

### Question 1)

The error message we get is "Exception in thread "main" java.net.SocketException: Connection reset" followed by a call stack. Before the error we are able to successfully get the correct responses for "GET", "HEAD", "POST" and "BLAHA". Is this an issue that has been raised by any other groups or something that has been seen in previous years?

Solution>

You can use

```
socket.shutdownInput();  
  
    socket.shutdownOutput();
```

before close the socket

### Question 2)

Are we supposed to implement the solution in such way that the "last modified" date actually is the "last modified" date of the file or are we just going to let it be a string?

We have the same question about the body in the servers reply, is it supposed to be an actual file that we get or just a string?

Should the solution be some universal solution or should it only cover the files/pages in the lab description?

Solution>

The "last modified date" must be the real last modified date of the file.

In the body entity is the actual content of the required file if the method "GET" is used and the file exists in the server's directory.

You can see in the sample output file that some requests are sent to acquire the content of the files "index.html", "empty.html", "nonexisting.text",... Based on the context of the files' names, you should create files accordingly. For example, you should create a file named "index.html" and save it in the same folder as the server's, file "empty.html" is an empty html file named "empty.html", "nonexisting" means there must not be such file in the server's directory, etc.

### Question 3)

Nothing works to get rid of "Exception in thread "main" java.net.SocketException: Connection reset". What to do?

Solution>

Use Different machines to run your WebServer and WebServerTest. Way to do that -

For ex, run Webserver on your laptop and know its ip address (say 129.16.138.192) and the port number displayed (say 52341) by the WebServer running.

(To know the ip address in windows use the command 'ipconfig' at command prompt and note the ipv4 address whereas on linux the command is 'ifconfig' and note the inet addr of eth0.)

Run WebServerTest on another machine passing the arguments as '-p 52341 -h 129.16.138.192' (without any quotes) .

Passing arguments at command line is straightforward whereas if you are using some IDE like eclipse or NetBeans then you should go to run configuration and pass the arguments. For example, in eclipse,

Run>Run Configuration

Select the correct file on the left side list of available java applications

Highlight the tab "Arguments" and write '-p 52341 -h 129.16.138.192' (without any quotes).

It should run without any difficulty.

### Question 4)

Should there be some response code to say that a file does not exist? How to implement that?

Solution>

Yes, there should be.

To implement that you can use either of the following two options :

1. When you create a file pointer to open the file requested by the GET method and the pointer comes out to be NULL then you can send a response with appropriate response code to the client that the

requested file does not exist.

2. You can use explicit file search commands like 'locate' on unix platform and 'locate32' on windows platform to search for the requested file before using the file pointer and give the response code accordingly. You might google or see man pages to know more about these commands.

#### Question 5)

Are we supposed to implement conditional GET? I can read in the instructions that is not necessary but in the test cases is included what leads me to confusion.

#### Solution>

No, you don't have to, but if you are interested and want to learn more, you can try to implement the conditional "GET", and can test your server by the test case.