

# 实验八：网络访问

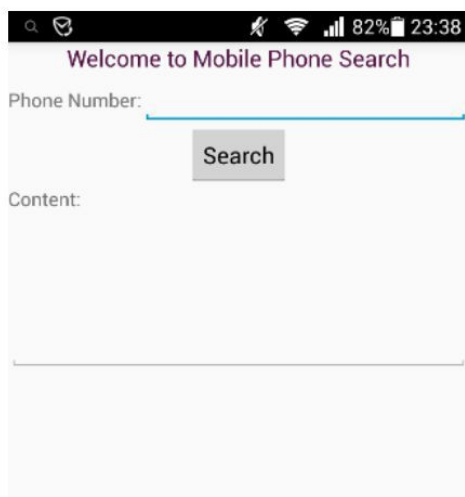
13331388 庄梓嘉

## 实验目的

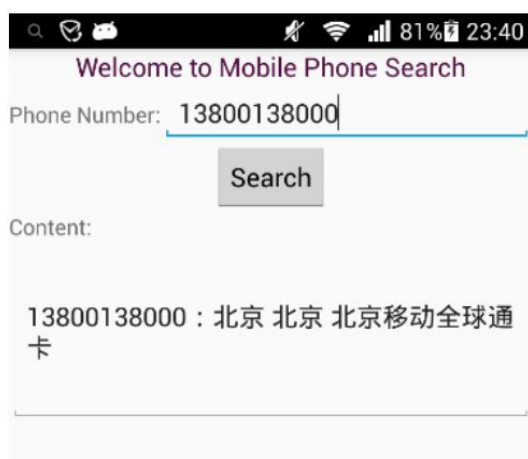
- 1.熟练使用 HttpURLConnection 访问 Webservice
- 2.熟练使用多线程以及 Handler 更新 UI
- 3.熟悉使用 XmlPullParser 解析 xml 文档数据

## 实验内容

(1) 程序界面如下图所示：



(2) 当输入手机号码后，点击 Search 按钮后，能够查询手机归属地：



## 【拓展项】

使用 Sax 和 Dom 方法其中一种方法，重新改写本实验中解析 xml 部分。

## 实验步骤

(1) 定义需要用到的 Web Service 地址。

```
private static final String url = "http://webservice.webxml.com.cn/WebServices/MobileCodeWS.asmx/getMobileCodeInfo";
```

(2) 建立一个 Http 连接并设置其访问方式和时间。

```
private void sendRequestWithURLConnection() {  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            HttpURLConnection connection = null;  
            try {  
                //create a connection use url  
                connection = (HttpURLConnection) ((new URL(url.toString())).openConnection());  
                //set method  
                connection.setRequestMethod("POST");  
                connection.setConnectTimeout(80000);  
                connection.setReadTimeout(80000);  
            }  
        }  
    }).start();  
}
```

(3) 将参数提交到网络连接，并且请求网络数据。

```
DataOutputStream out = new DataOutputStream(connection.getOutputStream());  
//use post method to our data  
out.writeBytes("mobileCode=" + phoneNumber.getText().toString() + "&userID=");  
//Log.v("test", phoneNumber.getText().toString());
```

(4) 提取数据，转化为字符串。

```
//get response data  
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);  
}  
}
```

( 5 ) 通过 handleMessage 函数，负责对子线程传递的消息进行分发和处理。

```
private Handler handler = new Handler() {  
    public void handleMessage(Message message) {  
        switch (message.what) {  
            case UPDATE_CONTENT:  
                content.setText(message.obj.toString());  
                break;  
            default:  
                break;  
        }  
    }  
};
```

( 6 ) 通过 XmlPullParser 解析 xml 内容。

```
private String parseXMLWithPull(String xml) {  
    String str = "";  
    try {  
        //use pull to parse xml  
        XmlPullParserFactory factory = XmlPullParserFactory.newInstance();  
        XmlPullParser parser = factory.newPullParser();  
        parser.setInput(new StringReader(xml));  
        int eventType = parser.getEventType();  
        while (eventType != XmlPullParser.END_DOCUMENT) {  
            switch (eventType) {  
                case XmlPullParser.START_TAG:  
                    if ("string".equals(parser.getName())) {  
                        str = parser.nextText();  
                    }  
                    break;  
                case XmlPullParser.END_TAG:  
                    break;  
                default:  
                    break;  
            }  
            eventType = parser.next();  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return str;  
}
```

