Test Report

- My web server -
Software Development Company™

Table of contents

Overview

This document presents the results of the tests performed on the “My web server” software. Both automated and manual tests have been executed. More information about these tests is found in the following tables.

Automated Tests

Test ID	Test Description	OS	Date	Status
TJU.1	Integration	Windows*	2017-12-11	Pass
TJU.2	Response	Windows*	2017-12-11	Pass
TJU.3	View	Windows*	2017-12-11	Fail
TJU.4	Grouped	Windows*	2017-12-14	Pass
TV.001	Vulnerability check	Windows*	2017-12-15	Fail

*The tests have been concluded on Windows 10 OS (64-bit).

Manual Tests

Test ID	Test Description	OS	Date	Status
TSS.1.1	Start Server	Windows*	2017-12-11	Pass
TSS.1.2	Wrong Socket	Windows*	2017-12-13	Pass
TSS.1.3	Taken Socket	Windows*	2017-12-13	Pass
TSS.1.4	Access Log Written	Windows*	2017-12-14	Fail
TSS.2.1	Stop Server	Windows*	2017-12-11	Pass
TSS.2.2	Server is stopped - write to access log	Windows*	2017-12-11	Fail
TS.1	HTTP 1.1 Status 200	Windows*	2017-12-12	Pass
TS.2	HTTP 1.1 Status 400	Windows*	2017-12-12	Pass
TS.3	HTTP 1.1 Status 403	Windows*	2017-12-12	Pass
TS.4	HTTP 1.1 Status 404	Windows*	2017-12-12	Pass
TS.5	HTTP 1.1 Status 405	Windows*	2017-12-12	Pass
TL.1	High Load (1 000 users)	Windows*	2017-12-13	Pass
TL.2	High Load (50 000 users)	Windows*	2017-12-13	Pass
TOS.1.1	Test Operating System	Windows*	2017-12-12	Fail
TGPL2.0	Test GPL-2.0	Windows*	2017-12-14	Fail
TB1.1	Test Browser	Windows*	2017-12-15	Pass
TP.1	Performance	Windows*	2017-12-15	Fail
TA.1	Acceptance	Windows*	2017-12-15	Pass

*The tests have been concluded on Windows 10 OS (64-bit).

Assessment

This section provides a description of how the test cases have been executed and their results.

Automated Tests

These automated tests have been performed with the help of the following tools:

- Java IDE (Eclipse Neon or newer)
- Testing frameworks JUnit, Mockito, and Vega

JUnit Tests

Automated tests using JUnit framework in Eclipse IDE.

Integration Test

TJU.1	**Integration**
Requirement	Req 1. The web server should be responsive under high load
Use Case	--
Expected result	All tests pass. A string shows how many times the server is created successfully, fails or throws an exception.
Result	One JUnit test fails. Output in console: fails: 0 oks: 2411 exceptions: 0 total 2411
Status	*passed*

Response Test

TJU.2	Response
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	--
Expected result	All tests pass. Response statuses should pop up as expected.
Result	All tests pass. Response statuses should pop up as expected.
Status	*passed*

View Test

TJU.3	View
Requirement	--
Use Case	--
Expected result	All tests pass. Console shows what is expected from the server.
Result	Test fails when the port is taken. Expected: <Port is taken[]> but was: <Port is taken [ ] > Automated test package tests\se\lnu\http\view;
Status	*failed*

Grouped Test

TJU.4	Grouped
Requirement	--
Use Case	--
Expected result	All tests pass.
Result	All tests pass.
Status	*passed*

Security Test

Automated test performed with the Vega security tool.

TV.001	Vulnerability check
Requirement	--
Use Case	UC1 Start Server
Expected result	No vulnerabilities are found.
Result	No specified character set (such as UTF-8) within the response header/ body was found.
Status	*failed*

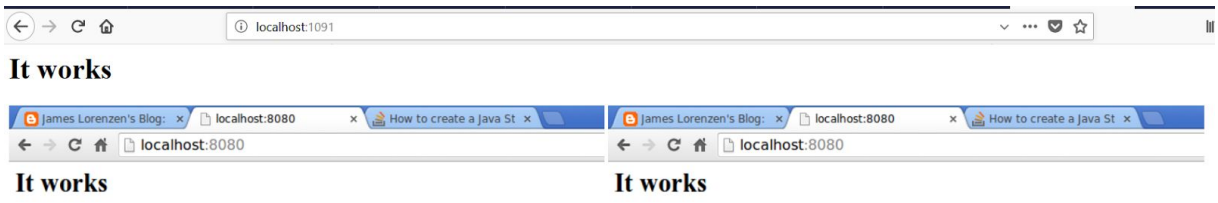
Manual tests

These manual tests have been performed with the aid of the following testing tools:

