

Name – Meghana Shivaprasad

Programming Assignment- Distributed systems

Description:

- We are trying to implement a simple webserver using Java. The webserver responds to client requests. The client requests are parsed according to the valid HTTP request format rules. We return the response to the user.
- First the client makes requests to the web server and the connection is accepted. The application is multithreaded, a new thread is spawn to handle the request, and the request is parsed. If there is no file given, then we load the index.html. (the html file is included in the directory which the java server loads when there is no file mentioned in the request)
- These are the types of files supported : jpeg/jpg, gif, ico, png and txt/html is supported
- There are response code which is returned when a client makes a request :
 - 200- the file is present and accessible (HTTP 200 OK message is returned)
 - 404- if file is not found.

Supported response Headers: Response code, Content-Type, Content-Length, Date

List of Files :

- Webserver.pdf
- Server.java
- Index_files (Folder containing images and supporting css/.js files of SCU homepage)
index.html – (HTML file containing the home page)
Makefile to build and run the webserver.
- Screenshots- output screenshots
- Forbidden.txt – Setting the permissions of a file or directory to chmod 000 essentially revokes all permissions .In a web server context, doing this to a file or directory can result in a "403 Forbidden" error when trying to access it via a web browser.

Instructions to run the program:

refer Makefile to run the program

Open the directory, if it is zipped or jar, unzip it and extract the contents . go the directory Webserver_Meghana

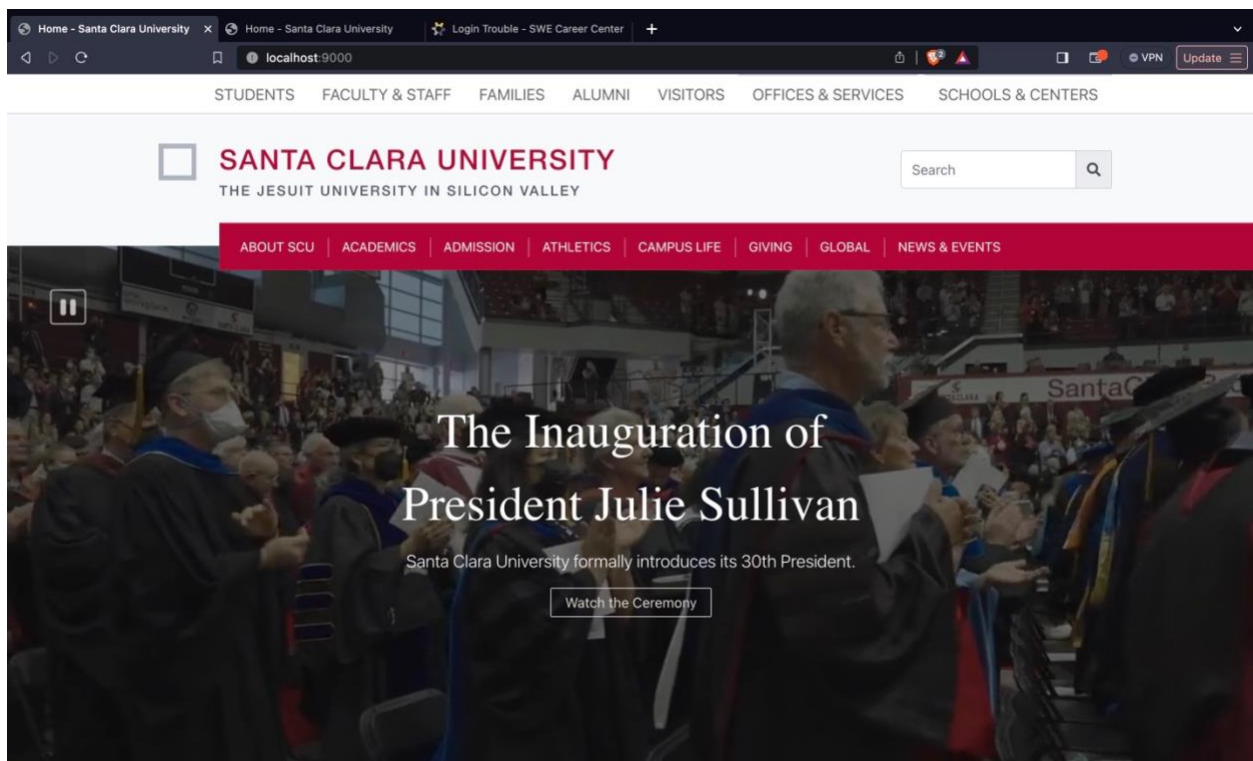
First we need to compile the code and then run the executable:
javac Server.java – to compile the code
java Server -document_root / -port 9000 – to run the program.
Go to the browser and run <http://localhost:9000>

Screenshots:

- Logs and Outputs:

1.Browser output:

Here client requests <http://localhost:9000> in brave browser with no specific file, so we just return the index.html file.



