TESTING STRATEGY

The following methods were used to test these requests:

1) netcat
2) Python script
3) Browser

Valid Request Tests (200 Response)

Test Case Name	Description	Request Message Sent
Initial line valid	We send a request that contains valid initial line, i.e. GET <url> HTTP/1.1. The server is expected to service this request and respond with the correct file contents.	GET /index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n
Host header present and correctly set	We send a request that contains the Host header properly set, with the right format (Host: <hostname>). The server is supposed to treat this as a valid request.	GET /index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n
Relative paths and subdirectories	The client sends a request to fetch a file that is present within a subdirectory of the server's document root. Any subdirectory of the root must be supported.	GET /subdir1/index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n
Valid URL and correct Content-Length of response body	When the client requests a valid file, the Content-Length header must be set appropriately in the response message.	GET /index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n
Append index.html to URL ending in /	If the requested URL ends in a /, we append the string "index.html" to it.	GET / HTTP/1.1\r\nHost: Ha1\r\n\r\n
Relative URL paths containing ..	If the client requests a relative path that contains .., it is resolved to absolute path as long as it is under the document root.	GET /subdir1/../index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n

Output:

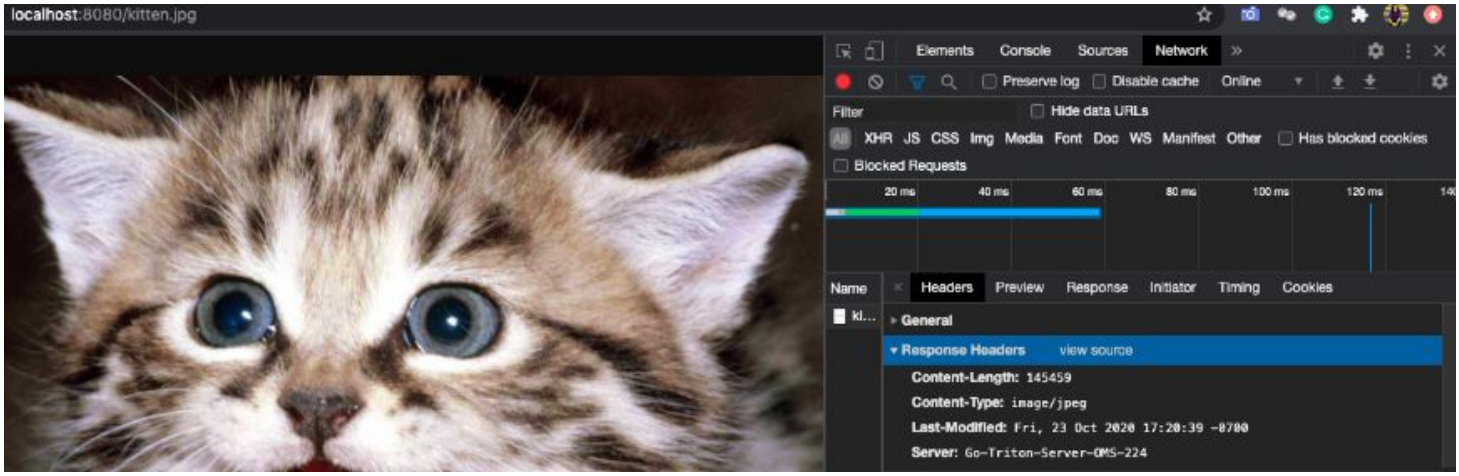
```
(base) Akshayas-Air:CSE 224 akshayaraju$ printf "GET /index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n" > 200.txt
(base) Akshayas-Air:CSE 224 akshayaraju$ cat 200.txt | nc localhost 8080
HTTP/1.1 200 OK
Content-Length:307
Content-Type:text/html
Server:Go-Triton-Server-OMS-224
Last-Modified:Fri, 23 Oct 2020 17:20:39 -0700

<html>
<head>
<title>Basic index file</title>
</head>
<body>
<h1>This is a basic index file</h1>
You can use this for testing.
<p>
<ul>
<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>
<li><a href=kitten.jpg alt="Kitten">Kitten photo</a>
<li><a href=subdir1/>A subdirectory</a>
</ul>
</body>
</html>
(base) Akshayas-Air:CSE 224 akshayaraju$
```

Note: connection closed after 5 seconds.

