|  |  |
| --- | --- |
|  | PROJECT 1: Building a simple Web Client and a Multithreaded Web Server |
|  | [Author name] |
|  | 2198-CSE-5335-002 Computer Networks  12/3/19 |

**PROJECT 1**

**CODE DOCUMENTATION**

**WEBSERVER.JAVA**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* NAME: VYSALI PUGHAZHENDI

\* STUDENT ID: 1001750473

\* COURSE # : 2198-CSE-5335-003

\* COURSE NAME: COMPUTER NETWORKS

\* PROJECT 1: Building a Simple Web Client and a Multithreaded Web Server

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class WebServer

{

static Socket *clientSocketObj* = null;

static ServerSocket *serverSocketObj* = null;

static int *port* = 80;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Main Class

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public static void main(String args[])

{

try

{

*serverSocketObj* = new ServerSocket(*port*);

}

catch (Exception e)

{

System.*out*.println(e);

}

while (true)

{

try

{

// Server accepts the client connection.

*clientSocketObj* = *serverSocketObj*.accept();

System.*out*.println("Accepted Client Connection");

while (true)

{

Date date = new Date();

String response = "Response Successful HTTP/1.1 200 OK \r\n\r\n" + date;

*clientSocketObj*.getOutputStream().write(response.getBytes("UTF-8"));

// Request Message

// Object creation for HTTPRequest

HttpRequest httpRequest = new HttpRequest(*clientSocketObj*);

Thread thread = new Thread(httpRequest);

thread.start();

// Thread sleeps for 10000 millisecs. - to retain the connection active for a few more mins.

thread.*sleep*(10000);

}

}

//Exception Handling

catch (Exception ex)

{

System.*out*.println(ex);

}

}

}

}

**WEBCLIENT.JAVA**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* NAME: VYSALI PUGHAZHENDI

\* STUDENT ID: 1001750473

\* COURSE # : 2198-CSE-5335-003

\* COURSE NAME: COMPUTER NETWORKS

\* PROJECT 1: Building a Simple Web Client and a Multithreaded Web Server

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.net.\*;

import java.io.\*;

public class WebClient

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Main Class

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public static void main(String[] args)

{

try

{

// Object Creation for Socket class

// Mention the port #

Socket socket = new Socket("localhost", 8080);

System.*out*.println("Server Connection - Success");

//Object Creation for PrintStream class.

PrintStream printStream = new PrintStream(socket.getOutputStream());

//Prints the below lines after the client is connected to the server.

printStream.println("GET index.html HTTP/1.1\r\n\r\n");

printStream.println("Hello World");

//Flushes the stream.

printStream.flush();

//Closing the connection and socket.

printStream.close();

socket.close();

System.*out*.println("Connection/Socket Closed");

}

// Exception Handling Block

catch (Exception ex)

{

System.*out*.println(ex);

}

}

}

**HTTPREQUEST.JAVA**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* NAME: VYSALI PUGHAZHENDI

\* STUDENT ID: 1001750473

## \* COURSE # : 2198-CSE-5335-003

\* COURSE NAME: COMPUTER NETWORKS

\* PROJECT 1: Building a Simple Web Client and a Multithreaded Web Server

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class HttpRequest implements Runnable {

Socket socketPort;

// Constructor

public HttpRequest(Socket s) throws Exception

{

System.*out*.println("socket #:" + s);

this.socketPort = s;

}

public void run()

{

try

{ //Calling Process Request Method

processRequest();

}

//Exception Handling block

catch (Exception e)

{

//Logs the exceptions

System.*out*.println(e);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* METHOD NAME: processRequest()

\* Throws: Exception base class

\* DESCRIPTION: This method gets the input request

\* and reads the input file. This method reads and

\* writes the content of the input request file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public void processRequest() throws Exception

{

//OBJECT CREATION

//Object creation for InputStream class

InputStream inputStream = socketPort.getInputStream();

//Object Creation for DataOutputStream class

DataOutputStream dataOutputStream = new DataOutputStream(socketPort.getOutputStream());

//Object creation for BufferedReader class

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

//Reads the inputs Request.

String httpReqMessage = bufferedReader.readLine();

System.*out*.println();

System.*out*.println("REQUEST Message :" + httpReqMessage);

String fileRead = bufferedReader.readLine();

while (fileRead != null && fileRead.length() != 0)

{

System.*out*.println(fileRead);

fileRead = bufferedReader.readLine();

}

StringTokenizer tokens = new StringTokenizer(httpReqMessage);

tokens.nextToken();

String fileName = tokens.nextToken();

fileName = "." + fileName;

//Input File - Read & Write.