2.logs from the terminal

```
meghanashivaprasad@Meghanas-MacBook-Air Webserver_Meghana %  
meghanashivaprasad@Meghanas-MacBook-Air Webserver_Meghana %  
meghanashivaprasad@Meghanas-MacBook-Air Webserver_Meghana % java Server -document_root / -port 9000  
  
*****  
Server socket opened on port: 9000  
ServerSocket[addr=0.0.0.0/0.0.0.0,localport=9000]  
  
*****  
Server waiting for new client requests  
Thread created: Thread[#20,Thread-0,5,main]  
Starting thread..  
  
*****  
Server waiting for new client requests  
  
Request GET / HTTP/1.1  
Sending response headers:  
HTTP/1.0 200 OK  
Date: Wed, 25 Oct 2023, 04:04:01 PDT  
Content-length: 36198  
Content-type: text/html  
  
connection closed  
Socket[addr=/0:0:0:0:0:0:1,port=55218,localport=9000]  
Thread created: Thread[#21,Thread-1,5,main]  
Starting thread..  
  
*****  
Server waiting for new client requests  
  
Request GET /index_files/f.css HTTP/1.1  
Sending response headers:  
HTTP/1.0 200 OK  
Date: Wed, 25 Oct 2023, 04:04:01 PDT  
Content-length: 4926  
Content-type: text/plain  
  
connection closed  
Socket[addr=/0:0:0:0:0:0:1,port=55219,localport=9000]  
Thread created: Thread[#22,Thread-2,5,main]  
Starting thread..  
  
*****  
Server waiting for new client requests  
  
Request GET /index_files/Claire_Alford_20220921_JG_0153_pp.jpg HTTP/1.1
```

