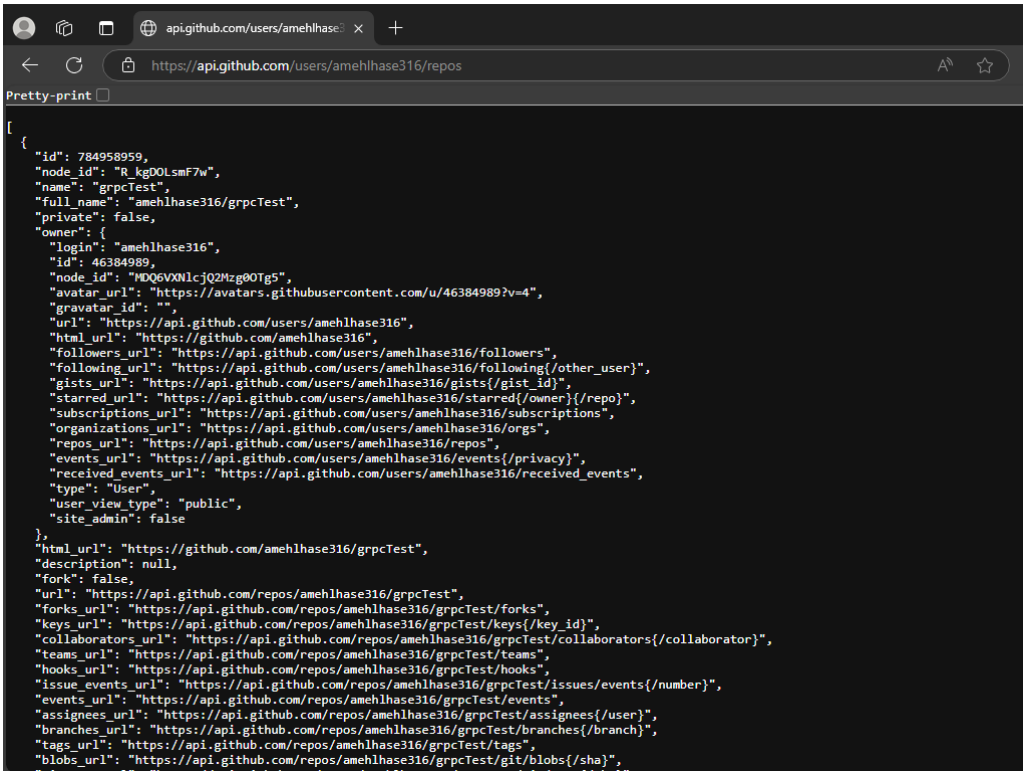
Matthew Lowber
SER321
Assignment 2

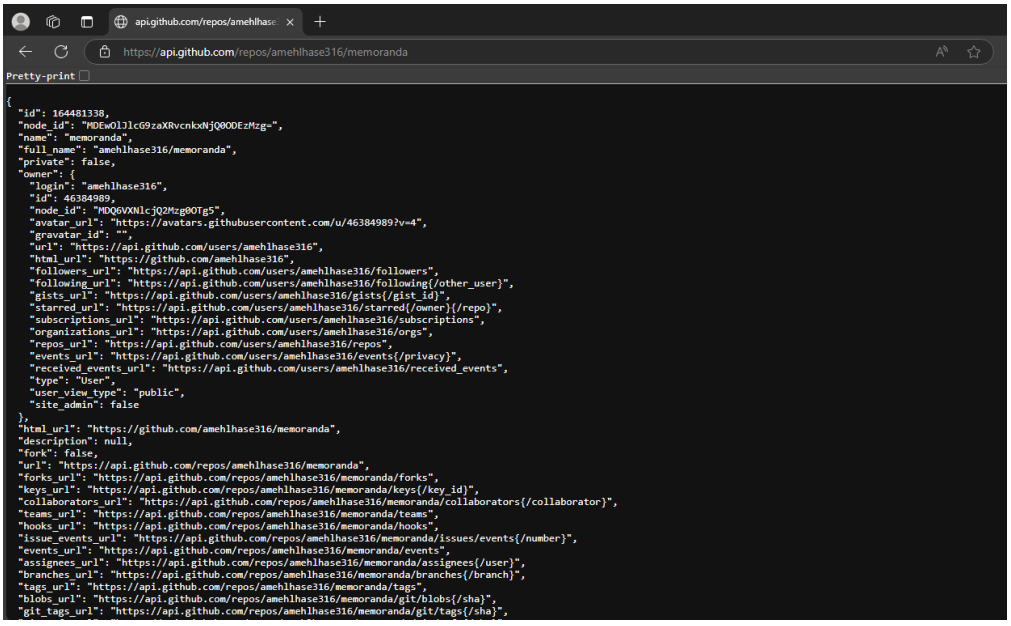
Task 1

https://api.github.com/users/amehlhase316/repos



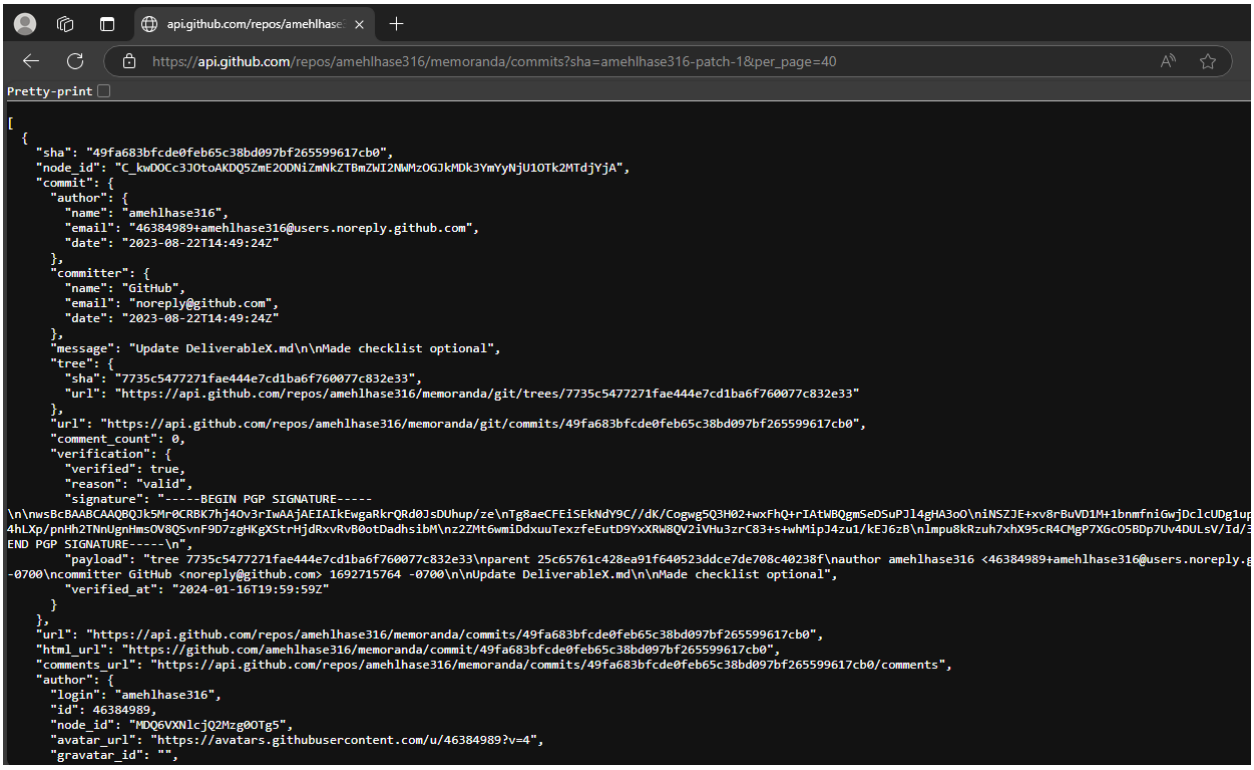

```
[
  {
    "id": 784958959,
    "node_id": "R_kgDOLsmF7w",
    "name": "grpcTest",
    "full_name": "amehlhase316/grpcTest",
    "private": false,
    "owner": {
      "login": "amehlhase316",
      "id": 46384989,
      "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
      "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
      "gravatar_id": "",
      "url": "https://api.github.com/users/amehlhase316",
      "html_url": "https://github.com/amehlhase316",
      "followers_url": "https://api.github.com/users/amehlhase316/followers",
      "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
      "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
      "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
      "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
      "repos_url": "https://api.github.com/users/amehlhase316/repos",
      "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
      "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
