

# Android Web 开发

## 使用说明

【源代码根目录】： 工程目录 INTERNET

【记号】:

(@编程练习): 表明该实验是需要编将工程文件和实验报告一起提交。

(@团队编程练习): 表明该实验是以小组为单位完成的, 每个小组完成一份程序和报告即可, 报告和工程和其他的实验报告和工程最后要分开交。

本次实验的个人编程部分, 请建立在名为 KH7 的 project 中。上传方式:

<http://disk.lehu.shu.edu.cn/index.aspx>

上传码:

**android-sybk (本科)**

**android-syyjs (研究生)**

【实验报告的要求】

- 1、 文件名规范: 学号+姓名+实验名称.doc
- 2、 内容格式见 实验报告格式.doc

【工程提交的要求】

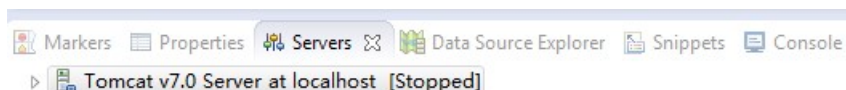
## 目录

Android	Web 开发 .....	1
	【课堂实验 KT8.1】: network 连接管理 .....	3
	【课堂实验 KT8.2】: 使用 http 进行基本的通信 .....	4
	【课堂实验 KT8.3】: WebView 应用 .....	7
	【课后实验 KH8.1】: 用 JSON 和 XML 保存数据 .....	10

### 【本讲额外说明】

Tools 目录中的三个文件，用于为测试本讲例子而搭建的 web 端服务器 jsp 环境。

- 1、eclipse-jee-kepler-SR1-win32-x86\_64 解压，获得 jee 开发 ide。
- 2、将 tomcat 插件解压后拷贝到步骤一中 plugin 目录。讲 tomcat 集成到 ide 中
- 3、本讲中 kt7.2 用来自动生成 xml 文件的库：jakarta-taglibs-standard-1.1.2（Jee-workspace 中已经导入了）。运行步骤一解压的 eclipse，将 workspace 指向发布中的 jee-workspace 即可。
  - a) 运行 jee-eclipse，选择 windows-preference。选择 tomcat，选择 tomcat7.0.x,设置好 tomcat base 目录（可以建一个），选择 tomcat-advance 设置好 tomcat base 目录。会自动下载 tomcat7.0.x 到该目录



- 4、代码中的 ip 地址请自行替换成自机器的 ip：比如 192.168.0.104。注意不要用 localhost 或者 127.0.0.1.即使在本机调试，上述地址也无法让虚拟机连接到本机上的 tomcat。

## 【课堂实验 KT8.1】：network 连接管理

【工程路径】:NetworkManagerSample

【工程描述】：该工程展示了如何检测网络连接的类型和状态。

【知识点注释】：

主要用到两个类

- [ConnectivityManager](#): Answers queries about the state of network connectivity. It also notifies applications when network connectivity changes.
- [NetworkInfo](#): Describes the status of a network interface of a given type (currently either Mobile or Wi-Fi).

需要添加权限

- [android.permission.INTERNET](#)—Allows applications to open network sockets.
- [android.permission.ACCESS\\_NETWORK\\_STATE](#)—Allows applications to access information about networks.

```

public boolean isOnline() {
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (null!=networkInfo) {
        boolean wifiConnected = networkInfo.getType() == ConnectivityManager.TYPE_WIFI;
        boolean mobileConnected = networkInfo.getType() == ConnectivityManager.TYPE_MOBILE;

        Log.d("isOnline", "Wifi connected: " + wifiConnected);
        Log.d("isOnline", "Mobile connected: " + mobileConnected);
    }

    return (networkInfo != null && networkInfo.isConnected());
}

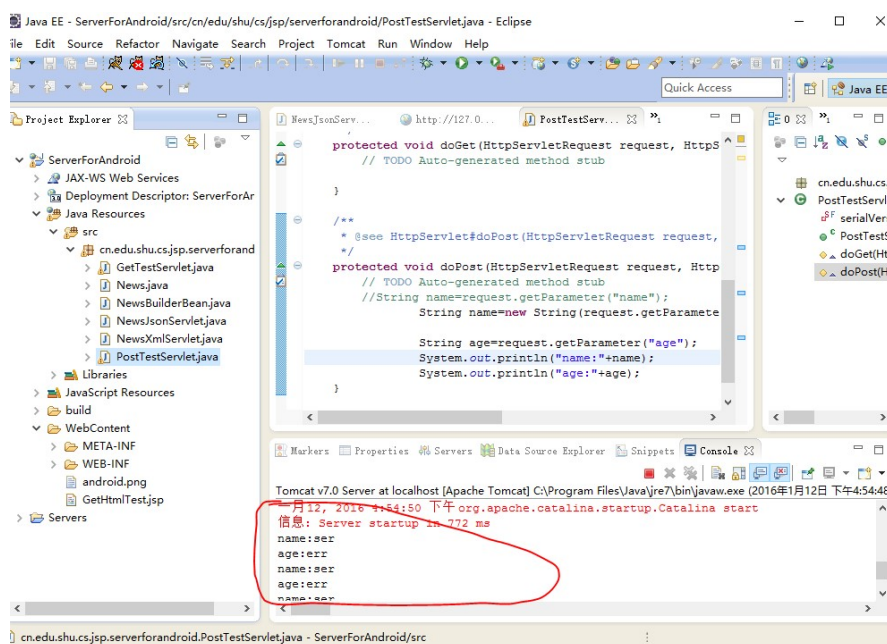
```

