1. 学习Tomcat

参考：<https://www.bilibili.com/video/BV1dJ411N7Um?p=4>

1. 安装Tomcat

参考：<https://zhuanlan.zhihu.com/p/37852137>

安装成功截图



1. 实现第一个任务

使用 socket API 实现简单的 HTTP 客户

（1） 使用 sendMessage 方法发送符合 http 协议定义的消息给 web 服务器以获得想要的页面

（2） 使用 receiveMessage 方法接收返回的页面源文件并显示。

代码

|  |
| --- |
| import java.io.BufferedInputStream;  import java.io.BufferedReader;  import java.io.BufferedWriter;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.OutputStreamWriter;  import java.net.InetSocketAddress;  import java.net.Socket;  import java.net.SocketAddress;  public class HttpSocketClient {  private Socket mSocket;  public static void main(String[] args) {  HttpSocketClient client = new HttpSocketClient();  try {  client.sendGet("www.baidu.com",80,"/");  } catch (IOException e) {  e.printStackTrace();  }  }  public HttpSocketClient() {  this.mSocket = new Socket();  }  /\*\* 在百度服务器面前，这就是一个正常的浏览器  \* @param host  \* @param port  \* @param path  \* @throws IOException  \*/  void sendGet(String host, int port, String path) throws IOException {  SocketAddress dest = new InetSocketAddress(host, port);  mSocket.connect(dest);  OutputStreamWriter streamWriter = new OutputStreamWriter(mSocket.getOutputStream());  BufferedWriter bufferedWriter = new BufferedWriter(streamWriter);  bufferedWriter.write("GET " + path + " HTTP/1.1\r\n");  bufferedWriter.write("Host: " + host + "\r\n");  bufferedWriter.write("Connection: " + "keep-alive" + "\r\n");  bufferedWriter.write("User-Agent: " + "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36" + "\r\n");  bufferedWriter.write("Accept: " + "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8" + "\r\n");  bufferedWriter.write("Accept-Language: " + "zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7" + "\r\n");  bufferedWriter.write("\r\n");  bufferedWriter.flush(); //flush一下很重要，等于说已经写完了  BufferedInputStream stream = new BufferedInputStream(mSocket.getInputStream());  BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(stream));  String line = null;  while ((line = bufferedReader.readLine())!=null) {  System.out.println(line);  }  bufferedReader.close();  bufferedWriter.close();  mSocket.close();  }  } |

参考：<https://www.cnblogs.com/ranyonsue/p/5984001.html>

1. 实现第二个任务

使用隐式表单域传输会话状态数据

（1） 编写客户端请求页面 form.html

（2） 编写 form.html 所触发的脚本 hiddenform.c

（3） 编写 hiddform.cgi 动态生成页面中触发的脚本 hidden.form2.c

学习html：<https://www.runoob.com/html/html-intro.html>

学习html中的表单：<https://www.runoob.com/html/html-forms.html>

什么是隐式表单：<https://www.debugease.com/javaweb/1013492.html>

隐式表单的作用：<https://www.jianshu.com/p/5bdf753acef2>

学习xml：<https://www.runoob.com/xml/xml-tutorial.html>

学习如何使用Tomcat开发：

<https://www.bilibili.com/video/BV1gJ411q7Lf?from=search&seid=600581961118443894&spm_id_from=333.337.0.0>

1. 实现第三个任务

3. 使用 cookie 传递状态数据

（1） 编写请求页面 cookie.html

（2） 编写 cookie.html 所触发的脚本 cookie.c

（3） 编写由 cookie.c 动态生成页面中触发的脚本 cookie2.c

学习cookie：

Cookie、Session、Token的区别: <https://www.bilibili.com/video/BV1ob4y1Y7Ep?from=search&seid=8681360534596789764&spm_id_from=333.337.0.0>

Cookie使用步骤与Cookie原理: https://blog.csdn.net/weixin\_43691058/article/details/104052875?utm\_medium=distribute.pc\_aggpage\_search\_result.none-task-blog-2~aggregatepage~first\_rank\_ecpm\_v1~rank\_v31\_ecpm-1-104052875.pc\_agg\_new\_rank&utm\_term=%E4%BD%BF%E7%94%A8cookie%E4%BC%A0%E9%80%92%E7%8A%B6%E6%80%81%E6%95%B0%E6%8D%AE&spm=1000.2123.3001.4430