

# TEST REPORT

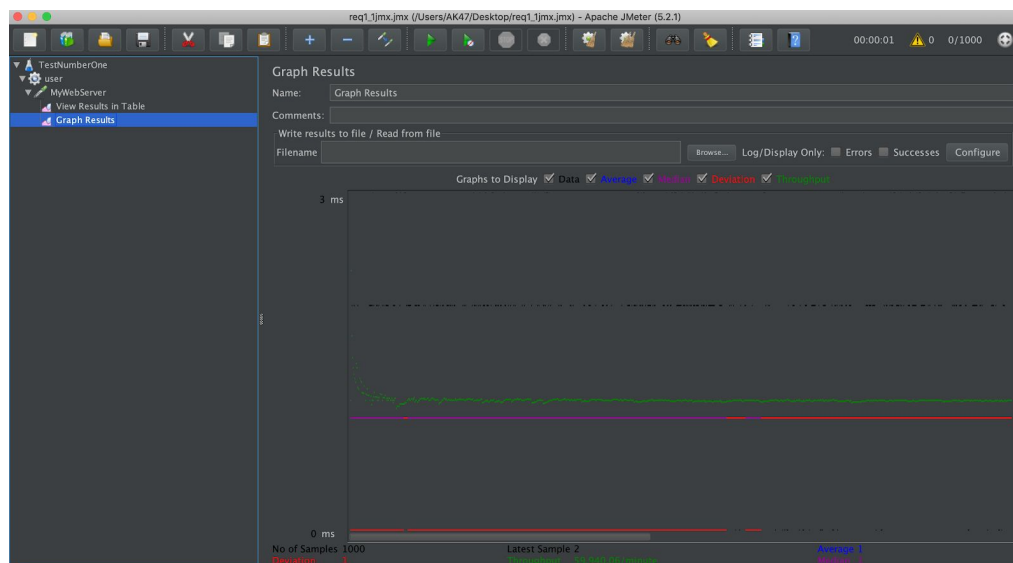
# Table of Contents

<b>I/Traceability Matrix</b>	<b>3</b>
<b>II/What has been tested</b>	<b>4</b>
TC 1.1.1 - The web server should be responsive under high load	4
TC 1.1.2 - The web server should be responsive under high load	5
TC 1.2.1 - The web server must follow minimum requirements for HTTP 1.1	6
TC 1.2.2 - The web server must follow minimum requirements for HTTP 1.1	7
TC 1.3.1 - The web server must work on Windows 10.	8
TC 1.3.2 - The web server must work on Mac OS Catalina.	8
TC 1.4 - The source code should be released under GPL-2.0.	9
TC 1.5 - The access log should be viewable from a text editor.	9
TC 1.6.1 - Start Server - The web server should start on the specific port	9
TC 1.6.2 - The web server should write into the access log file	10
TC 1.6.3 -The web server should not start due to taken socket	10
TC 1.6.4 - The web server should not start due to restrictions on the shared resource container	11
TC 1.6.5 - Server Start - Access log could not be written to	11
TC 1.7 - The web server should stop when asked to stop	12
TC 1.8.1 - The web server should share the requested resource statues 200 and write in the log	12
TC 1.8.2 - The web server should give 404 Not Found	13
TC 1.8.3 - The web server should give 403 Forbidden	13
TC 1.8.4 - The web server should give 400 Bad requests	15
TC 1.8.5 - The web server should encounter an error when processing a request	15
TCJU 2.1 - Integration	16
TCJU 2.2 - Response	16
TCJU 2.3 - View	16
TCJU 2.4 - All the JUnit tests suite	17
TA - Acceptance Testing	17
<b>II/What has not been tested</b>	<b>18</b>

## I/Traceability Matrix

### Traceability Matrix

[illegible]

