<String>();
    try {
        XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
        XmlPullParser parser = factory.newPullParser();
        parser.setInput(new StringReader(str_xml));
        int eventType = parser.getEventType();
        while(eventType != XmlPullParser.END_DOCUMENT) {
            switch (eventType) {
                case XmlPullParser.START_TAG:
                    if ("string".equals(parser.getName())) {
```

```

        String str = parser.nextText();
        list.add(str);
    }
    break;
case XmlPullParser.END_TAG:
    break;
default:
    break;
}
eventType = parser.next();
}
} catch (Exception e) {
    e.printStackTrace();
}
return list;
}

```

【使用 Handler 更新 UI】

UI 的更新可分为两种情况：访问异常的“查询为空”、“免费用户高速访问限制”、“免费用户每日访问次数限制”等，此类情况通过一律使用“Toast”处理，此处不显示代码；另一种情况为访问正常，获取到想到的信息，从 XmlPullParser 解析得到的字符串数组链表中提取有用信息。主要是利用正则表达式，并结合其他方法进一步解析字符串，下面以各项指数的信息（界面中 ListView 中信息）获取为例。（代码并不完整。）

```

String exponent = response.get(6);
if (!"暂无预报".equals(exponent)) {
    Pattern p1 = Pattern.compile("(; +?)(.*?)(。 +?)");
    Matcher m1 = p1.matcher(exponent);
    ArrayList<String> list_content = new ArrayList<>();
    while (m1.find()) {
        list_content.add(m1.group());
    }
    listItems = new ArrayList<HashMap<String, String>>();
    for (int i = 0; i < list_exponents.length; i++) {
        HashMap<String, String> map = new HashMap<String, String>();
        map.put("exponent", list_exponents[i]);
        map.put("content", list_content.get(i).substring(1, list_content.get(i).length() - 1));
        listItems.add(map);
    }
}
SimpleAdapter simpleAdapter = new SimpleAdapter(MainActivity.this, listItems,
R.layout.list_item,
    new String[]{"exponent", "content"}, new int[] {R.id.exponent, R.id.content});
listView.setAdapter(simpleAdapter);

```

查询不到完整的数据时，尽可能多地解析数据并显示出来，如下图华盛顿的天气查询：后五天的天气状况数据可查询得到，依旧显示，各项指数和当天天气无数据，显示为“暂无”。



RecyclerView 的实现（可横向拉动界面底部的五天天气状况）

【关键代码】

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:visibility="gone"
    android:scrollbars="none"/>
```

设置 RecyclerView 横向滚动:

```
mRecyclerView = (RecyclerView)findViewById(R.id.recyclerView);
LinearLayoutManager layoutManager = new LinearLayoutManager(this);
layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
mRecyclerView.setLayoutManager(layoutManager);
```

设置 RecyclerView 适配器:

```
mRecyclerView.setAdapter(new WeatherAdapter(MainActivity.this, weather_list));
mRecyclerView.setVisibility(View.VISIBLE);
```

【遇到的问题与解决方法】

1. 网络检测中因为忽视了手机网络有多种:“移动数据”和“Wifi 无线网络”,使得测试结果一直有问题,后来经过网上搜索,成功改正。
2. 实验中解析得到字符串后,一直纠结如何得到字符串中想要的信息,最初想实现指定位置截取子字符串的接口,但具体是哪个位置不好确认,该方法不实际,最后使用正则表达式匹配相关字符串,主要是“:”、“;”、“.”等符号,进行字符串的分隔,再除去字符串头尾的符号即可得到需要的信息。用到了 `Pattern.compile()`、`matcher()`、`split()` 等方法。
3. 有些查询结果与预期结果不一致,相关信息获取不到,却使得程序崩溃,多次查询获取多各路情况的结果后,将信息获取不到的或与预期不一致的,设置为“暂无”并显示,通过这样,显示的查询结果板块是一样的,只是信息不同而已,比较好操作。
4. RecyclerView 在使用时,一直闪退,界面不能显示,发现是 RecyclerView.java 中的 `onMeasure()` 不匹配,于是上网求解原因,发现是 LinearLayoutManager 没有设置,在 handler 更新 UI 的代码中设置 `LinearLayoutManager layoutManager = new LinearLayoutManager(this);`
`layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL)`后,还是闪退,经过多次调试,发现当两行代码放在 `onCreate()` 中时,界面显示成功,RecyclerView 得到应用。
5. 本次实验因为网站维修原因,免费用户不可使用,实验最初为求方便,用了实验文档中一张截图的用户 ID (应该是 TA 注册的),一开始一切如常,成功返回数据,但当要断开模拟器网络,截图写实验报告时,再次连接,却无法成功访问,以为是模拟器网络没有接好,一直推敲着,之后又看代码又修改,花了好多时间,最后发现是用户 ID 验证失败,人品太差,可能刚好过期了。注册后,使用新的用户 ID 即可。

【实验心得与体会】

1. 本次实验目的是访问 Webservice 并解析 xml 文档数据,访问 Webservice 的方法有多种,实验用的是使用 `HttpURLConnection` 访问 Webservice,使用 `XmlPullParser` 解析 xml 文档数据。同时还用到了之前学过的多线程以及 Handler 更新 UI。通过实验,我对 `HttpURLConnection` 的使用有了一定的了解,

使用过程主要有连接、数据提交、获取返回数据、断开连接。连接时用 `openConnection()`，数据提交和获取返回数据是以流的形式，分别为 `getOutputStream()`、`getInputStream()`，断开即为 `disconnect()`。

另外也熟悉了 `XmlPullParser`，其解析实质是对每个字符串的 `Tag` 进行匹配，然后 `switch` 选择处理。

2. 拓展实验为熟悉 `RecyclerView` 控件的使用，这是一个强大的滑动组件，是 `ListView` 的升级版，灵活性高，可以横向、纵向滑动，实验过程中还学会了 `Android Studio` 导入 `jar` 包的方法，也掌握了 `RecyclerView` 控件的简单使用。
3. 本次实验过程中在布局方面还用到了多种布局，有 `LinearLayout`、`RelativeLayout`、`GridLayout`，在引入 `RecyclerView` 之前，还使用了 `TableLayout` 来显示五天的天气状况，通过对这些布局的综合使用和控件的属性设置，我对安卓基础界面编程更加得心应手了。