( 7 ) 将 message 传递回并补充完整 sendRequestWithURLConnection , 启动线程。

```
        Message message = new Message();
        message.what = UPDATE_CONTENT;
        message.obj = parseXMLWithPull(response.toString());
        handler.sendMessage(message);
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (connection != null) {
            connection.disconnect();
        }
    }
}
}).start();
}
```

( 8 ) 在 Manifest 里添加权限。

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

### 【拓展项】

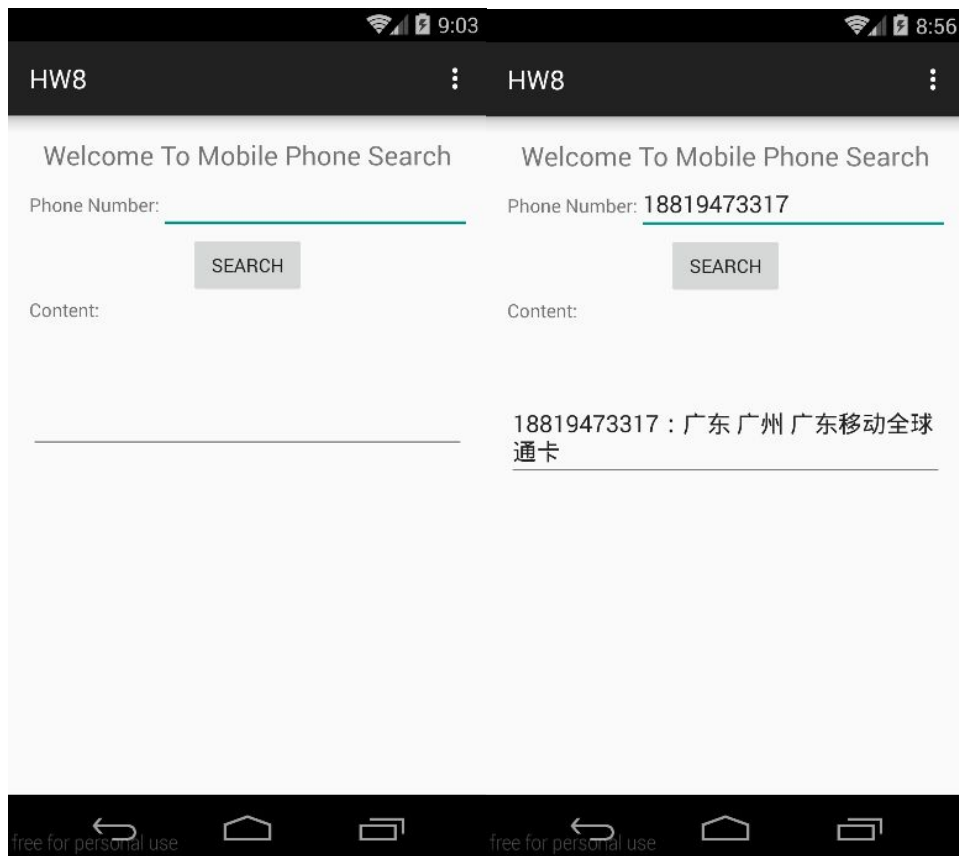
通过 DocumentBuilder 解析 xml。

```
//use dom to parse xml
private String parseXMLWithDom(String xml) {
    String str = "";
    try {
        //use pull to parse xml
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(xml);
        Element element = document.getDocumentElement();
        str = element.getFirstChild().getNodeValue();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return str;
}
```

## 实验结果

初始页面

搜索成功



## 参考资料

手机归属地 Webservice 地址：

<http://webservice.webxml.com.cn/WebServices/MobileCodeWS.aspx?p=getMobileCodeInfo>

免费使用 Webservice：

[http://www.webxml.com.cn/zh\\_cn/web\\_services.aspx?offset=1](http://www.webxml.com.cn/zh_cn/web_services.aspx?offset=1)

Android 系列--DOM、SAX、Pull 解析 XML：

<http://www.android100.org/html/201510/08/188783.html>

## 实验总结

这次实验要求使用网络访问，并且通过 PULL 解析 XML 文件。在使用到 Web Service 时要注意到输出流的格式问题，变量名和变量值检查清楚。在测试哪里出错的时候，可以在不同的地方添加 Log 输出来判断。有个比较坑的地方是，有时候不是代码问题，是该网站的服务暂时停止了服务，需要过段时间才能使用，浪费了好多时间纠结好久哪里代码问题。与上学期 Win8 应用开发时候相比较，也有类似的作业，所以感觉上比较容易上手。