Test Case Name	Description	Request Message Sent
Support the MIME types specified in mime.types	The client is allowed to request a large number of file types, as listed in the mime.types file. The server sets the corresponding MIME type in the form of Content-Type header in the response.	GET /kitten.jpg HTTP/1.1\r\nHost: Ha1\r\n\r\n

Output:



Note: The command line tests produced a large amount of binary data in the response so we have included the browser screenshot.

Test Case Name	Description	Request Message Sent
Support with default MIME type when requested type is not specified in mime.types	If the requested file type is not in the supported MIME types list, the default content type is set to application/octet-stream.	GET /index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n

Output:

```
(base) Akshayas-Air:CSE 224 akshayaraju$ printf "GET /subdir1/subdir11/maoyo.giaogiao HTTP/1.1\r\nHost: Ha1\r\n\r\n" > 200.txt
(base) Akshayas-Air:CSE 224 akshayaraju$ cat 200.txt | nc localhost 8080
HTTP/1.1 200 OK
Server:Go-Triton-Server-OMS-224
Last-Modified:Fri, 23 Oct 2020 17:20:39 -0700
Content-Length:4
Content-Type:application/octet-stream
```

Test Case Name	Description	Request Message Sent
Connection: close header	When the Connection: close header is set, the server responds with a message that also has Connection: close set and the server closes the connection after processing this request.	GET / HTTP/1.1\r\nHost: Ha1\r\nConnection: close\r\n\r\n

Output:

```
(base) Akshayas-Air:CSE 224 akshayaraju$ printf "GET / HTTP/1.1\r\nHost: Ha1\r\nConnection: close\r\n\r\n" > 200.txt
(base) Akshayas-Air:CSE 224 akshayaraju$ cat 200.txt | nc localhost 8080
HTTP/1.1 200 OK
Last-Modified:Fri, 23 Oct 2020 17:20:39 -0700
Content-Length:307
Content-Type:text/html
Connection:close
Server:Go-Triton-Server-OMS-224

<html>
<head>
<title>Basic index file</title>
</head>
<body>
<h1>This is a basic index file</h1>
You can use this for testing.
<p>
<ul>
<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>
<li><a href=kitten.jpg alt="Kitten">Kitten photo</a>
<li><a href=subdir1/>A subdirectory</a>
</ul>
</body>
</html>
(base) Akshayas-Air:CSE 224 akshayaraju$
```

Note: the connection was closed immediately after receiving the response text.

Malformed requests (400 response)

Test Case Name	Description	Request Message Sent
Initial line invalid - missing GET	If the request message is missing the GET keyword in the first line, it is treated as a Bad Request and 400 response is generated and closes the connection.	GETT / HTTP/1.1\r\nHost: Ha1\r\n\r\n
Initial line invalid - missing /	If the requested URL does not begin with / in the first line, it is treated as a Bad Request and 400 response is generated and closes the connection.	GET index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n
Initial line invalid - missing HTTP/1.1	If the request message is missing the HTTP/1.1 keyword in the first line, it is treated as a Bad Request and 400 response is generated and closes the connection.	GETT / HTTP/1.2\r\nHost: Ha1\r\n\r\n
Invalid headers	If the request message contains invalid headers, such as missing colon(:), it is treated as a Bad Request and 400 response is generated and closes the connection.	GETT / HTTP/1.1\r\nHost Ha1\r\n\r\n
Missing Host header	If the request message does not contain a valid Host header, the server must generate a 400 Bad Request error and closes the connection.	GETT / HTTP/1.1\r\n\r\n

Output:

Note: this connection was closed immediately after receiving the 400 response.

