Test Cases

TC 1.1

Use case: 1 main scenario

Requirement: 3

Scenario: Start the server

test steps

- open the source code in your IDE
- In terminal run: “cd bin”
- In terminal run: “java se.lnu.http.HTTPServerConsole 9000 ~/YourFilePath/MyWebServer-master/bin/se/lnu/http/resources/inner/”

expected

- The following messages is shown in the IDE Terminal:
  “HTTP Server object constructed”
  “HTTP Server started”
  “Accept”

TC 1.2

Use case: 1 main scenario

Requirement: 3

Scenario: Navigate to page

test steps

- Test case 1.1 (start server)
- in your browser search för localhost:9000

expected

- The following messages is shown in the IDE Terminal:

  “ClientThread started nr: 1”

  “ClientThread started nr: 2”

  “ClientThread 2 served file : index.html”

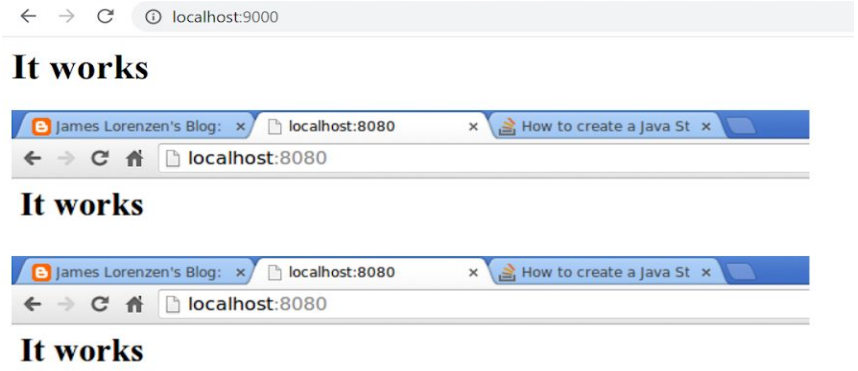
  “ClientThread 2 served file : works.png”

  “ClientThread 1 served file : works2.png”

  “ClientThread stopped nr: 2”

  “ClientThread stopped nr: 1”

- The text “It works” is displayed in your browser
- Two images is shown



TC 1.3

Use case: 1 alternative scenario 4a

Requirement: N/A

Scenario: The server should not be able to start with a taken port

test steps

- Start a server on port 9000 for another project
- Start the server with the taken port 9000

expected

- The system should present the message:
  “Socket 9000 was taken”
- Server stops

TC 1.4

Use case: 1 alternative scenario 4b

Requirement: N/A

Scenario: The web server could not be started due to restriction on the shared resource container

test steps

- Try to start the webserver using a restricted folder path: /resources/

expected

- The system should present the error message:
  “No access to folder /resources”

TC 1.5

Use case: 1 alternative scenario 4c

Requirement: N/A

Scenario: Can not write to access log

test steps

- Start the server without permission to write

expected

- The system should present the error message. “Cannot write to server log file log.txt”

TC 2.1

Use case: 2

Requirement: N/A

Scenario: Stop the server (server needs to be started)

test steps

- In your IDE Terminal write “stop” and press enter.

expected

- The following messages should be shown in your terminal:
  “HTTP Server Accept thread stopped”
  “HTTP Server stopped”

