

Project: CLIENT-SERVER TCP/IP WEB SERVER

Developer & Publisher: ROHIT CHANDA

Date: October 12, 2022

Description

The goal of this project is to build a functional client-server TCP/IP based web server. This project will throw light upon the basics of Distributed Programming, client/server architectures and structures, and issues in building high performance servers. While we need to focus on the concepts of Distributed Systems that enable network communication, it is also important to understand the structure of these systems that make use of the global Internet.

The project basically deals with a client/server architecture based on the concepts of TCP/IP Protocol wherein the client requests for a particular web page and the server responds accordingly. If the web page is available, then the server sends it in its response. Otherwise, it generates appropriate error messages for the client to understand.

Working and Understanding of the Code

In this project, I have built a fully functional executable Web Server which listens to HTTP requests made by clients through a port. There are 2 arguments that need to be provided by the client through the console to create a successful HTTP request: ***documentRoot*** and the ***port***. These are ***args[1]*** and ***args[3]*** respectively which can be provided while running the ***Client*** and ***Server*** interfaces. The Web Server is **multithreaded** in nature which essentially means that it accepts multiple requests and hands them out to individual threads to process them. This is conceived by implementing the ***Runnable interface*** for processing the requests and starting new unique threads as well. I have implemented the go-to ***HTTP Status Codes*** such as 404, 405, 200, 403, etc. for dealing with the specific situations.

The Web Server is designed to accept and process requests made through the ***GET*** method and thus throws the 405 Error when it encounters a request made through an invalid method (any other request method invalid in our scenario). The server accesses ***InputStream*** from socket and reads the HTTP request to get the request method and file. The Web Server responds with respective status codes 200, 400, 403, 404, 405 depending on the request. It also sends Content-Type, Content-Length and Date headers with every response. Each request is not blocked by other requests, and runs on own thread, thereby independent of other requests. When index.html is not submitted with the request, the server responds with the default index.html. The socket is closed after response is sent to the client via ***OutputStream***.

List of Files

1. Readme.pdf file
2. Pom.XML
3. eclipseConsoleLogs.log
4. Screenshots_WebServer_Rohit_Chanda.pdf file
5. webserver.tar.gz

Instructions for running the Program

To run the jar, use the following in the command line :-

% java -jar WebServer.jar -documentRoot /provideTheLocation/to/the/document/root -port 8888

Example :-

```
% java -jar WebServer.jar "-documentRoot" "/Users/ronnie/Desktop/Projects/com.webserver.pa1.rohitchanda/src/main/resources/document-root/" "-port" "8888"
```