



# 《移动终端开发技术》 电子教案

## 第二单元 使用 HTTP 协议访问网络

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	使用 HTTP 协议访问网络	单元教学学时	3 课时
		在整体设计中的位置	第 27 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练运用 HttpURLConnection 及 OkHttp 访问网络。	1、掌握使用使用 HttpURLConnection 访问网络的方法； 2、掌握使用 OkHttp 访问网络的方法。	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	教学重点：使用 HttpURLConnection 访问网络 教学难点：使用 OkHttp 访问网络		
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。		
课前需掌握的知识点	HTTP（Hyper Text Transfer Protocol）协议即超文本传输协议。 过去，Android 上发送 http 请求一般有两种方式，HttpURLConnection 和 HttpClient。在 Android6.0 系统中 HttpClient 的功能被完全移除了。		
教学任务分解	任务一、使用 HttpURLConnection 访问网络 任务二、使用 OkHttp 访问网络		
教学总结			

## 一、情景导入

日常生活中，大多数人遇到了问题都会使用手机进行百度搜索。在浏览器的地址栏中输入百度的网址点击搜索，此时就会进入百度主页。这个访问百度的过程就是通过 HTTP 协议完成的，所谓的 HTTP（Hyper Text Transfer Protocol）协议即超文本传输协议，它规定了浏览器和万维网服务器之间互相通信的规则。

当客户端在与服务器端建立连接后，向服务器端发送的请求，被称作 HTTP 请求。服务器端接收到请求后会做出响应，称为 HTTP 响应。

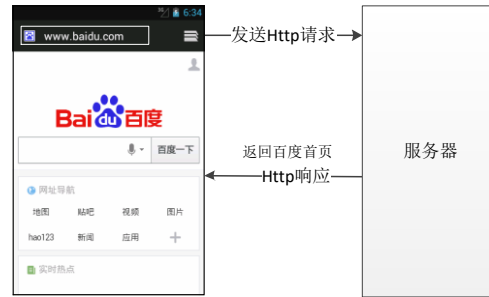


图 1 访问网络原理

## 二、复习

使用 WebView 加载网页。

## 三、知识讲解

### 任务一、使用 HttpURLConnection 访问网络

过去，Android 上发送 http 请求一般有两种方式，HttpURLConnection 和 HttpClient。在 Android6.0 系统中 HttpClient 的功能被完全移除了。

使用 HttpURLConnection 的步骤：

第一步、获取 HttpURLConnection 的实例：

```
URL url = new URL("http://www.baidu.com");
```

```
HttpURLConnection conn =(HttpURLConnection)url.openConnection();
```

第二步、设置 HTTP 请求所使用的方法：GET 和 POST

GET 表示从服务器获取数据，POST 表示提交数据给服务器。

```
conn.setRequestMethod("GET");
```

第三步、进行自由定制：

```
conn.setConnectTimeout(5000); //连接超时
```

```
conn.setReadTimeout(5000); //读取超时
```

第四步、调用 `getInputStream()`方法获取到服务器返回的输入流

```
InputStream is = conn.getInputStream();
```

对输入流进行读取。

第五步、调用 `disconnect()`方法将这个 HTTP 连接关闭

```
conn.disconnect();
```

### 案例、使用 HttpURLConnection 访问网络



图 2 访问百度网页运行效果

新建项目 HttpURLConnection 代码如下：

#### Activity\_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/send_request"
        android:text="send request"/>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/response_text"/>
    </ScrollView>
</LinearLayout>
```

#### MainActivity.java

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    TextView responseText;
```



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button sendRequest=(Button) findViewById(R.id.send_request);
    responseText=(TextView) findViewById(R.id.response_text);
    sendRequest.setOnClickListener(this);
    /* sendRequest.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            sendRequestWithHttpURLConnection();
        }
    });*/
}