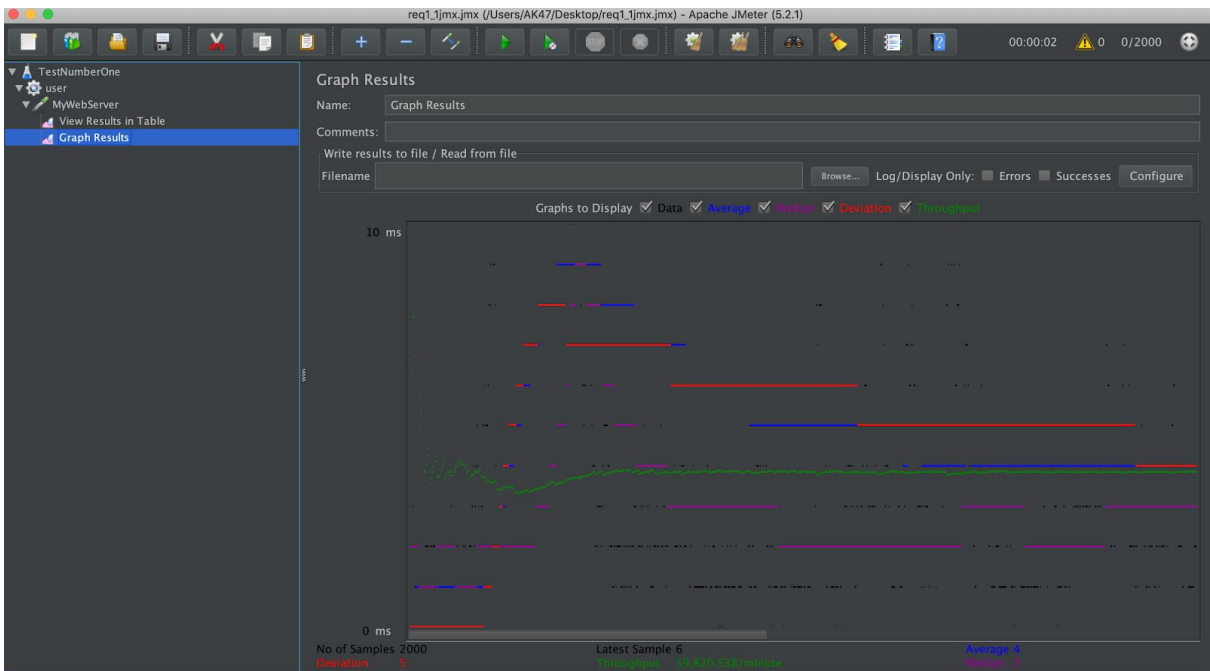
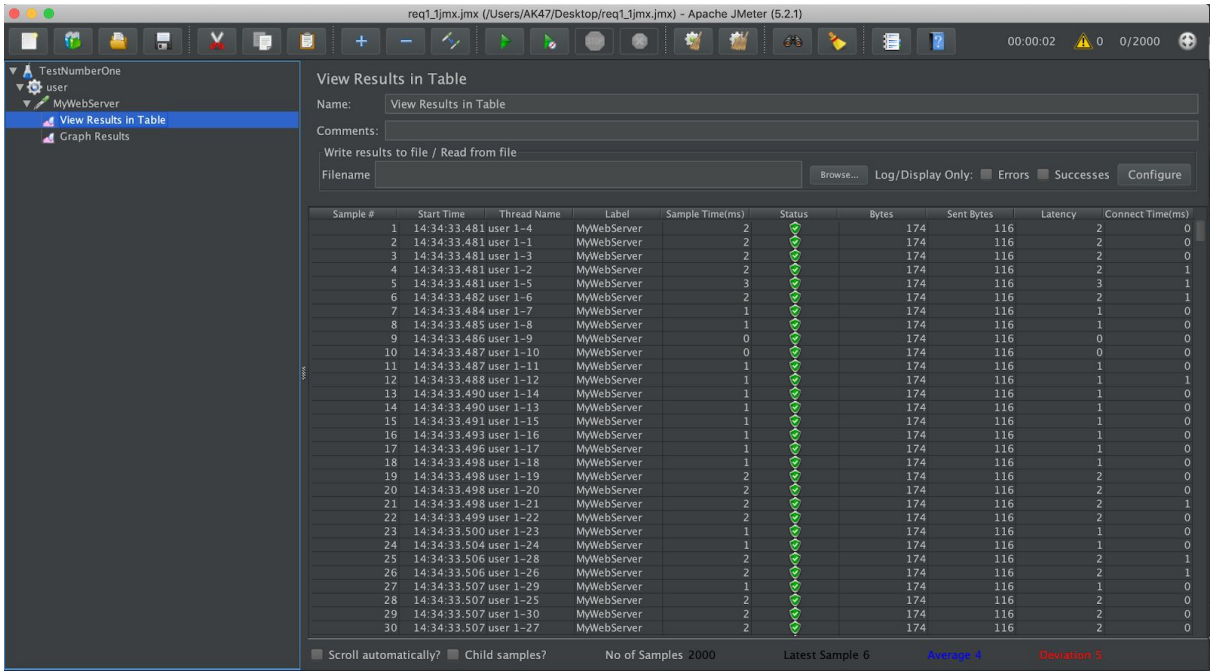