FileNotFound Requests (400 response)

Test Case Name	Description	Request Message Sent
URL not resolving to an existing file	If the requested URL points to a file that does not exist, the server returns a 404 FileNotFound error but keeps the connection open.	GET /index1.html HTTP/1.1\r\nHost: Ha1\r\n\r\n
URL Escaping document root	If the client tries to request a file that is above the document root, the server again responds with 404 FileNotFound but keeps the connection open.	GET /../index.html HTTP/1.1\r\nHost: Ha1\r\n\r\n

Output:

```
(base) Akshayas-Air:CSE 224 akshayaraju$ printf "GET /index1.html HTTP/1.1\r\nHost: Ha1\r\n\r\n" > 404.txt
(base) Akshayas-Air:CSE 224 akshayaraju$ cat 404.txt | nc localhost 8080
HTTP/1.1 404 Not Found
Server:Go-Triton-Server-OMS-224
Content-Length:0

HTTP/1.1 400 Bad Request
Server:Go-Triton-Server-OMS-224
Content-Length:0
```

Note: the connection stays open for 5 seconds and then closes.

Timeout Requests

Test Case Name	Description	Request Message Sent
5 seconds timeout (after successful read)	Once a request is successfully processed, the server refreshes a 5 second timer and closes the connection after this. Until then, new requests are accepted.	nc localhost 8080
Partial request	If the server reaches the 5 second timeout and a partial request was read by then, the server responds with 400 Bad Request and closes the connection.	GET

Output:

```
(base) Akshayas-Air:CSE 224 akshayaraju$ printf "GET" > timeout.txt
(base) Akshayas-Air:CSE 224 akshayaraju$ cat timeout.txt | nc localhost 8080
HTTP/1.1 400 Bad Request
Server:Go-Triton-Server-OMS-224
Content-Length:0

(base) Akshayas-Air:CSE 224 akshayaraju$ nc localhost 8080
```

Note: the server waits for 5 seconds for a full request and responds with 400 Bad Request for a partial request.

Pipelining

This section was mostly tested using a Python script very similar to the example provided to us in the Testing Strategies document.

Test Case Name	Description	Request Message Sent
Pipelined requests - all valid	Client sends multiple requests in series, all of them being valid. The server processes all 3 and responds in the same sequence, and closes the connection after timeout/Connection:close header.	GET /index.html HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 1\r\n\r\nGET / HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 2\r\n\r\nGET /index.html HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 3\r\n\r\n"

Output:

```
[(base) Akshayas-Air:CSE 224 akshayaraju$ python testing_script.py
b'tent-Length:307\r\nContent-Type:text/html\r\n\r\n'
b'<html>\n<head>\n<title>Basic index file</title>\n</head>\n<body>\n<h1>This is a basic index file</h1>\nYou'
b' can use this for testing.\n<p>\n<ul>\n<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>\n<li><a h'
b'ref=kitten.jpg alt="Kitten">Kitten photo</a>\n<li><a href=subdir1/>A subdirectory</a>\n</ul>\n</body>\n<'
b'/html>\n'
b'HTTP/1.1 200 OK\r\nLast-Modified:Fri, 23 Oct 2020 17:20:39 -0700\r\nContent-Length:307\r\nContent-Type:tex'
b't/html\r\nServer:Go-Triton-Server-OMS-224\r\n\r\n'
b'<html>\n<head>\n<title>Basic index file</title>\n</head>\n<body>\n<h1>This is a basic index file</h1>\nYou'
b' can use this for testing.\n<p>\n<ul>\n<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>\n<li><a h'
b'ref=kitten.jpg alt="Kitten">Kitten photo</a>\n<li><a href=subdir1/>A subdirectory</a>\n</ul>\n</body>\n<'
b'/html>\n'
b'HTTP/1.1 200 OK\r\nLast-Modified:Fri, 23 Oct 2020 17:20:39 -0700\r\nContent-Length:307\r\nContent-Type:tex'
b't/html\r\nServer:Go-Triton-Server-OMS-224\r\n\r\n'
b'<html>\n<head>\n<title>Basic index file</title>\n</head>\n<body>\n<h1>This is a basic index file</h1>\nYou'
b' can use this for testing.\n<p>\n<ul>\n<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>\n<li><a h'
b'ref=kitten.jpg alt="Kitten">Kitten photo</a>\n<li><a href=subdir1/>A subdirectory</a>\n</ul>\n</body>\n<'
b'/html>\n'
```

Note: 3 pipelined requests in the same connection were processed one after the other and the connection stayed open after the third request for 5 seconds before timing out.