TC 3.1

Use case: 3

Requirement: 2

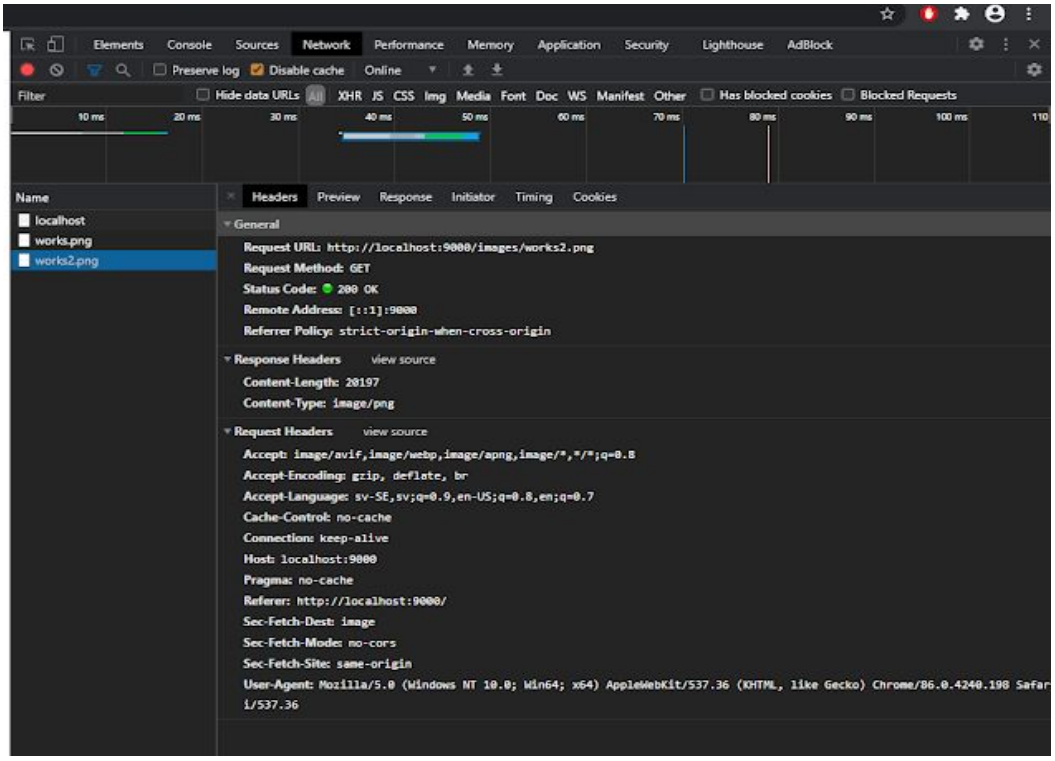
Scenario: Successfully access a shared resource

test steps

- Start the server
- Go to your browser and navigate to the page

expected

- The shared resource is shown in the browser and a success message is written to the access log
- In network section in your browser get the status code 200 OK



TC 3.2

Use case: 3

Requirement: 2

Scenario: A shared resource can not be found

test steps

- Start the server
- Go to your browser and navigate to the page
- Try to access a non existing resource ex “localhost:9000/page”

expected

- The system presents that the resource cannot be found
- In your browser you will receive the message: “404 Not found”

TC 3.3

Use case: 3

Requirement: 2

Scenario: Shared resource is outside the shared resource container

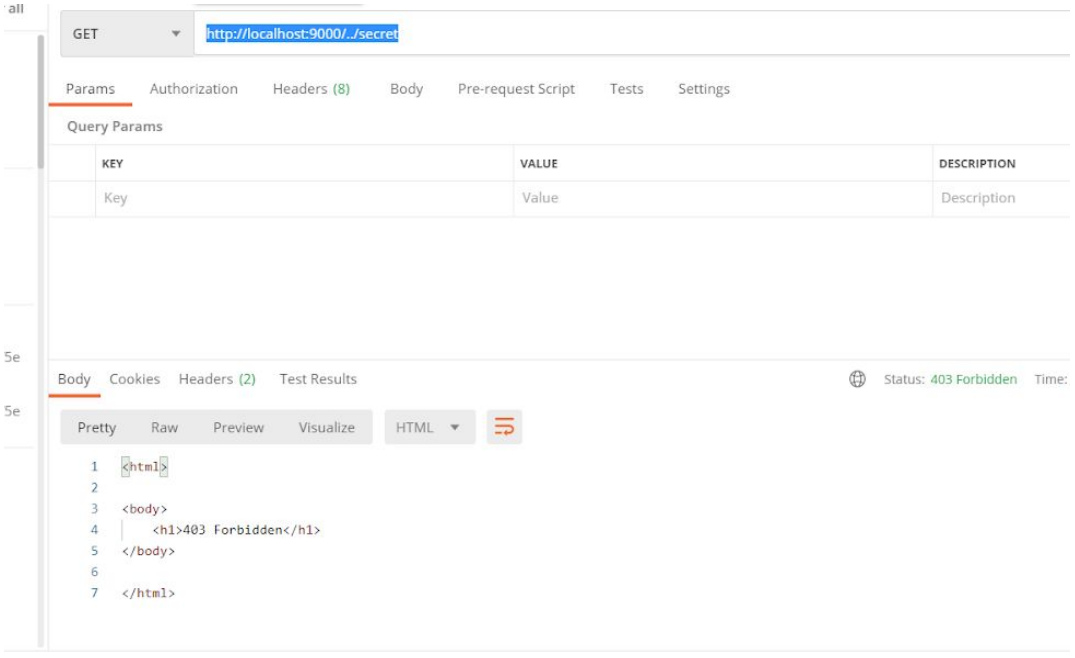
test steps

- Start the server
- Try to access a resource outside of the shared container : “http://localhost:9000/../secret”

expected

- The system presents that the resource is forbidden
- In your browser you will receive the message: “403 Forbidden”