FileInputStream fileInputStream = null;

boolean IsFileFound = true;

try

{

fileInputStream = new FileInputStream(fileName);

}

//Exception Handling block

catch (Exception ex)

{

IsFileFound = false;

System.*out*.println(ex);

}

// Response Message

String message = null;

String messageContent = null;

String messageBody = null;

if (IsFileFound)

{

message = "File is found - 200 Success";

messageContent = "Content-type: " + messageContentType(fileName) + "\r\n";

}

else

{

message = "HTTP/1.1 404 Not Found";

messageContent = "File Not Found or does not exists";

messageBody = "<HTML>" + "<HEAD><TITLE>File Not Found</TITLE></HEAD>" + "<BODY>Not Found</BODY></HTML>";

}

if (IsFileFound)

{

//Calling File reading method

fileRead(fileInputStream, dataOutputStream);

fileInputStream.close();

}

else

{

dataOutputStream.writeBytes(messageBody);

}

dataOutputStream.writeBytes("\r\n");

//Closing the connection and socket.

dataOutputStream.close();

bufferedReader.close();

socketPort.close();

System.*out*.println("Connection/Socket Closed");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* METHOD NAME: messageContentType()

\* Parameters: filename

\* DESCRIPTION: This method writes message in the input file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public String messageContentType(String fileName)

{

if (fileName.endsWith(".htm") || fileName.endsWith(".html")) {

return "text/html";

}

return "application/octet-stream";

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* METHOD NAME: fileRead()

\* Throws: Exception base class

\* Parameters: InputFileStream, Output

\* DESCRIPTION: This method reads the input file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public void fileRead(FileInputStream fileInputStream, OutputStream output)

throws Exception

{

int i = 0;

//object creation.

byte[] bytes = new byte[1024];

while ((i = fileInputStream.read(bytes)) != -1)

{

//writes the input file content.

output.write(bytes, 0, i);

}

}

}

**INDEX.HTML**

Building a Simple Web Client and a Multithreaded Web Server.

Objectives:

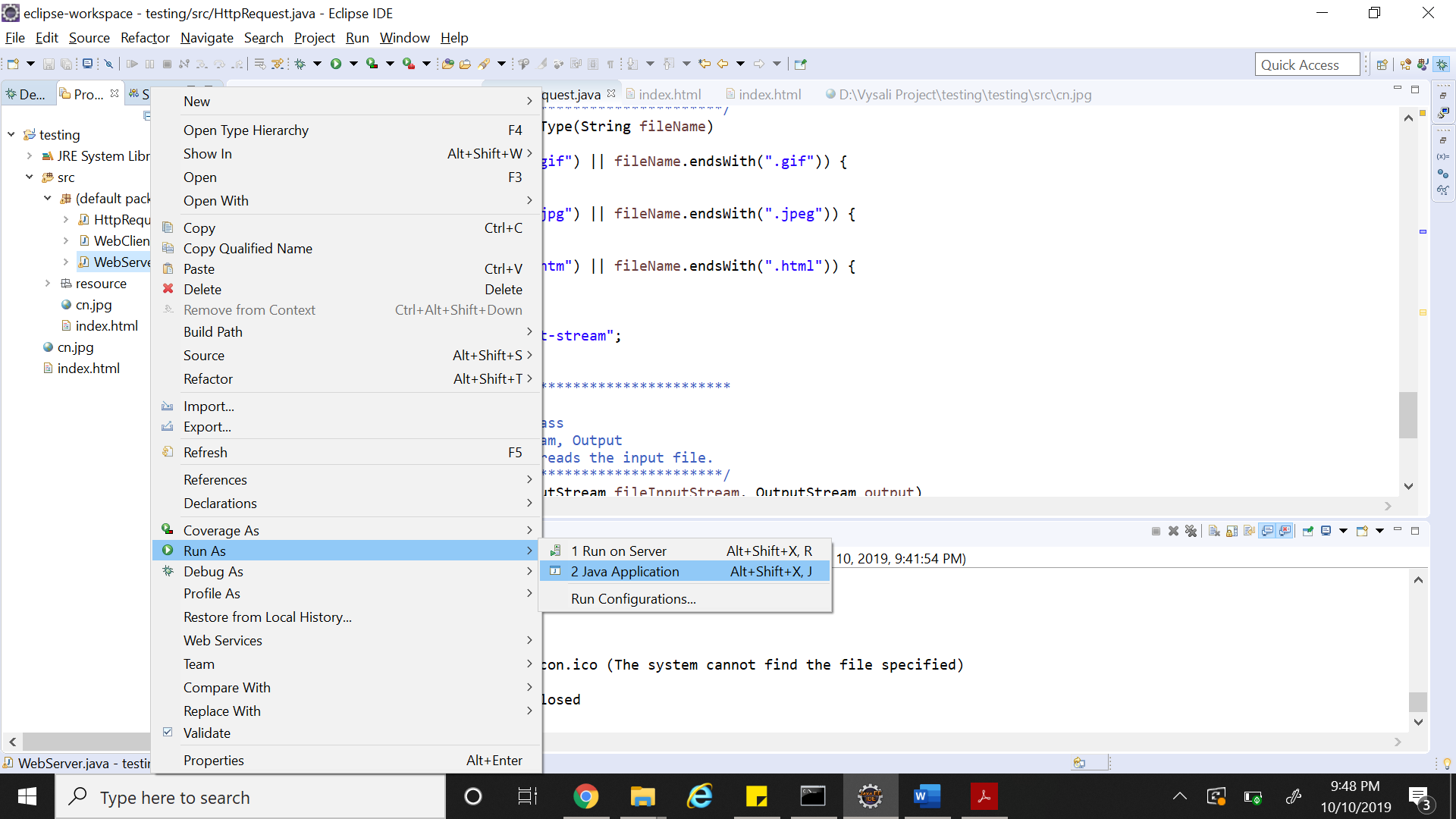
To understand client-server communication via sockets.