## 【课堂实验 KT8.2】：使用 http 进行基本的通信

【工程路径】： WebSample

【工程描述】：该工程展示如何通过http协议，同web服务器进行基本的数据通信。展示了下载图片，下载html文本，下载并解析xml，下载并解析json，通过get方法上传，通过post方法上传等。

【实验步骤】：运行程序，点击按钮观看结果。最后两项get,post方法测试，请输入任意用户名和密码，点击按钮后，请切换到web服务器控制台，查看输出



【知识点注释】

类: InternetTools 包含了和web服务器通信的代码。提供了一系列静态函数供本例中各activity调用。如:

---

```
static final String IPADDRESS="http://111.186.110.163"; //开发机 ip, 请根据实际情况修改
static final String AUTHORITY=IPADDRESS+":8080/ServerForAndroid/";
public static Bitmap getImage(String path) throws Exception{
    path=AUTHORITY+"android.png" ;
    Log.i("address", path);
    URL url=new URL(path);
    HttpURLConnection conn=(HttpURLConnection)url.openConnection();
    conn.setConnectTimeout(5000);
    conn.setRequestMethod("GET");
    if(conn.getResponseCode()==200){
        Log.i("200", "good");
        InputStream inStream=conn.getInputStream();
        Bitmap bitmap=BitmapFactory.decodeStream(inStream);
        //save as a file
        /*
        byte[] data=WebSampleTools.read(inStream);
        File file=new File("icon.jpg");
        FileOutputStream outputstream=new FileOutputStream(file);
        outputstream.write(data);
        outputstream.close();
        */

        return bitmap;
    }

    return null;
}
```

---

以上代码使用HttpURLConnection类建立http连接。根据返回的响应号是否等于200判定请求是否成功。

注意: 注释中的代码可以把请求获得的图片保存为本地图片。

其余几个函数也大致如此:

---

```
public static List<News> getNews(String path) throws Exception{
    //a debug connection string
    //path=AUTHORITY+"NewsXmlServlet" ;
    path=AUTHORITY+"NewsJsonServlet" ;
    URL url=new URL(path);
    HttpURLConnection conn=(HttpURLConnection)url.openConnection();
    conn.setConnectTimeout(5000);
    conn.setRequestMethod("GET");
```

```

        if(conn.getResponseCode()==200){
            InputStream inStream=conn.getInputStream();
            //return parseXML(inStream);
            return parseJSON(inStream);
        }
        return null;
    }
}

```

该函数用在 GetNewsActivity，目的在于展示如何从服务器获取一个 xml 数据或者 json 格式的数据。

```

        //return parseXML(inStream);
        return parseJSON(inStream);
    }
}

```

分别解析这两种格式的数据，本例中的 web 端采用了 jsp 开发。Web 端代码 NewsJsonServlet 和 NewsXmlServlet 分别产生这两种格式的数据。

GetMethodTestActivity 和 PostMethodTestActivity 分别演示了如何通过 get 方式和 post 方式向 web 服务端发送数据。比如常见的用 get 方式发送用户登录信息。用 post 方式发送用户注册信息。

```

public static boolean getMethosTest(String path,String name,String password) throws
Exception{

```

```

    //a debug connection string
    path=AUTHORITY+"GetTestServlet" ;
    Map<String,String> params=new HashMap<String,String>();
    params.put("name",name);
    params.put("password", password);

    StringBuilder sb=new StringBuilder(path);
    sb.append("?");
    for(Map.Entry<String,String> entry:params.entrySet()){
        sb.append(entry.getKey()).append("=");
        sb.append(URLEncoder.encode(entry.getValue(),"UTF-8"));
        sb.append("&");
    }
    sb.deleteCharAt(sb.length()-1);

    URL url=new URL(sb.toString());
    HttpURLConnection conn=(HttpURLConnection)url.openConnection();
    conn.setConnectTimeout(5000);
    conn.setRequestMethod("GET");
    if(conn.getResponseCode()==200){
        return true;
    }
}