Test Case Name	Description	Request Message Sent
3 Pipelined requests in the order: valid, malformed and valid	When the first request is valid and the second one malformed, the server processes the first one successfully, returns a 400 Bad Request for the second one and closes the connection without reading the third request.	"GET /index.html HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 1\r\n\r\nGETT / HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 2\r\n\r\nGET /index.html HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 3\r\n\r\n"

Output:

```
[(base) Akshayas-Air:CSE 224 akshayaraju$ python testing_script.py
b't/html\r\nServer:Go-Triton-Server-OMS-224\r\n\r\n'
b'<html>\n<head>\n<title>Basic index file</title>\n</head>\n<body>\n<h1>This is a basic index file</h1>\nYou'
b' can use this for testing.\n<p>\n<ul>\n<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>\n<li><a h'
b'ref=kitten.jpg alt="Kitten">Kitten photo</a>\n<li><a href=subdir1/>A subdirectory</a>\n</ul>\n</body>\n<'
b'/html>\n'
b'HTTP/1.1 400 Bad Request\r\nServer:Go-Triton-Server-OMS-224\r\nContent-Length:0\r\n\r\n'
b''
(base) Akshayas-Air:CSE 224 akshayaraju$
```

Note: The first request in the pipeline was processed, and since the second one was a partial request, the server timed out and responded with 400 Bad Request.

Test Case Name	Description	Request Message Sent
3 Pipelined requests with Connection: close header in the second one	The client sends two requests, the second one having Connection: close header set. The server processes both and closes the connection immediately after processing the second one.	GET /index.html HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 1\r\n\r\nGET / HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 2\r\n**Connection:close**\r\n\r\nGET /index.html HTTP/1.1\r\nHost: Ha1\r\nRequestNo: 3\r\n\r\n

Output:

```
(base) Akshayas-Air:CSE 224 akshayaraju$ python testing_script.py
b'tent-Length:307\r\nContent-Type:text/html\r\n\r\n'
b'<html>\n<head>\n<title>Basic index file</title>\n</head>\n<body>\n<h1>This is a basic index file</h1>\nYou'
b' can use this for testing.\n<p>\n<ul>\n<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>\n<li><a h'
b'ref=kitten.jpg alt="Kitten">Kitten photo</a>\n<li><a href=subdir1/>A subdirectory</a>\n</ul>\n</body>\n<'
b'/html>\n'
b'HTTP/1.1 200 OK\r\nServer:Go-Triton-Server-OMS-224\r\nLast-Modified:Fri, 23 Oct 2020 17:20:39 -0700\r\nCon'
b'tent-Length:307\r\nContent-Type:text/html\r\nConnection:close\r\n\r\n'
b'<html>\n<head>\n<title>Basic index file</title>\n</head>\n<body>\n<h1>This is a basic index file</h1>\nYou'
b' can use this for testing.\n<p>\n<ul>\n<li><a href=UCSD_Seal.png alt="UCSD Seal">UCSD seal</a>\n<li><a h'
b'ref=kitten.jpg alt="Kitten">Kitten photo</a>\n<li><a href=subdir1/>A subdirectory</a>\n</ul>\n</body>\n<'
b'/html>\n'
b''
(base) Akshayas-Air:CSE 224 akshayaraju$
```

Note: Even though we sent 3 pipelined requests, the third one was not processed because the second one had Connection: close header set.

Concurrency

For this component, we tested the performance by running the server on one teammate's machine and sending requests from the same machine, and also the other teammate's machine at the same time. The following test cases were tested:

Test Case Name	Description
2 Concurrent clients - both sending valid request	Each request is handled separately (but concurrently) and the responses are sent to the two different clients. Each connection is kept open unless timeout or Connection:close force closes it.
2 Concurrent clients - one sending valid request and other one sending malformed request	The valid request is processed and the appropriate response is sent, while the malformed request is responded to with a 400 error and that connection is closed.
2 Concurrent clients - one sending valid request, and the other sending request with URL that escapes doc root	Each concurrent request is processed separately and the appropriate response messages must be sent.

Each of these tests produced results that matched the expected behavior. This was easy to verify because we received the appropriate response messages on both clients.