- jMeter, SmartMeter.io , Postman, and cURL

Start Server

TSS.1.1	Start Server
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC1 Start Server
Expected result	"HTTP Server started" is shown in the terminal window. "It works" image is shown on the web page.
Result	"HTTP Server started" is shown in the terminal window. "It works" image is shown on the web page.
Status	*passed*

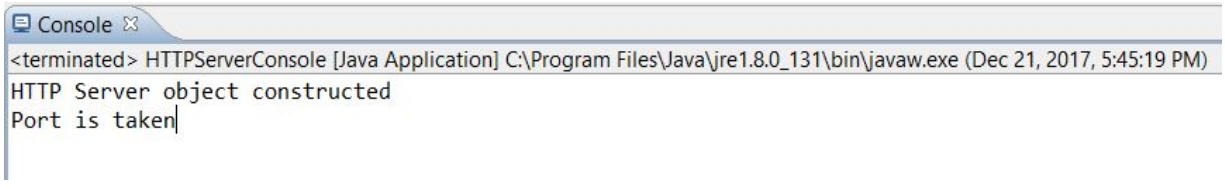


Wrong Socket

TSS.1.2	Wrong Socket
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC1 Start Server
Expected result	“Enter a valid port 1-65535 and a optional URL” is shown in the terminal window.
Result	“Enter a valid port 1-65535 and a optional URL” is shown in the terminal window.
Status	*passed*

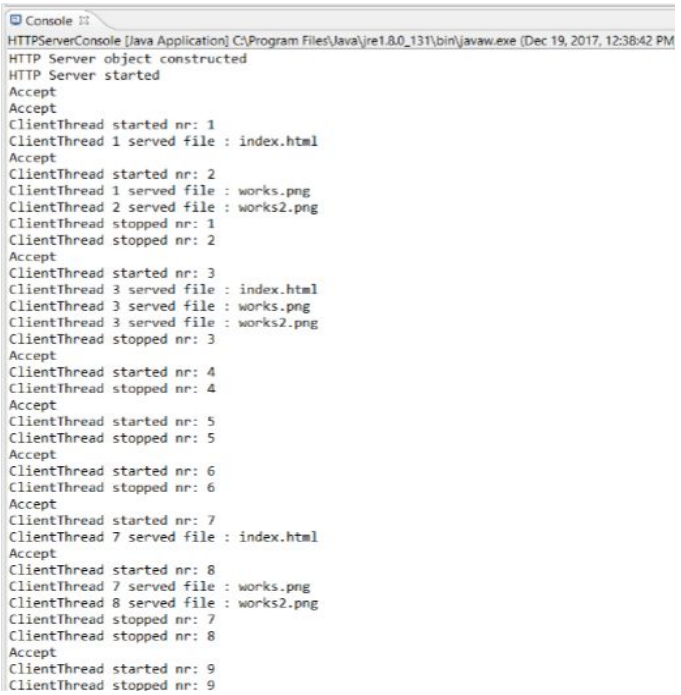
Taken Socket

TSS.1.3	Taken Socket
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC1 Start Server
Expected result	“Port is taken” is shown in the console window.
Result	“Port is taken” is shown in the console window.
Status	*passed*



Access Log written (when request is sent)

TSS.1.4	Access log written
Requirement	Req 5. The access log should be viewable from a text editor.
Use Case	UC1 Start Server
Expected result	Log is created. The access log is viewable in a text-editor.
Result	Access Log is not created.No error message popped up either. The requests are only written in the console.
Status	*failed*