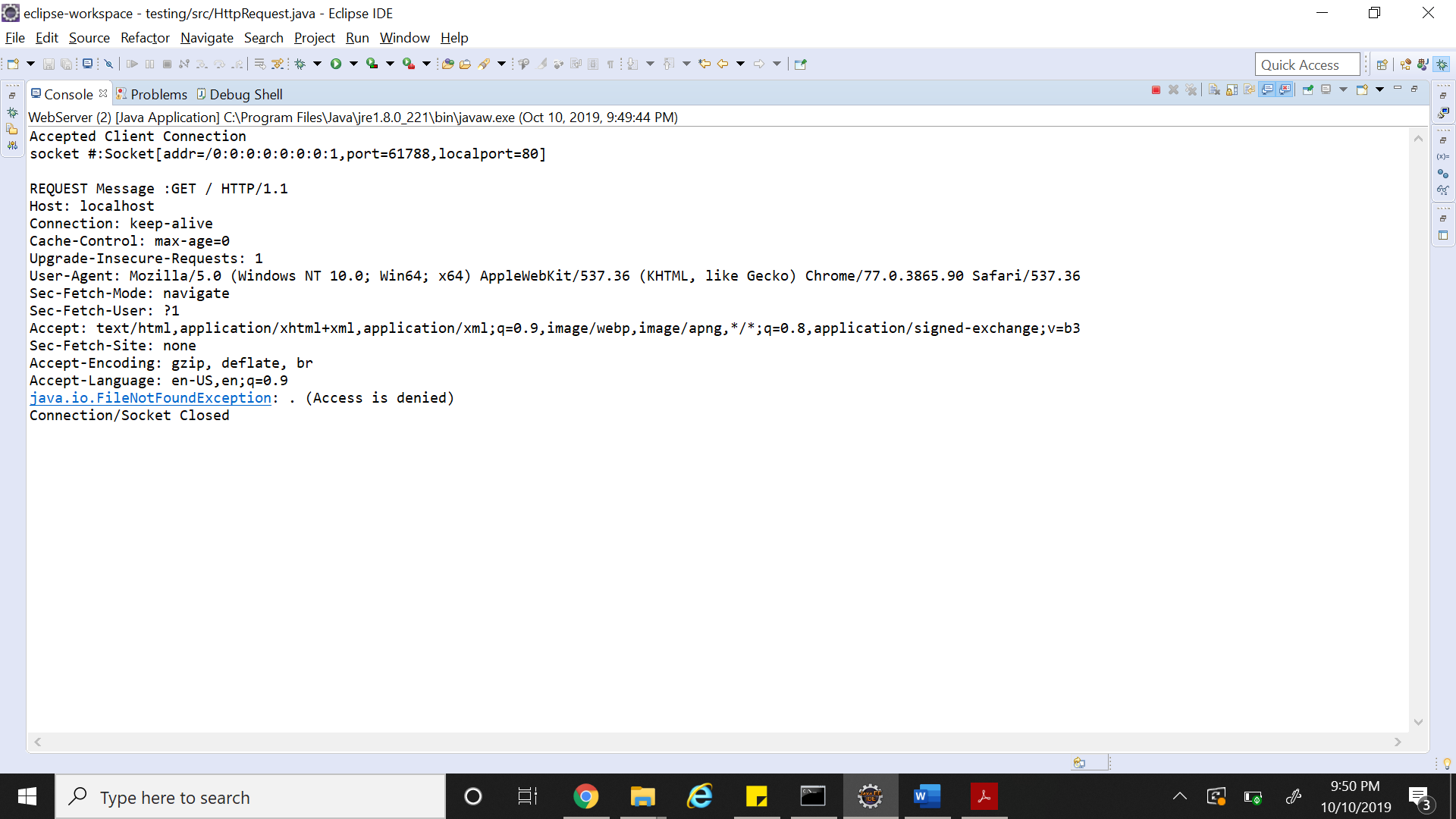
To gain exposure to the basic operations of a Web server and client.