# TC 1.1.2 - The web server should be responsive under high load

TestID: REQ1.2

Test Result: **PASSED**

Proof:



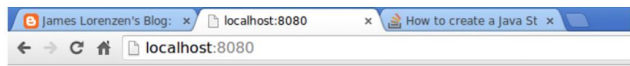
TC 1.2.1 - The web server must follow minimum requirements for HTTP 1.1

TestID: REQ2.1

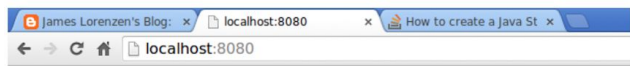
Test Result: PASSED

Proof:

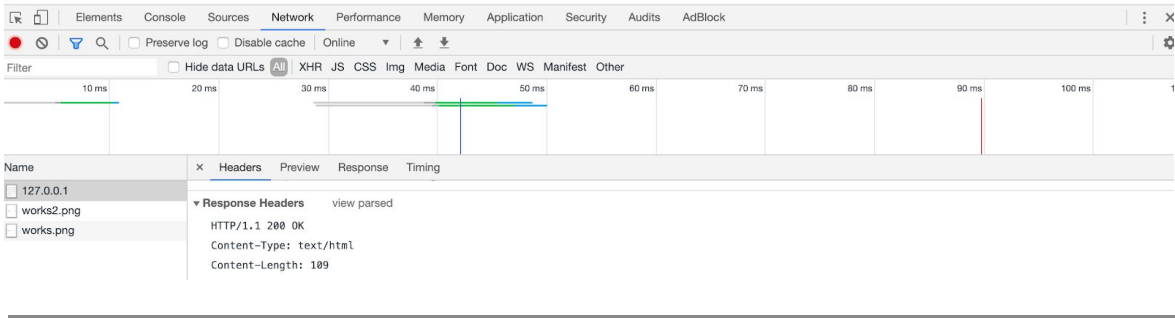
It works



It works



It works



## TC 1.2.2 - The web server must follow minimum requirements for HTTP 1.1

TestID: REQ2.2

Test Result: **PASSED**

Proof:

```
* Trying ::1...
* TCP_NODELAY set
* Connected to localhost (::1) port 9000 (#0)
> GET / HTTP/1.1
> Host: localhost:9000
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Type: text/html
< Content-Length: 109
<
<html>
<body>
<h1>It works</h1>


</body>
* Connection #0 to host localhost left intact
```

---

## TC 1.3.1 - The web server must work on Windows 10.

TestID: REQ3.1

Test Result: **PASSED**

Proof:

Windows edition

Windows 10 Home

© 2019 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz

Installed memory (RAM): 8.00 GB (7.89 GB usable)

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this device. [Support Information](#)

HP

```
HTTP Server object constructed
HTTP Server started
Accept
Accept
ClientThread started nr: 2
ClientThread started nr: 1
ClientThread 1 served file : index.html
ClientThread 1 served file : works.png
ClientThread 2 served file : works2.png
ClientThread stopped nr: 1
ClientThread stopped nr: 2
```