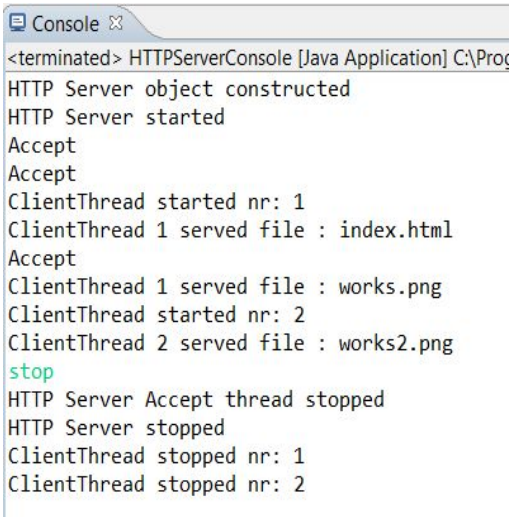
Stopping Server

TSS.2.1	Stop Server
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC2 Stop Server
Expected result	"HTTP Server stopped" is shown in the terminal window. Web page will not load.
Result	"HTTP Server stopped" is shown in the terminal window. Web page will not load.
Status	*passed*

```
C:\I                p>java -jar WebServer.jar 1091 \D:\lnu\java_courses\MyWebServer-master\tests\se\lnu\http\resources\
inner
HTTP Server object constructed
HTTP Server started
Accept
Accept
ClientThread started nr: 1
ClientThread 1 served file : index.html
ClientThread 1 served file : works.png
ClientThread 1 served file : works2.png
stop
HTTP Server Accept thread stopped
HTTP Server stopped
ClientThread stopped nr: 1
```

When Server is stopped – Access log is written

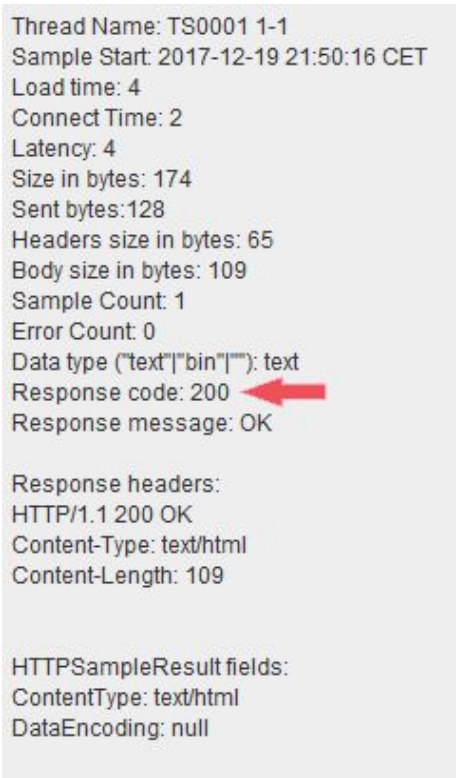
TSS.2.2	Server is stopped –> Access log is written
Requirement	Req 5. The web server must work on Linux, Mac, Windows*.
Use Case	UC2 Stop Server
Expected result	"HTTP Server stopped" is shown in the last line of the log file.
Result	No log file was created, so there is also no "HTTP Server stopped" on the last line. The log shows only in the console.
Status	*failed*



Status Tests (JMeter)

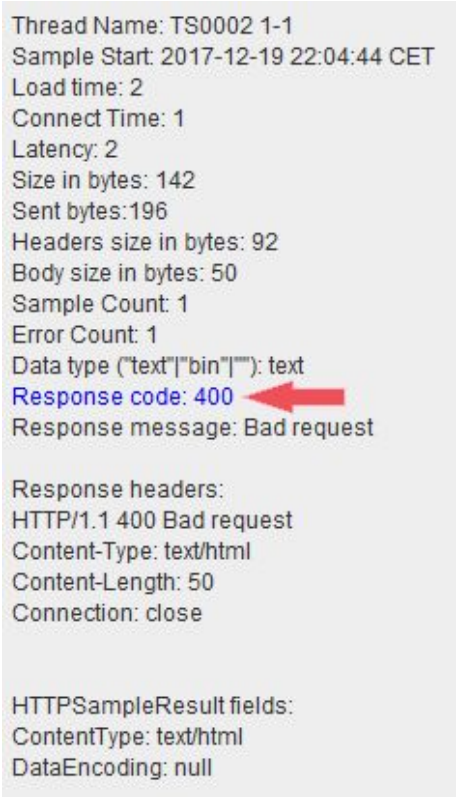
HTTP 1.1 Status 200

TS.1	HTTP 1.1 Status 200
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Expected result	Server responds with status code 200
Result	Server response with status code 200 - OK.
Status	*passed*



