转 JAVA编写的一个简单的Socket实现的HTTP响应服务器

2013年09月24日 15:45:59　阅读数：6932

JAVA编写的一个简单的Socket实现的HTTP响应服务器，看后就很容易理解Web服务器的原理了。

```java
package test.io;

import java.net.*;
import java.io.*;

/**
 * 一个简单的Socket实现的HTTP响应服务器。<br>
 * 只用于熟悉HTTP协议的目的，可以看到浏览器发过来的数据格式。
 *
 * @author  */
public class MyWebServer {
  public static void main(String[] args) {
    Socket socket = null;
    try {
      // 创建一个监听8000端口的服务器Socket
      ServerSocket s = new ServerSocket(8000, 3);
      System.out.println("MyWebServer等待来自浏览器的连接\n");
      while (true) {
        socket = s.accept();
        System.out.println("连接已建立。端口号：" + socket.getPort());
        new MyWebServerThread(socket).start();
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}

class MyWebServerThread extends Thread {
  private Socket socket;

  MyWebServerThread(Socket socket) {
    this.socket = socket;
  }

  @Override
  public void run() {
    try {
      InputStreamReader is = new InputStreamReader(socket.getInputStream());
      char[] bs = new char[2048];
      PrintStream out;
      out = new PrintStream(socket.getOutputStream());
      StringBuilder msg = new StringBuilder();
      // 如果10毫秒还没有数据，则视同没有新的数据了。
      // 因为有Keep-Alive的缘故，浏览器可能不主动断开连接的。
      // 实际应用，会根据协议第一行是GET还是 POST确定。
      socket.setSoTimeout(10);
      //
      // 此处读入请求数据并做相应的处理
      //
      int len = -1;
      try {
        while ((len = is.read(bs)) != -1) {
          msg.append(bs, 0, len);
          msg.append("\n");
        }
      } catch (Exception ex) {
        // ex.printStackTrace();
      }
      // 下面是由服务器直接生成的主页内容
      // 1、首先向浏览器输出响应头信息
      out.println("HTTP/1.1 200 OK");
      out.println("Content-Type:text/html;charset:GBK");
      out.println();
      // 2、输出主页信息
      out
          .println("<HTML><BODY>"
              + "<center>"
              + "<H1>HTTP协议测试服务器,当前时间："
              + new java.util.Date()
              + "</h1>"
              + "<form method='get'>username:<input type='text' name='username'/>password:<input type='text' name='password'/><input type='submit' value='GET测试'/></form><br/>"
              + "<form method='post'>username:<input type='text' name='username'/>password:<input type='text' name='password'/><input type='submit' value='POST测试'/></form><br/>"
              + "<form method='post'  enctype='multipart/form-data'>phototitle:<input type='text' name='phototitle'/>photo:<input type='file' name='photo'/><input type='submit' value='Upload测试'/></form>"
              + "</center>您提交的数据如下:<pre>" + msg.toString() + "</pre></BODY></HTML>");
      out.flush();
      out.close();
      is.close();
      System.out.println("close");
      // 关闭连接
      socket.close();
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

文章标签： JAVA Socket HTTP 服务器

个人分类： J2EE 计算机网络