It works

James Lorenzen's Blog: x localhost:8080 x How to create a Java St x

localhost:8080

It works

James Lorenzen's Blog: x localhost:8080 x How to create a Java St x

localhost:8080

It works

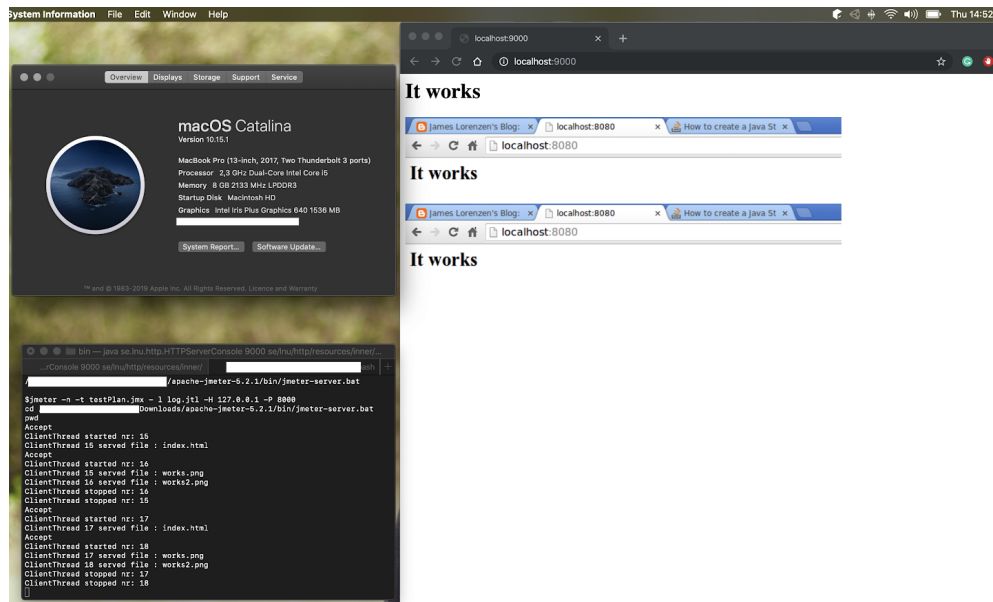
---

## TC 1.3.2 - The web server must work on Mac OS Catalina.

TestID: REQ3.2

Test Result: **PASSED**

Proof:



---

## TC 1.4 - The source code should be released under GPL-2.0.

TestID: REQ4.1

Test Result: **FAIL**

**Explanation:** MyWebServer has an MIT Licence because the licence is not under the obligation to stay open source and could be privatized. Due to this licence, it allows a person to redistribute and or selling copies included with the condition of adding a reference to this copyright notice.