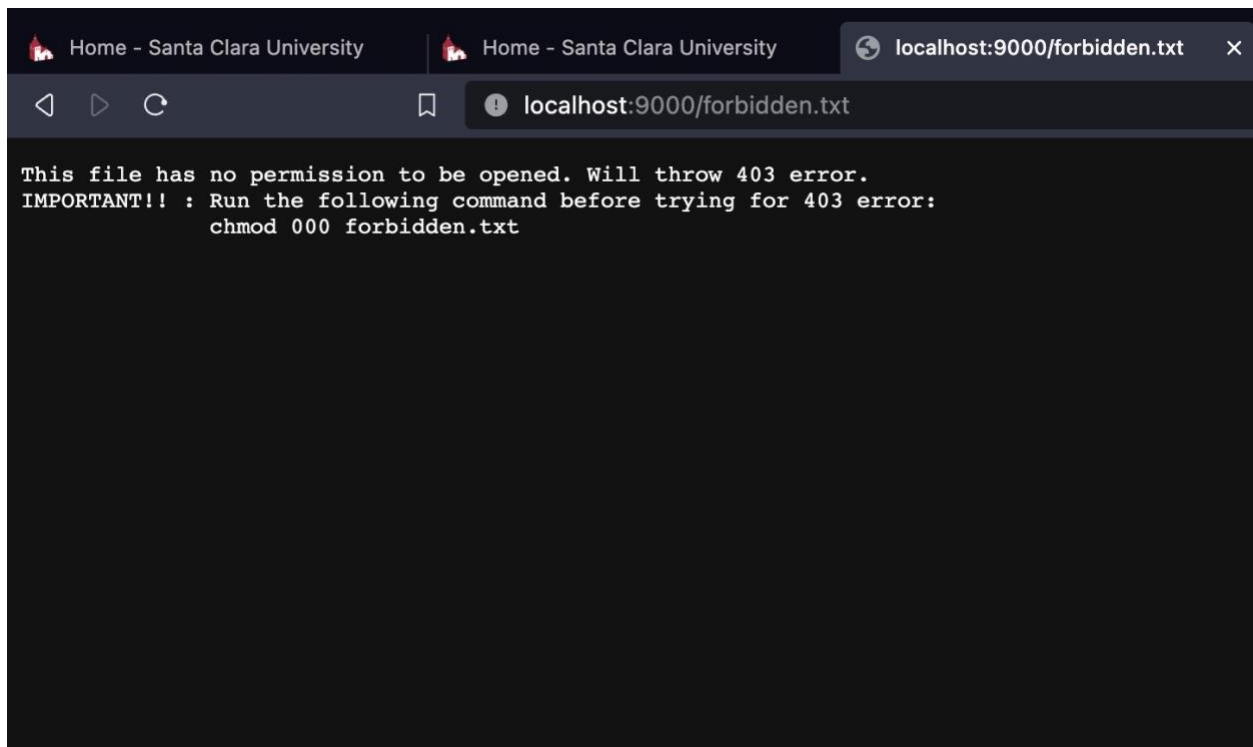
3. Response Headers

×	Headers	Preview	Response	Initiator	Timing
▼ General					
Request URL:		http://localhost:9000/			
Request Method:		GET			
Status Code:		● 200 OK			
Remote Address:		[::1]:9000			
Referrer Policy:		strict-origin-when-cross-origin			
▼ Response Headers		<input type="checkbox"/> Raw			
Content-Length:		36198			
Content-Type:		text/html			
Date:		Wed, 25 Oct 2023, 04:13:05 PDT			
▼ Request Headers		<input type="checkbox"/> Raw			
Accept:		text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8			
Accept-Encoding:		gzip, deflate, br			
Accept-Language:		en-US,en			
Cache-Control:		max-age=0			
Connection:		keep-alive			
Host:		localhost:9000			
Sec-Ch-Ua:		"Chromium";v="116", "Not)A;Brand";v="24", "Brave";v="116"			
Sec-Ch-Ua-Mobile:		?1			
Sec-Ch-Ua-Platform:		"Android"			
Sec-Fetch-Dest:		document			
Sec-Fetch-Mode:		navigate			
Sec-Fetch-Site:		none			
Sec-Fetch-User:		?1			
Sec-Gpc:		1			
Upgrade-Insecure-Requests:		1			
User-Agent:		Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Mobile Safari/537.36			

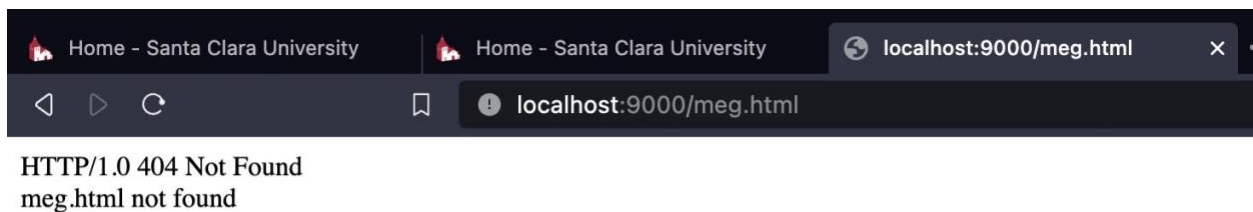
Status :

400, 403, 404, 200

403: the file is not found and the response code is 403



404: When the file does not exist, 404 response code.



200: When the request has been successful , 200 status code is sent back to the client

The screenshot shows the Chrome DevTools Network tab with the 'Headers' sub-tab selected. The response is a 200 status code, indicating a successful request. The 'General' section shows the request URL, method (GET), and status code (200 from disk cache). The 'Response Headers' section lists various headers including Accept-Ranges, Alt-Svc, Content-Length, Content-Security-Policy, Content-Type, Date, Etag, Last-Modified, Server, Vary, and Via. The 'Request Headers' section shows the Referer, Sec-Ch-Ua, Sec-Ch-Ua-Mobile, Sec-Ch-Ua-Platform, and User-Agent.

Header	Value
Request URL:	https://www.scu.edu/media/institutional-pages/giving/images/campaign-banner.png
Request Method:	GET
Status Code:	200 (from disk cache)
Remote Address:	34.107.151.86:443
Referrer Policy:	strict-origin-when-cross-origin
Accept-Ranges:	bytes
Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Content-Length:	26150
Content-Security-Policy:	frame-ancestors 'self' t4.scu.edu cms.scu.edu cms01.scu.edu thetrustproject.org media.scu.edu ecampus.scu.edu hrdev.scu.edu hrusr.scu.edu t4dev.scu.edu 166.78.46.137 campaign.scu.edu vanillasoft.net 129.210.247.132
Content-Type:	image/png
Date:	Wed, 25 Oct 2023 23:08:32 GMT
Etag:	"6626-5b331091b9a2f"
Last-Modified:	Tue, 03 Nov 2020 10:07:51 GMT
Server:	Apache
Vary:	X-Forwarded-Proto
Via:	1.1 google
Referer:	https://www.scu.edu/
Sec-Ch-Ua:	"Chromium";v="116", "Not)A;Brand";v="24", "Brave";v="116"
Sec-Ch-Ua-Mobile:	?1
Sec-Ch-Ua-Platform:	"Android"
User-Agent:	Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Mobile Safari/537.36

400: Unsupported HTTP method is requested then 400 is returned

3. Reference:

<https://www.geeksforgeeks.org/socket-programming-in-java/>
<https://www.geeksforgeeks.org/java-threads/>