

基础知识：用Java实现Web服务器 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E5_9F_BA_E7_A1_80_E7_9F_A5_E8_c104_145059.htm 超文本传输协议

(HTTP) 是位于TCP/IP 协议的应用层，是最广为人知的协议，也是互连网中最核心的协议之一，同样，HTTP 也是基于 C/S 或 B/S 模型实现的。事实上，我们使用的浏览器如Netscape 或IE 是实现HTTP 协议中的客户端，而一些常用的Web 服务器软件如Apache、 IIS 和iPlanet Web Server 等是实现HTTP 协议中的服务器端。Web 页由服务端资源定位，传输到浏览器，经过浏览器的解释后，被客户所看到。Web 的工作基于客户机/服务器计算模型，由Web 浏览器(客户机)和Web服务器(服务器)构成，两者之间采用超文本传送协议 (HTTP) 进行通信。HTTP协议是Web浏览器和Web服务器之间的应用层协议，是通用的、无状态的、面向对象的协议。一个完整的HTTP协议会话过程包括四个步骤：
 连接，Web浏览器与Web服务器建立连接，打开一个称为Socket (套接字) 的虚拟文件，此文件的建立标志着连接建立成功；
 请求，Web浏览器通过Socket向Web服务器提交请求。
 HTTP的请求一般是GET或POST命令 (POST用于FORM参数的传递) ；
 应答，Web浏览器提交请求后，通过HTTP 协议传送给Web服务器。Web服务器接到后，进行事务处理，处理结果又通过HTTP传回给Web浏览器，从而在Web浏览器上显示出所请求的页面；
 关闭连接，应答结束后Web浏览器与Web服务器必须断开，以保证其它Web浏览器能够与Web服务器建立连接。Java实现Web服务器功能的程序设计

编程思路 根据上述HTTP协议的会话过程，本实例中实现了GET请求的Web服务器程序的方法,方法如下：通过创建ServerSocket 类对象，侦听用户指定的端口(为8080)，等待并接受客户机请求到端口。创建与Socket相关联的输入流和输出流，然后读取客户机的请求信息。若请求类型是GET，则从请求信息中获取所访问的HTML 文件名；如果HTML 文件存在，则打开HTML 文件，把HTTP 头信息和HTML 文件内容通过Socket 传回给Web浏览器，然后关闭文件，否则发送错误信息给Web 浏览器。最后关闭与相应Web 浏览器连接的Socket。用Java编写Web服务器httpServer.java文件的源代码如下：

```
//httpServer.java import java.net.*.import java.io.*.import java.util.*.import java.lang.*.public class httpServer{ public static void main(String args[]) { int port. ServerSocket server_socket. //读取服务器端口号 try { port = Integer.parseInt(args[0]). } catch (Exception e) { port = 8080. } try { //监听服务器端口，等待连接请求 server_socket = new ServerSocket(port). System.out.println("httpServer running on port " server_socket.getLocalPort()). //显示启动信息 while(true) { Socket socket = server_socket.accept(). System.out.println("New connection accepted " socket.getInetAddress() ":" socket.getPort()). //创建分线程 try { httpRequestHandler request = new httpRequestHandler(socket). Thread thread = new Thread(request). //启动线程 thread.start(). } catch(Exception e) { System.out.println(e). } } } catch (IOException e) { System.out.println(e). } }}class httpRequestHandler implements Runnable{ final static String CRLF = "\r\n". Socket socket.
```

```
InputStream input. OutputStream output. BufferedReader br. // 构造方法
public httpRequestHandler(Socket socket) throws Exception {
    this.socket = socket;
    this.input = socket.getInputStream();
    this.output = socket.getOutputStream();
    this.br = new BufferedReader(new InputStreamReader(socket.getInputStream()));
} // 实现Runnable接口的run()方法
public void run() {
    try {
        processRequest();
    } catch (Exception e) {
        System.out.println(e);
    }
}
private void processRequest() throws Exception {
    while(true) {
        //读取并显示Web 浏览器提交的请求信息
        String headerLine = br.readLine();
        System.out.println("The client request is " + headerLine);
        if(headerLine.equals(CRLF) || headerLine.equals("")) break;
        StringTokenizer s = new StringTokenizer(headerLine);
        String temp = s.nextToken();
        if(temp.equals("GET")) {
            String fileName = s.nextToken();
            fileName = "." + fileName; // 打开所请求的文件
            FileInputStream fis = null;
            boolean fileExists = true;
            try {
                fis = new FileInputStream(fileName);
            } catch (FileNotFoundException e) {
                fileExists = false;
            } // 完成回应消息
            String serverLine = "Server: a simple java httpServer";
            String statusLine = null;
            String contentTypeLine = null;
            String entityBody = null;
            String contentLengthLine = "error";
            if (fileExists) {
                statusLine = "HTTP/1.0 200 OK" + CRLF;
                contentTypeLine = "Content-type: " + contentType(fileName) + CRLF;
                contentLengthLine = "Content-Length: " + (new Integer(fis.available())).toString() + CRLF;
            } else {
                statusLine = "HTTP/1.0 404 Not Found" + CRLF;
                contentTypeLine = "text/html";
                entityBody = "" + "404 Not Found"
            }
        }
    }
}
```

```
"404 Not Found" "usage:http://yourHostName:port/"
"fileName.html" . } // 发送到服务器信息
output.write(statusLine.getBytes()).
output.write(serverLine.getBytes()).
output.write(contentTypeLine.getBytes()).
output.write(contentLengthLine.getBytes()).
output.write(CRLF.getBytes()). // 发送信息内容 if (fileExists) {
sendBytes(fis, output) . fis.close(). } else {
output.write(entityBody.getBytes()). } } //关闭套接字和流 try {
output.close(). br.close(). socket.close(). } catch(Exception e) {} }
private static void sendBytes(FileInputStream fis, OutputStream os)
throws Exception { // 创建一个 1K buffer byte[] buffer = new
byte[1024] . int bytes = 0 . // 将文件输出到套接字输出流中 while
((bytes = fis.read(buffer)) != -1 ) { os.write(buffer, 0, bytes). } }
private static String contentType(String fileName) { if
(fileName.endsWith(".htm") || fileName.endsWith(".html")) {
return "text/html". } return "fileName". }} 100Test 下载频道开通 ,
各类考试题目直接下载。 详细请访问 www.100test.com
```