---

## TC 1.5 - The access log should be viewable from a text editor.

TestID: REQ5.1

Test Result: **FAIL**

**Explanation:** There is no access log in this system.



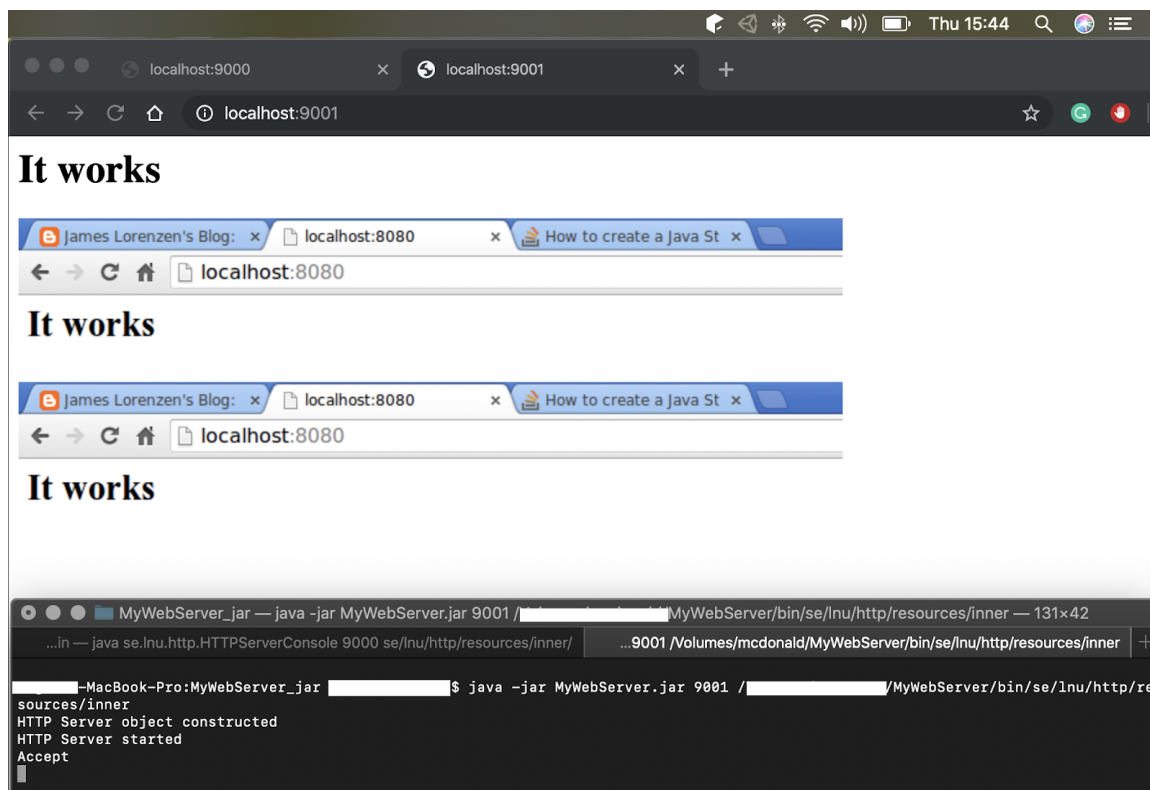
---

## TC 1.6.1 - Start Server - The web server should start on the specific port

TestID: REQ6.1

Test Result: **PASSED**

Proof:



---

## TC 1.6.2 - The web server should write into the access log file

TestID: REQ6.2

Test Result: **FAIL**

Explanation:

The server does start with the specific port, but it does not write into the access log file as it is missing.

---

### TC 1.6.3 -The web server should not start due to taken socket

**TestID:** REQ6.3

**Test Result:** **FAIL**

**Explanation:** Although it gives us Port is taken, it does not give us what the requirement tells which is: “Socket XX was taken”.

**Proof**

```
MacBook-Pro:MyWebServer_jar [redacted] $ java -jar MyWebServer.jar 9000 /var/www/MyWebServer/bin/se.lnu/http/re
sources/inner
HTTP Server object constructed
Port is taken
MacBook-Pro:MyWebServer_jar [redacted] $
```

---

### TC 1.6.4 - The web server should not start due to restrictions on the shared resource container

**TestID:** REQ6.4

**Test Result:** **FAIL**

**Explanation:**

The web server crashes with a Thread exception. Therefore it is a fail because it makes the whole server crash and does not show any appropriate error message. And does not give us the error message asked in the requirement: “No access to folder XX”(XX is the shared resource container provided, Example “\var\www”)

**Proof:**

```
Exception in thread "main" se.lnu.http.exceptions.NotADirectoryException: se.lnu.http.exceptions.NotADirectoryException: directory
does not exists
    at se.lnu.http.view.ConsoleView.getDirectory(ConsoleView.java:77)
    at se.lnu.http.HTTPServerConsole.runConsole(HTTPServerConsole.java:36)
    at se.lnu.http.HTTPServerConsole.main(HTTPServerConsole.java:27)
Caused by: se.lnu.http.exceptions.NotADirectoryException: directory does not exists
    at se.lnu.http.view.ConsoleView.getDirectory(ConsoleView.java:70)
    ... 2 more
```

---

## TC 1.6.5 - Server Start - Access log could not be written to

TestID: REQ6.5

Test Result: **FAIL**

### Explanation:

The access log does not exist and no error message appears in the console.

### Proof:

```
MacBook-Pro:MyWebServer_jar ~$ java -jar MyWebServer.jar 9001 /Volumes/mcdonald/MyWebServer/bin/se/lnu/http/re
sources/inner
HTTP Server object constructed
HTTP Server started
Accept
```

---

## TC 1.7 - The web server should stop when asked to stop

TestID: REQ7.1

Test Result: **PASSED**

### Proof:

```
HTTP Server object constructed
HTTP Server started
Accept
Accept
ClientThread started nr: 1
ClientThread stopped nr: 1
Accept
ClientThread started nr: 2
ClientThread 2 served file : index.html
Accept
ClientThread started nr: 3
ClientThread 2 served file : works.png
ClientThread 3 served file : works2.png
ClientThread stopped nr: 2
ClientThread stopped nr: 3
stop
HTTP Server Accept thread stopped
HTTP Server stopped
```

---

## TC 1.8.1 - The web server should share the requested resource statues 200 and write in the log

TestID: REQ8.1

Test Result: **FAIL**

### Explanation:

The web server output content in the browser and the status 200 is shown. However, the access log is never accessed therefore the webserver cannot write a success message in there.

---

## TC 1.8.2 - The web server should give 404 Not Found

TestID: REQ8.2

Test Result: **PASSED**

### Proof:



**404 Not found**

A screenshot of the 'Réseau' (Network) tab in a browser's developer tools. It shows two network requests, both with a status of 404. The first request is for the 'document' (HTML) and the second is for 'favicon.ico' (image).