@Override
public void onClick(View view) {
    if(view.getId()==R.id.send_request){
        sendRequestWithHttpURLConnection();
    }
}

private void sendRequestWithHttpURLConnection(){
    //开启线程来发起网络请求
    new Thread(new Runnable() {
        @Override
        public void run() {
            HttpURLConnection connection=null;
            BufferedReader reader=null;
            try{
                URL url=new URL("https://www.baidu.com");
                connection=(HttpURLConnection)url.openConnection();
                connection.setRequestMethod("GET");
                connection.setConnectTimeout(8000);
                connection.setReadTimeout(8000);
                InputStream in=connection.getInputStream();
                //下面对获取到的输入流进行读取
                reader=new BufferedReader(new InputStreamReader(in));
                StringBuilder response=new StringBuilder();
                String line;
                while((line=reader.readLine())!=null){
                    response.append(line);
                }
                showResponse(response.toString());
            }catch (Exception e){
```



```
        e.printStackTrace();
    }finally {
        if(reader!=null){
            try{
                reader.close();
            }catch (IOException e){
                e.printStackTrace();
            }
        }
        if(connection!=null){
            connection.disconnect();
        }
    }
}
}).start();
}

private void showResponse(final String response){
    //将线程切换到主线程, 才能进行 UI 操作
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            //在这里进行 UI 操作, 将结果显示到界面上
            responseText.setText(response);
        }
    });
}
}
```

#### AndroidManifest.xml 声明访问网络权限

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.cyy.httpurlconnection">
    <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

#### 任务二、使用 OkHttp 访问网络

OkHttp 是 Square 公司开发的开源项目, 已经成为广大 Android 开发者首选的网络通信库。

使用 OkHttp 的步骤:

第一步、在项目中添加 OkHttp 库的依赖。

编辑 app/build.gradle 文件

```
dependencies {
    ...
    compile 'com.android.support:appcompat-v7:25.0.1'
```



```
testCompile 'junit:junit:4.12'  
compile 'com.squareup.okhttp3:okhttp:3.4.1'  
}
```

第二步、创建一个 OkHttpClient 实例。

```
OkHttpClient client=new OkHttpClient();
```

第三步、创建 Request 对象。

```
Request request=new Request.Builder().url("http://www.baidu.com").build();
```

通过 url()方法来设置目标的网络地址, Request 是 OkHttpClient 中访问的请求, Builder 是辅助类。

第四步、创建 Call 对象。

```
Response response=client.newCall(request).execute();
```

通过 OkHttpClient 的 newCall()方法创建 Call 对象, execute()方法来发送请求并获取服务器返回的数据

第五步、得到返回的具体内容。

```
String responseData=response.body().string();
```

response.body()返回 ResponseBody 类可以方便的获取 string

案例、使用 OkHttpClient 访问网络。

修改项目 HttpURLConnection, 代码如下:

```
@Override  
public void onClick(View view) {  
    if(view.getId()==R.id.send_request){  
        sendRequestWithOkHttp();  
        // sendRequestWithHttpURLConnection();  
    }  
}  
  
private void sendRequestWithOkHttp(){  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            try{  
                //创建 OkHttpClient 实例  
                OkHttpClient client=new OkHttpClient();  
                //创建 Request 对象, url()方法来设置目标的网络地址  
                Request request=new  
Request.Builder().url("http://www.baidu.com").build();  
                //通过 OkHttpClient 的 newCall()方法创建 Call 对象, execute()方法来发  
送请求并获取服务器返回的数据  
                Response response=client.newCall(request).execute();  
                //得到返回的具体内容, response.body()返回 ResponseBody 类可以方便的获  
取 string  
                String responseData=response.body().string();
```



```
        showResponse(responseData);  
    }catch (Exception e){  
        e.printStackTrace();  
    }  
    }  
}).start();  
}
```

#### 四、演示作品

各小组演示汇报作品，教师与其他小组总结评价作品，各小组课后完善作品并提交到教学平台。

#### 五、知识巩固

总结知识点，使用教学平台中的随堂练习题巩固本所学知识。

拓展  
作业

运用 `HttpURLConnection` 及 `OkHttp` 访问网络，编写代码并提交运行结果。

教学  
后记