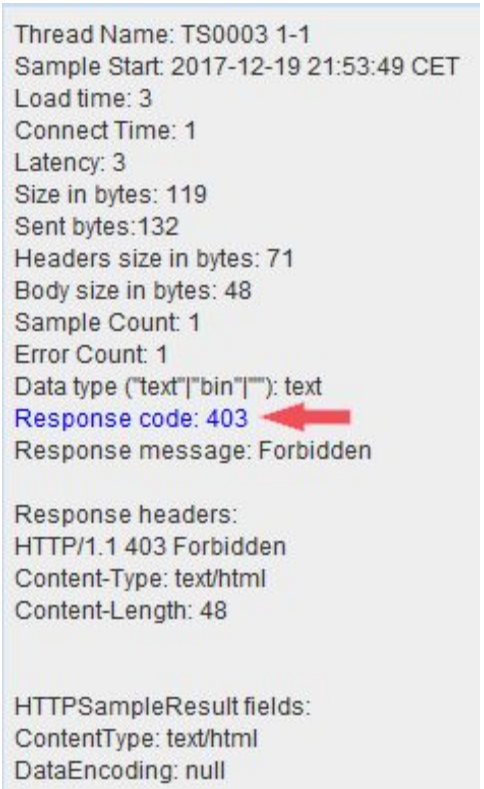
HTTP 1.1 Status 400

TS.2	HTTP 1.1 Status 400
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Expected result	Server responds with status code 400
Result	Server response with status code 400 - Bad request.
Status	*passed*



HTTP 1.1 Status 403

TS.3	HTTP 1.1 Status 403
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Expected result	Server responds with status code 403
Result	Server response with status code 403 - Forbidden.
Status	*passed*

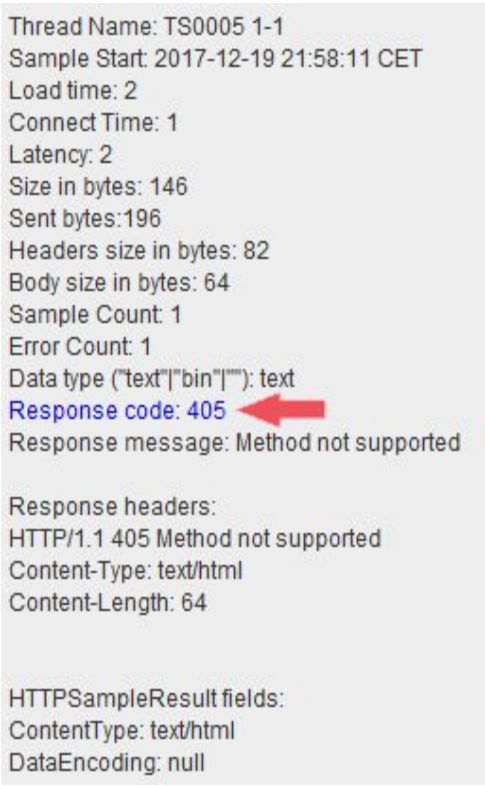


HTTP 1.1 Status 404

TS.4	HTTP 1.1 Status 404
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Expected result	Server responds with status code 404
Result	Server response with status code 404 - Not Found.
Status	*passed*

HTTP 1.1 Status 405

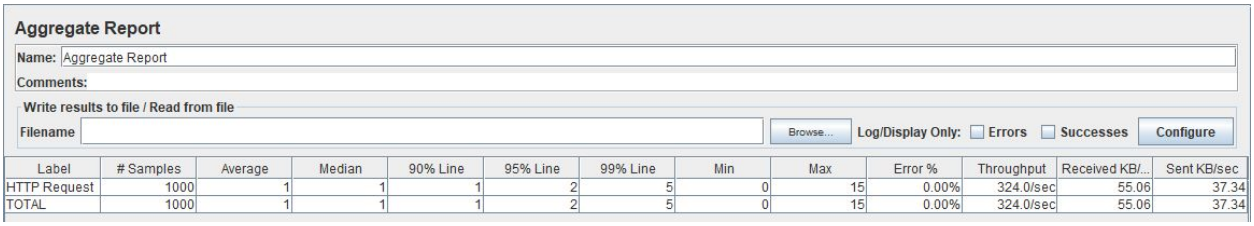
TS.5	HTTP 1.1 Status 405
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Expected result	Server responds with status code 405
Result	Server response with status code 405 - Method Not Supported.
Status	*passed*