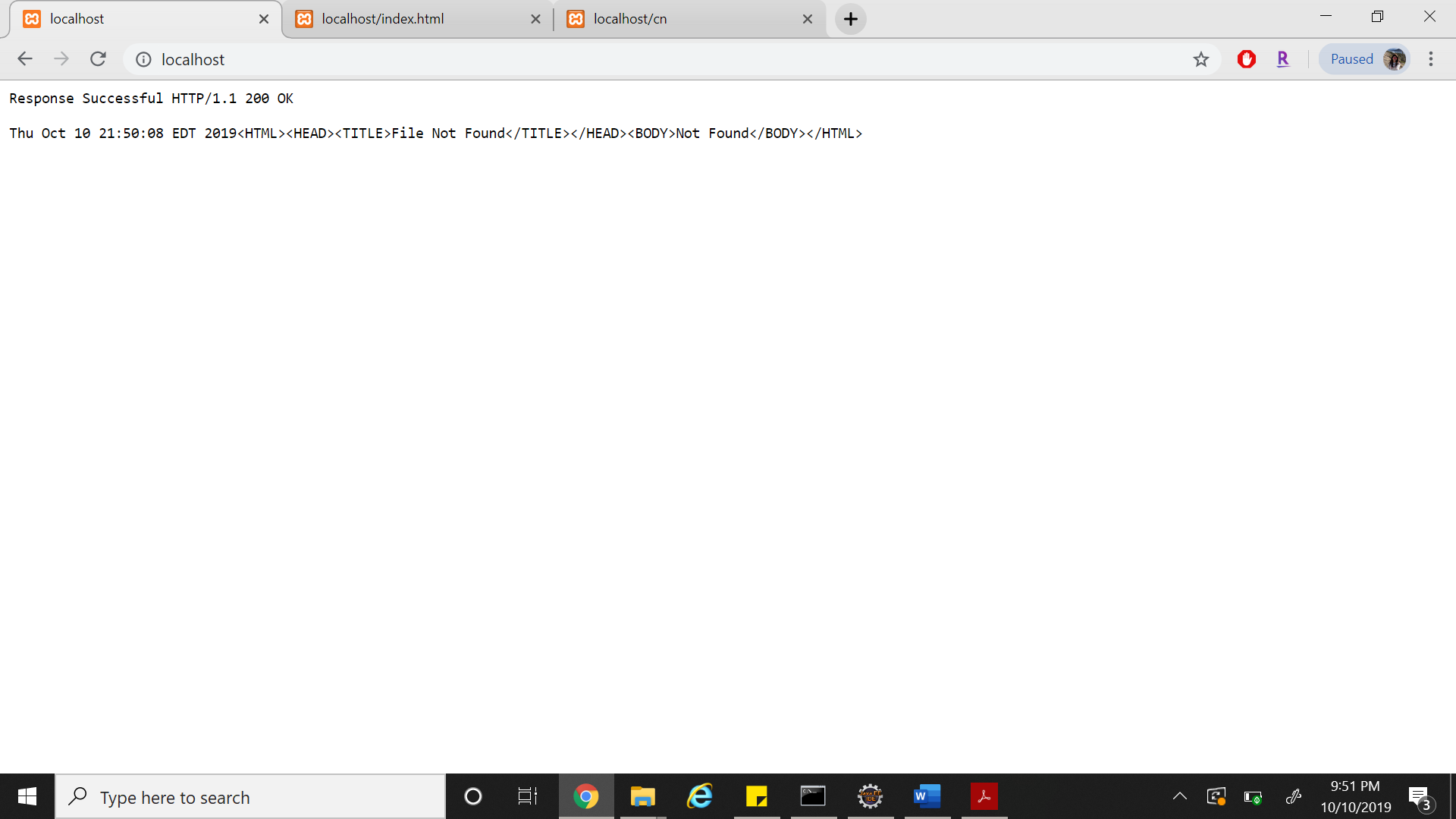
To explore basic structures of HTTP messages.