      "type": "User",
      "user_view_type": "public",
      "site_admin": false
    },
    "html_url": "https://github.com/amehlhase316/grpcTest",
    "description": null,
    "fork": false,
    "url": "https://api.github.com/repos/amehlhase316/grpcTest",
    "forks_url": "https://api.github.com/repos/amehlhase316/grpcTest/forks",
    "keys_url": "https://api.github.com/repos/amehlhase316/grpcTest/keys{/key_id}",
    "collaborators_url": "https://api.github.com/repos/amehlhase316/grpcTest/collaborators{/collaborator}",
    "teams_url": "https://api.github.com/repos/amehlhase316/grpcTest/teams",
    "hooks_url": "https://api.github.com/repos/amehlhase316/grpcTest/hooks",
    "issue_events_url": "https://api.github.com/repos/amehlhase316/grpcTest/issues/events{/number}",
    "events_url": "https://api.github.com/repos/amehlhase316/grpcTest/events",
    "assignees_url": "https://api.github.com/repos/amehlhase316/grpcTest/assignees{/user}",
    "branches_url": "https://api.github.com/repos/amehlhase316/grpcTest/branches{/branch}",
    "tags_url": "https://api.github.com/repos/amehlhase316/grpcTest/tags",
    "blobs_url": "https://api.github.com/repos/amehlhase316/grpcTest/git/blobs{/sha}",
```