Load Tests (JMeter)

High Load (1 000 users)

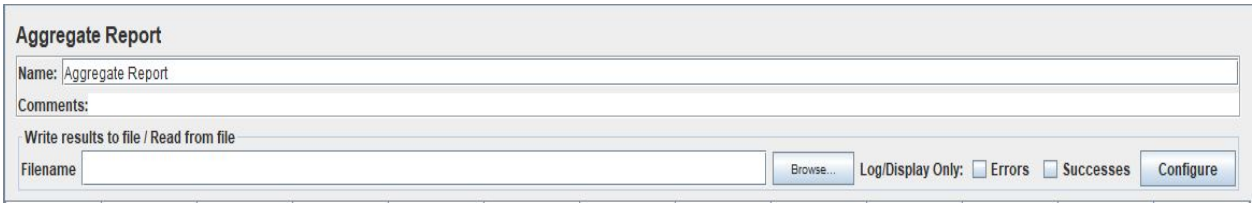
TL.1	High Load (1 000 users)
Requirement	Req 1. The web server should be responsive under high load
Use Case	--
Expected result	The Server is still responsive
Result	The Server is still responsive
Status	*passed*



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	1000	1	1	1	2	5	0	15	0.00%	324.0/sec	55.06	37.34
TOTAL	1000	1	1	1	2	5	0	15	0.00%	324.0/sec	55.06	37.34

High Load (50 000 users)

TL.2	High Load (50 000 users)
Requirement	Req 1. The web server should be responsive under high load
Use Case	--
Expected result	The Server is still responsive
Result	The Server is still responsive
Status	*passed*



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	50000	5	5	11	13	19	0	141	0.00%	9518.4/sec	1617.38	1096.84
TOTAL	50000	5	5	11	13	19	0	141	0.00%	9518.4/sec	1617.38	1096.84

Other

Operating System

TOS.1.1	Test Operating System
Requirement	Req 3. The web server must work on Linux, Mac, Windows*
Use Case	--
Expected result	The server should work in all the operative systems
Result	The testing department has no computers with Linux nor Mac available Test passed for Windows 10.
Status	*failed*

GPL-2.0

TGPL2.0	Test GPL-2.0
Requirement	Req 4. The source code should be released under GPL-2.0.
Use Case	--
Expected result	The server is under GPL license.
Result	My Web Server has a MIT license where the permission is granted free of charge to any person with a copy of the software without any restriction, distribution and or selling copies included with the condition of adding a reference to this copyright notice. This licence does not obligate to keep a project in open source and can be encrypted.
Status	*failed*

Browser

TB.1.1.	Test browser
Requirement	--
Use Case	--
Expected result	“It works” image is shown on the web page in all the browsers.
Result	“It works” image is shown on the web page in all the browsers.
Status	*passed*

Performance

TP.1.	Performance
Requirement	--
Use Case	--
Expected result	All tests succeed.
Result	No test succeeded. (*Detailed report can be offered upon request.)*
Status	*failed*

Acceptance

TA.1	Acceptance
Requirement	--
Use Case	--
Expected result	All tests succeed.
Result	Most of the tests succeed.
Status	*passed*

Evaluation

Upon evaluating the product by executing several different types of testing, the testing team has concluded that the software “My web server” does not fulfill all the necessary requirements provided by the SDC. The few flaws encountered could be too costly for the client and we suggest to avoid further development unless the budget allows it.

First of all the software does not follow the requirement regarding the licensing of the product under GPL-2.0. “My Web Server” having a MIT license where the permission is granted free of charge to any person with a copy of the software without any restriction.

Since the GPL is a copyright license, the copyright holders of the software are the ones who have the power to enforce the GPL. Furthermore, because the web server is an abandonware, the original developers unlikely to be found and negotiated with, we recommend that SDC reconsider their choice of software switching instead to WSO2 IoT Server or Kaa.

Moreover, the software does not provide, as recommended, an access log viewable from a text editor. Lastly, the limited budget provided by the company did not allow the testing team to test the “My web server” software on different operating systems and is unable to clearly state whether this requirement completely passes.