**OUTPUT:**

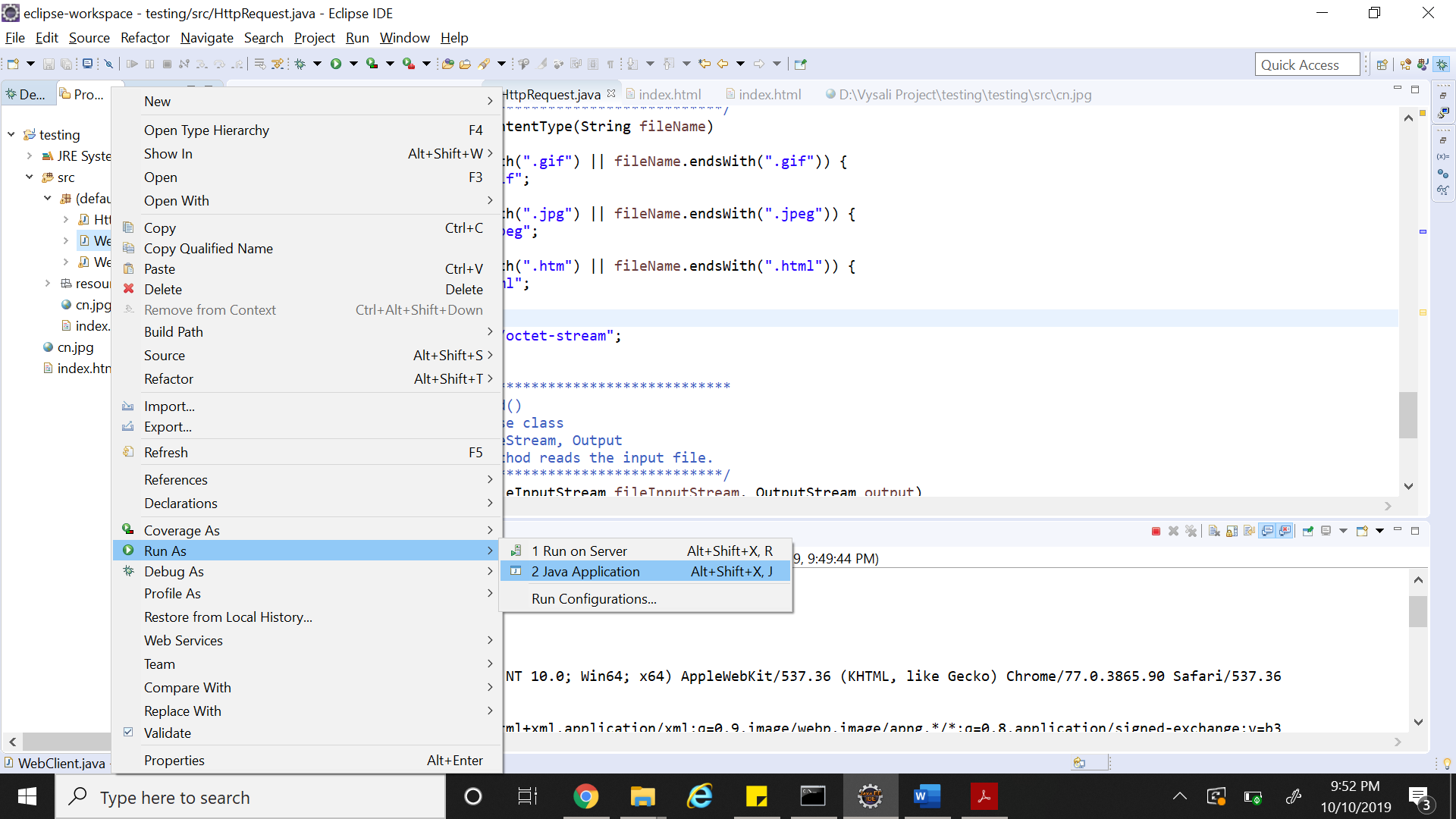
Step 1: Run Webserver.java

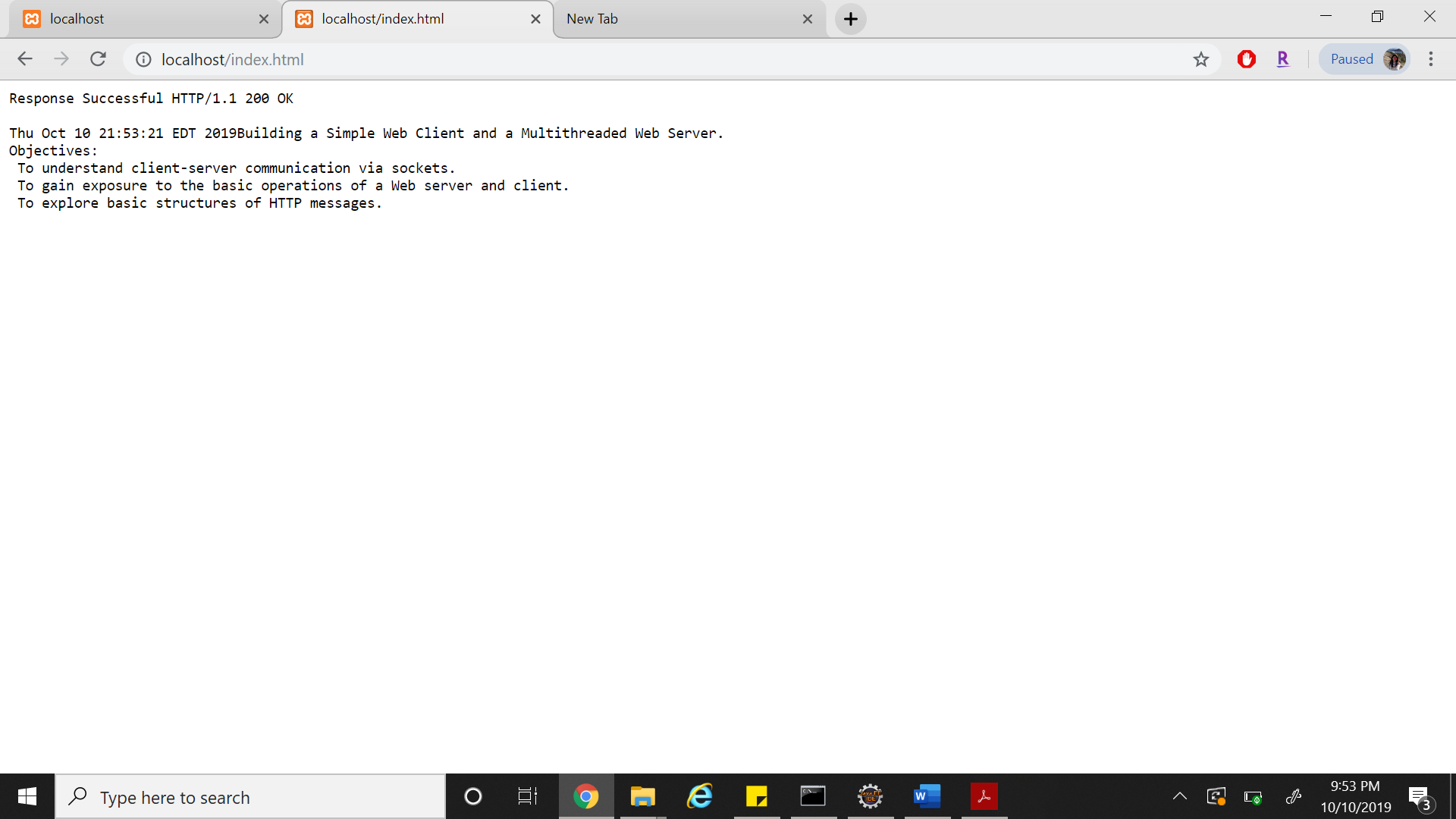






Step 2: Run Webclient.java





**REFERENCES:**

1. <https://www.codeproject.com/Articles/452052/Build-Your-Own-Web-Server>
2. <https://javarevisited.blogspot.com/2015/06/how-to-create-http-server-in-java-serversocket-example.html>