État	Méthode	Domaine	Fichier	Source	Type	Transfert
404	GET	localhost:9001	/	document	html	119 o
404	GET	localhost:9001	favicon.ico	img	html	mis en cache

### TC 1.8.3 - The web server should give 403 Forbidden

TestID: REQ8.3

Test Result: **PASSED**

Proof:

The screenshot displays two panels from the Apache JMeter GUI. The top panel, titled 'Request Body' and 'Request Headers', shows a list of request components: a GET request to 'http://localhost:9001/../../secret.html', a 'GET data:' field, and a '[no cookies]' note. The bottom panel, titled 'Text', 'Sampler result', 'Request', and 'Response data', shows the 'Sampler result' for an 'HTTP Request GET'. It lists various metrics: Thread Name (Thread Group 1-1), Sample Start (2019-12-12 21:17:30 CET), Load time (1), Connect Time (0), Latency (1), Size in bytes (119), Sent bytes (130), Headers size in bytes (71), Body size in bytes (48), Sample Count (1), and Error Count (1). The 'Data type' is listed as 'text'. The 'Response code' is highlighted in blue as '403', and the 'Response message' is 'Forbidden'. At the bottom, the 'HTTPSampleResult fields' are shown: 'ContentType: text/html' and 'DataEncoding: null'.

```
1 GET http://localhost:9001/../../secret.html
2
3 GET data:
4
5
6 [no cookies]
7
```

Text  
Sampler result Request Response data

HTTP Request GET

Thread Name:Thread Group 1-1  
Sample Start:2019-12-12 21:17:30 CET  
Load time:1  
Connect Time:0  
Latency:1  
Size in bytes:119  
Sent bytes:130  
Headers size in bytes:71  
Body size in bytes:48  
Sample Count:1  
Error Count:1  
Data type ("text"|"bin"|""):text  
Response code:403  
Response message:Forbidden

HTTPSampleResult fields:  
ContentType: text/html  
DataEncoding: null

---

## TC 1.8.4 - The web server should give 400 Bad request

TestID: REQ8.4

Test Result: **PASSED**

Proof:

```
</html> [redacted]:~curl -d "h1=Hey" localhost:9000/  
<html><body><h1>400 Bad request</h1></body></html>MacBo
```

---

## TC 1.8.5 - The web server should encounter an error when processing a request

TestID: REQ8.5

Test Result: **PASSED**

Output: Response code 405.

Proof:

```
1 POST http://localhost:9001/indet.html  
2  
3 POST data:  
4  
5  
6 [no cookies]  
7
```

The screenshot shows the 'Sampler result' tab of an HTTP Request POST sampler in JMeter. The left pane shows the sampler name 'HTTP Request POST' with a red error icon. The right pane displays the following details:

- Thread Name: Thread Group 1-1
- Sample Start: 2019-12-12 21:16:37 CET
- Load time: 4
- Connect Time: 3
- Latency: 4
- Size in bytes: 146
- Sent bytes: 210
- Headers size in bytes: 82
- Body size in bytes: 64
- Sample Count: 1
- Error Count: 1
- Data type ("text"|"bin"|""): text
- Response code: 405**
- Response message: Method not supported

At the bottom, the 'HTTPSampleResult fields' are listed:

- ContentType: text/html
- DataEncoding: null

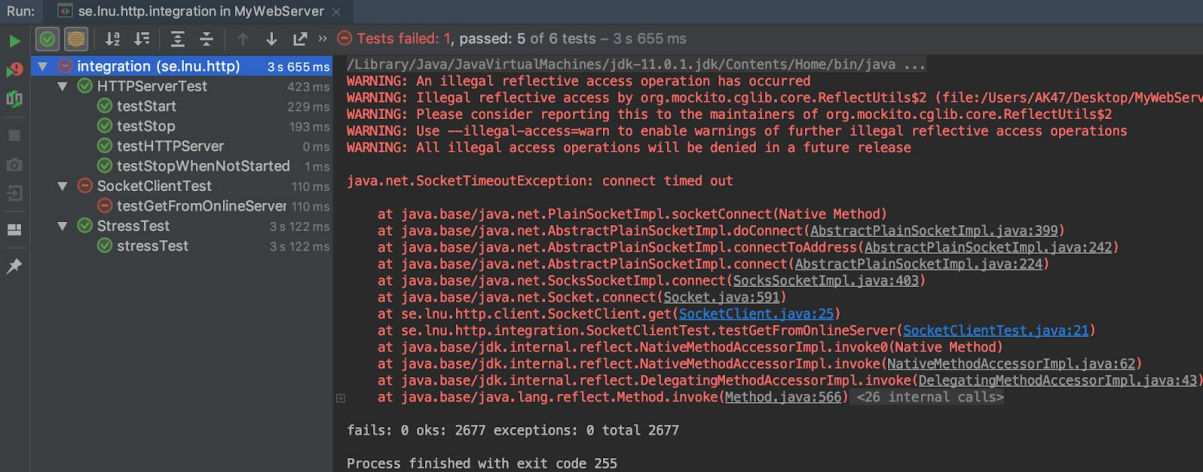
---

## TCJU 2.1 - Integration

TestID: REQ9.1

Test Result: **FAIL**

Output:



```
Run: se.lnu.http.integration in MyWebServer x
Tests failed: 1, passed: 5 of 6 tests - 3 s 655 ms

integration (se.lnu.http) 3 s 655 ms
  HTTPServerTest 423 ms
    testStart 229 ms
    testStop 193 ms
    testHTTPServer 0 ms
    testStopWhenNotStarted 1 ms
  SocketClientTest 110 ms
    testGetFromOnlineServer 110 ms
  StressTest 3 s 122 ms
    stressTest 3 s 122 ms

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.mockito.cglib.core.ReflectUtils$2 (file:/Users/AK47/Desktop/MyWebServ
WARNING: Please consider reporting this to the maintainers of org.mockito.cglib.core.ReflectUtils$2
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release

java.net.SocketTimeoutException: connect timed out

    at java.base/java.net.PlainSocketImpl.socketConnect(Native Method)
    at java.base/java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:399)
    at java.base/java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:242)
    at java.base/java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:224)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:403)
    at java.base/java.net.Socket.connect(Socket.java:591)
    at se.lnu.http.client.SocketClient.get(SocketClient.java:25)
    at se.lnu.http.integration.SocketClientTest.testGetFromOnlineServer(SocketClientTest.java:21)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:566) <26 internal calls>

fails: 0 oks: 2677 exceptions: 0 total 2677

Process finished with exit code 255
```

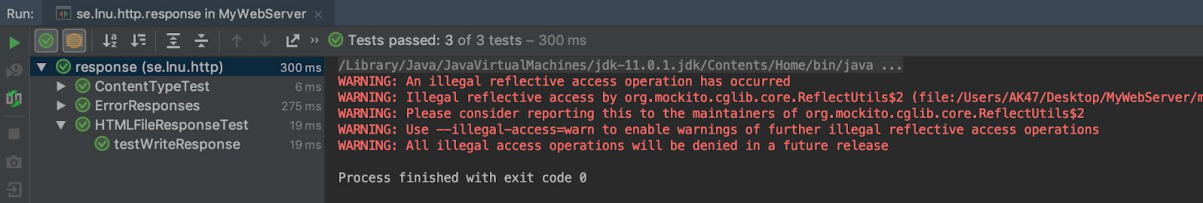
---

## TCJU 2.2 - Response

TestID:REQ9.2

Test Result: **PASSED**

Output:



```
Run: se.lnu.http.response in MyWebServer x
Tests passed: 3 of 3 tests - 300 ms

response (se.lnu.http) 300 ms
  ContentTypeTest 6 ms
  ErrorResponses 275 ms
  HTMLFileResponseTest 19 ms
    testWriteResponse 19 ms

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.mockito.cglib.core.ReflectUtils$2 (file:/Users/AK47/Desktop/MyWebServer/m
WARNING: Please consider reporting this to the maintainers of org.mockito.cglib.core.ReflectUtils$2
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release

Process finished with exit code 0
```

**Explanation:** Most of the prioritization tests pass therefore, the test leader decided to consider it as a passed



---

## II/What has not been tested

Requirement 3 was not tested on all possible Operating Systems, which has been decided by the test leader. The reason behind it is because many operating systems are not important for what is related to the Internet of Things (IoT). Moreover, it requires more sophisticated hardware and software to make sure that the testing team can test at full potential the web server.