Linnéuniversitetet
Kalmar Växjö

Report

Assignment 2

1DV701

Author: Noa Kron
Michal Stastný
Semester: Spring 2024
Email nk222sj@student.lnu.se
ms227tr@student.lnu.se

Contents

1 Problem 1

1.1 Discussion

The server is implemented using the ServerSocket class. It listens for client connections on a specific port. When a client connects, it creates a Socket object and opens an InputStream to receive data from the client.

The server reads the HTTP request from the client and determines the requested file. If the file exists in the local directory, the server sends a "200 OK" response along with the file content. If the file does not exist, the server sends a "404 Not Found" response. The server also handles a special case where a specific URL is hardcoded to trigger a "302 Found" response.

The server continues to accept and handle client connections until it is manually terminated.

2 Problem 2

2.1 Discussion

The server has been extended to support additional HTTP response types: "302 Found", "404 Not Found", and "500 Internal Server Error". These are implemented as separate methods that send the appropriate HTTP response to the client.

3 Problem 3

3.1 Login

The login is implemented via simple POST request. The server extracts the credentials from the body and compares them to credentials.txt. If they are correct the server sends 200 OK, if they are incorrect the server sends 401 Unauthorized.

3.2 Image upload

The multipart-form parser in our server is able to only parse png images. It extracts the whole request into a buffer and then finds the boundary and the end of the part headers. Then it copies everything in between into a byte array and writes it to a file.
The name of the file is a MD5 hash of the request containing the file.

4 Additional Details

The server also includes a method firstIndexOf which is used to find the first index of a smaller byte array in a larger byte array. This method is used in handling POST requests, specifically when the server needs to parse the body of the request.
In the main method, the server is started by creating a ServerSocket object with the specified port number. The server then enters a loop where it waits for client connections. When a client connects, the server creates a Socket object for the client and opens an InputStream to receive data from the client.

The server reads the HTTP request from the client and determines the HTTP method (GET or POST) and the requested file or action. Depending on the HTTP method and the requested file or action, the server calls the appropriate method to handle the request and send a response to the client.

The server continues to accept and handle client connections until it is manually terminated. If an exception occurs, the server prints an error message and continues to accept and handle client connections. If an unexpected error occurs, the server prints an error message and terminates.

The image upload does not work over Chrome. This is because the way Chrome sends HTTP, it starts with the header and because our code expects to get the whole file size to declare that it is finished we end up with a null value in the split command.

4.1 Collaboration

Noa Kron made all the necessary code for task one. Everything regarding setting up the server, processing GET requests and responding with correct files.

Michal Stastný is responsible for all the code which processes the POST request. In other words the image upload function and login function.

When It comes to the http codes both have equal share of of work.

The git history may not represent this as we often shared code and other media over messaging platforms.

Michal: 50%

Noa: 50%