```

```

        return false;
    }

    public static boolean postMethodTest(String path,String name,String age) throws
    Exception{
        //a debug connection string
        path=AUTHORITY+"PostTestServlet" ;
        Map<String,String> params=new HashMap<String,String>();
        params.put("name",name);
        params.put("age", age);

        StringBuilder sb=new StringBuilder();
        for(Map.Entry<String,String> entry:params.entrySet()){
            sb.append(entry.getKey()).append("=");
            sb.append(URLEncoder.encode(entry.getValue(),"UTF-8"));
            sb.append("&");
        }
        sb.deleteCharAt(sb.length()-1);

        byte[] data=sb.toString().getBytes();

        URL url=new URL(path);
        HttpURLConnection conn=(HttpURLConnection)url.openConnection();
        conn.setConnectTimeout(5000);
        conn.setRequestMethod("POST");
        conn.setDoOutput(true);

        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn.setRequestProperty("Content-Length",data.length+"");
        OutputStream outputStream=conn.getOutputStream();
        outputStream.write(data);
        outputStream.flush();
        if(conn.getResponseCode()==200){
            return true;
        }
        return false;
    }
}

```

## 【课堂实验 KT8.3】：WebView 应用

【工程路径】: {……/07}/ WebViewSample

【工程描述】：该工程展示使用webview显示html页面，通过页面中的js脚本来调用本地api。本例实现了一个用html语言编写的通讯录UI。

ID	NAME	PHONE
1	tom	<a href="#">66133</a>
2	Mary	<a href="#">66313</a>
3	Jone	<a href="#">66432</a>

【知识点注释】：

```
wv=(WebView)this.findViewById(R.id.webview);  
wv.getSettings().setJavaScriptEnabled(true); //让webview  
支持javascript  
wv.getSettings().setSaveFormData(false);  
wv.addJavascriptInterface(new SharpJavaScript(),  
"sharp");//生成SharpJavaScript类的实例，在javascript中用“sharp”  
来引用。  
wv.getSettings().setDefaultTextEncodingName("UTF-8") ;  
wv.loadUrl("file:///android_asset/phone.html");//html编  
写的UI，phone.html放在assets目录中，请注意访问它的方法。
```

```
public class SharpJavaScript {  
    public void contactlist(){  
        try{  
            String json=buildJson(getContacts());
```



```

        wv.loadUrl("javascript:show('"+json+"'");
    }catch(Exception e){
        e.printStackTrace();
    }
}

public void call(String phone){
    Intent intent=new Intent(Intent.ACTION_CALL,Uri.parse("tel:"+phone));
    MainActivity.this.startActivity(intent);
}

public List<Contact> getContacts(){
    List<Contact> contacts=new ArrayList<Contact>();
    contacts.add(new Contact(1,"tom","66133"));
    contacts.add(new Contact(2,"Mary","66313"));
    contacts.add(new Contact(3,"Jone","66432"));
    return contacts;
}

public String buildJson(List<Contact> contacts) throws Exception{
    JSONArray array=new JSONArray();
    for(Contact contact:contacts){
        JSONObject jb=new JSONObject();
        jb.put("id",contact.getId());
        jb.put("name",contact.getName());
        jb.put("phone",contact.getPhone());
        array.put(jb);
    }
    return array.toString();
}
}

```

本例中类 SharpJavaScript 用来从 android 端本地获取通讯录信息，转换成 json 的方式供 html 中的 javascript 读取。请注意，本例用 getContacts() 方法模拟返回本地通讯录信息，实际中可以自行编写代码获取真正的通讯录数据。

用来作为 UI 的本地 phone.html 文件，保存在 assets 目录中。

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>html phone</title>
<script type="text/javascript">

```

```

function show(jsondata){
    var jsonobjs=eval(jsondata);
    var table=document.getElementById("personTable");
    for(var i=0;i<jsonobjs.length;i++){
        var tr=table.insertRow(table.rows.length);
        var td1=tr.insertCell(0);
        var td2=tr.insertCell(1);
        td2.align="center";
        var td3=tr.insertCell(2);
        td1.innerHTML=jsonobjs[i].id;
        td2.innerHTML=jsonobjs[i].name;
        td3.innerHTML="<a
href='javascript:sharp.call(\""+jsonobjs[i].phone+"\"')>"+jsonobjs[i].phone+"</a>";
    }
}

</script>
</head>
<body onload="javascript:sharp.contactlist()">
<table border="0" width="100%" id="personTable" cellspacing="0">
<tr>
<td width="15%">ID</td>
<td align="center">NAME</td>
<td width="15%">PHONE</td>
</tr>
</table>
</body>
</html>

```

请注意本例中 html\javascript\java 三类代码调用关系:

- 1、WebView 加载 phone.html  
`wv.loadUrl("file:///android_asset/phone.html")`
- 2、phone.html 装载中触发 javascript 事件，从而调用 java 中的 contactlist() 方法  
`<body onload="javascript:sharp.contactlist()">`
- 3、contactlist() 方法获取到通讯录 json 数据，在通知 webview 执行 phone.html 中的 javascript 函数 show()  
`String json=buildJson(getContacts());`  
`wv.loadUrl("javascript:show('"+json+"')");`
- 4、show () 函数获取到了 json 字符串，抽取后显示在 webview 中。

## 【课后实验 KH8.1】:用 JSON 和 XML 保存数据

【工程描述】:（@编程练习）请就数据存储章节中的比赛管理 app 的要求，运用 json 或者

xml 文件替代 sqlite 作为数据存储方式。