如何用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_ti2020/144/2021_2022__E5_A6_82_E4_BD_95_E7_94_A8J_c104_144335.htm 一、HTTP协议的作用原理 WWW是以Internet作为传输媒介的一个应用系统，WWW网上最基本的传输单位是Web网页。WWW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协议是基于TCP/IP协议之上的协议，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。HTTP协议的作用原理包括四个步骤：（1） 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。（2） 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）。GET命令的格式为：GET 路径/文件名 HTTP/1.0 文件名指出所访问的文件，HTTP/1.0指出Web浏览器使用的HTTP版本。（3） 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。 例：假设客户机与www.mycompany.com：8080/mydir/index.html建立了连接，就会发送GET命令：GET /mydir/index.html HTTP/1.0.主机名为www.mycompany.com的Web服务器从它的文档空间中搜索子目录mydir的文件index.html.如果找到该文件，Web服务器

把该文件内容传送给相应的Web浏览器。 为了告知 Web浏览器传送内容的类型，Web服务器首先传送一些HTTP头信息，然后传送具体内容（即HTTP体信息），HTTP头信息和HTTP体信息之间用一个空行分开。 常用的HTTP头信息有：① HTTP 1.0 200 OK 这是Web服务器应答的第一行，列出服务器正在运行的HTTP版本号和应答代码。代码“ 200 OK ”表示请求完成。② MIME_Version：1.0 它指示MIME类型的版本。③ content_type：类型 这个头信息非常重要，它指示HTTP体信息的MIME类型。如：content_type：text/html指示传送的数据是HTML文档。④ content_length：长度值 它指示HTTP体信息的长度（字节）。（4） 关闭连接：当应答结束后，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。 二、Java实现Web服务器功能的程序设计 根据上述HTTP协议的作用原理，实现GET请求的Web服务器程序的方法如下：（1） 创建ServerSocket类对象，监听端口8080.这是为了区别于HTTP的标准TCP/IP端口80而取的；（2） 等待、接受客户机连接到端口8080，得到与客户机连接的socket；（3） 创建与socket字相关联的输入流instream和输出流outstream；（4） 从与socket关联的输入流instream中读取一行客户机提交的请求信息，请求信息的格式为：GET 路径/文件名 HTTP/1.0（5） 从请求信息中获取请求类型。如果请求类型是GET，则从请求信息中获取所访问的HTML文件名。没有HTML文件名时，则以index.html作为文件名；（6） 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；（7）

关闭与相应Web浏览器连接的socket字。 下面的程序是根据上述方法编写的、可实现多线程的Web服务器，以保证多个客户机能同时与该Web服务器连接。 程序1：WebServer.java文件 //WebServer.java 用JAVA编写Web服务器 import java.io.*； import java.net.*； public class WebServer { public static void main （String args[]） { int i=1， PORT=8080； ServerSocket server=null； Socket client=null； try { server=new ServerSocket （PORT）； System.out.println（"Web Server is listening on port " server.getLocalPort（））； for （；；） {client=server.accept （）； //接受客户机的连接请求 new ConnectionThread（client，i）。start（）；i；}} catch （Exception e） {System.out.println（e）；}}} /* ConnnectionThread类完成与一个Web浏览器的通信 */ class ConnectionThread extends Thread { Socket client； //连接Web浏览器的socket字 int counter； //计数器 public ConnectionThread（Socket cl，int c） { client=cl； counter=c；} public void run（） //线程体 {try { String destIP=client.getInetAddress（）。toString（）； //客户机IP地址 int destport=client.getPort（）； //客户机端口号 System.out.println（"Connection " counter "：connected to " destIP " on port " destport "."）； PrintStream outstream=new PrintStream（client.getOutputStream（））； DataInputStream instream=new DataInputStream（client.getInputStream（））； String inline=instream.readLine（）； //读取Web浏览器提交的请求信息 System.out.println（"Received：" inline）； if （getrequest（inline））{ //如果是GET请求 String filename=getfilename（inline）； File file=new File（filename）；

```
if （file.exists（））{//若文件存在，则将文件送给Web浏览器
System.out.println（filename＂requested.）；outstream.println
（"HTTP/1.0 200 OK"）；outstream.println（"MIME_version
：1.0"）；outstream.println（"Content_Type：text/html"）；
int len=（int）file.length（）；outstream.println
（"Content_Length：＂len）；outstream.println（""）；sendfile
（outstream，file）；//发送文件 outstream.flush（）；}else{
//文件不存在时 String notfound=" Error 404-file not found"；
outstream.println（"HTTP/1.0 404 no found"）；
outstream.println（"Content_Type：text/html"）；
outstream.println（"Content_Length：＂notfound.length（）2）
；outstream.println（""）；outstream.println（notfound）；
outstream.flush（）；}}long m1=1；while（m1 client.close（
）；}catch（IOException e）{System.out.println（"Exception
：＂e）；}}/* 获取请求类型是否为“ GET ”*/ boolean
getrequest（String s）{if（s.length（）>0）{if（s.substring（0
，3）。equalsIgnoreCase（"GET"））return true；}return false
；}/* 获取要访问的文件名 */ String getfilename（String s）{
String f=s.substring（s.indexOf（＇ ＇）1）；f=f.substring（0
，f.indexOf（＇ ＇））；try{if（f.charAt（0）==＇/＇）
f=f.substring（1）；}catch（StringIndexOutOfBoundsException
e）{System.out.println（"Exception：＂e）；}if（f.equals（""
））f="index.html"；return f；}/*把指定文件发送给Web浏览
器 */ void sendfile（PrintStream outs，File file）{try{
DataInputStream in=new DataInputStream（new FileInputStream
（file））；int len=（int）file.length（）；byte buf[]=new
```

byte[len]；in.readFully（buf）；outs.write（buf，0，len）；outs.flush（）；in.close（）；}catch（Exception e）{ System.out.println（"Error retrieving file."）；System.exit（1）；}}} 程序中的ConnectionThread线程子类用来分析一个Web浏览器提交的请求，并将应答信息传回给Web浏览器。其中，getrequest（）方法用来检测客户的请求是否为“ GET ”；getfilename（s）方法是从客户请求信息s中获取要访问的HTML文件名；sendfile（）方法把指定文件内容通过socket传回给Web浏览器。 对上述程序的getrequest（）方法和相关部分作修改，也能对POST请求进行处理。 三、运行实例 为了测试上述程序的正确性，将编译后的WebServer.class、ConnectionThread.class和下面的index.html文件置于网络的某台主机的同一目录中（如：主机NT40SRV的C：JWEB目录）。 程序2：index.html文件这是用JAVA写出的WEB服务器主页 1998年8月28日 首先在该主机上用java命令运行WebServer.class： C：jweb>java webserver 然后在客户机运行浏览器软件，在URL处输入WebServer程序所属的URL地址（如：http://nt40srv：8080/index.html），就在浏览器窗口显示出指定的HTML文档。 注意，不能缺省端口号8080，如缺省，则运行该主机的正常WEB服务器。 说明，不具备网络条件的可在安装了Windows 95的单机上进行测试，方法是用localhost或127.0.0.1代替URL地址的域名部分，即URL地址为http://localhost：8080. 