(using postman it looks like below)



TC 3.4

Use case: 3

Requirement: 2

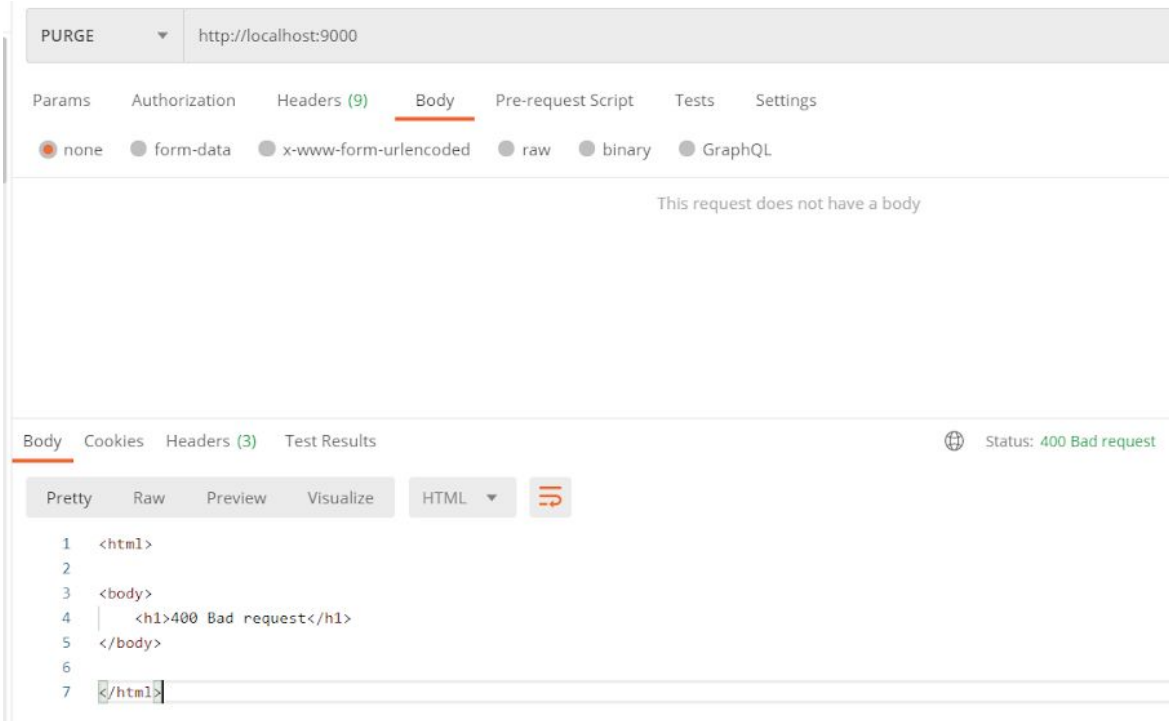
Scenario: Resource request is invalid or malformed

test steps

- Start the server
- Send a Purge request for localhost:9000 using Postman

expected

- The system presents that the request cannot be handled
- You will receive the message: “400 Bad request”



TC 4.1

Use case: N/A

Requirement: 1

Scenario: The web server should be responsive under high load.

test steps

- Test case 1.1(start server)
- Open Jmeter
- Create a Thread group in the Test Plan
- Write 500 in number of threads
- Write ramp-up period of 1 sec
- Add a sampler from thread group htp protocol
- Add a listener from sampler
- Press start test
- Repeat 5 times
- check so no one crashes
- Take the median worth of latency

expected

- No crashes or latency over 200

TC 4.2

Use case: N/A

Requirement: 1

Scenario: The web server should be responsive under high load.

test steps

- Test case 1.1(start server)
- Open Jmeter
- Create a Thread group in the Test Plan
- Write 200 in number of threads
- Write ramp-up period of 1 sec
- Add a sampler from thread group htp protocol
- Add a listener from sampler
- Press start test
- Repeat 5 times
- check so no one crashes
- Take the median worth of latency

expected

- No crashes or latency over 200

TC 5.1

Use case: N/A

Requirement: 5

Scenario: Access log should be viewable from a text editor

test steps

- Test case 1.1 (start server)
- Open the access log in your IDE and read content

expected

- The access log should have some content

TC 6.1

Use case: N/A

Requirement: 4

Scenario: The source code should be released under GPL-2.0

test steps

- Go to your IDE and open the file called LICENCE
- Check content

expected

- Content should be compatible with GPL-2.0

expected content