https://api.github.com/repos/amehlhase316/memoranda



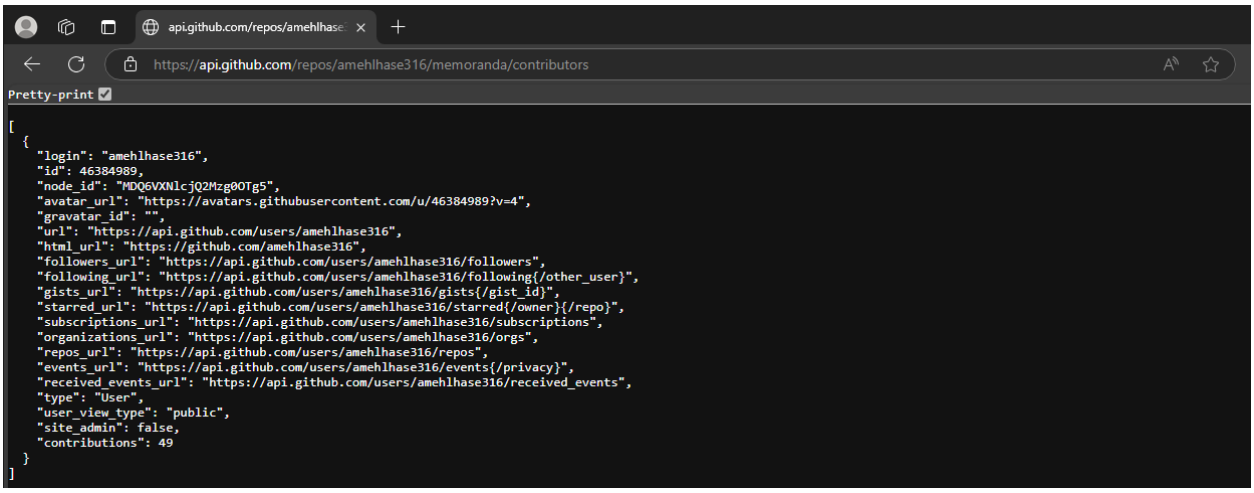

```
{
  "id": 164481338,
  "node_id": "MDEwOlJlcG9zaXRvcnkxNjQ0ODEzMzg=",
  "name": "memoranda",
  "full_name": "amehlhase316/memoranda",
  "private": false,
  "owner": {
    "login": "amehlhase316",
    "id": 46384989,
    "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
    "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amehlhase316",
    "html_url": "https://github.com/amehlhase316",
    "followers_url": "https://api.github.com/users/amehlhase316/followers",
    "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
    "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
    "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
    "repos_url": "https://api.github.com/users/amehlhase316/repos",
    "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
    "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false
  },
  "html_url": "https://github.com/amehlhase316/memoranda",
  "description": null,
  "fork": false,
  "url": "https://api.github.com/repos/amehlhase316/memoranda",
  "forks_url": "https://api.github.com/repos/amehlhase316/memoranda/forks",
  "keys_url": "https://api.github.com/repos/amehlhase316/memoranda/keys{/key_id}",
  "collaborators_url": "https://api.github.com/repos/amehlhase316/memoranda/collaborators{/collaborator}",
  "teams_url": "https://api.github.com/repos/amehlhase316/memoranda/teams",
  "hooks_url": "https://api.github.com/repos/amehlhase316/memoranda/hooks",
  "issue_events_url": "https://api.github.com/repos/amehlhase316/memoranda/issues/events{/number}",
  "events_url": "https://api.github.com/repos/amehlhase316/memoranda/events",
  "assignees_url": "https://api.github.com/repos/amehlhase316/memoranda/assignees{/user}",
  "branches_url": "https://api.github.com/repos/amehlhase316/memoranda/branches{/branch}",
  "tags_url": "https://api.github.com/repos/amehlhase316/memoranda/tags",
  "blobs_url": "https://api.github.com/repos/amehlhase316/memoranda/git/blobs{/sha}",
  "git_tags_url": "https://api.github.com/repos/amehlhase316/memoranda/git/tags{/sha}",
```

https://api.github.com/repos/amehlhase316/memoranda/commits?sha=amehlhase316-patch-1&per_page=40




```
[
  {
    "sha": "49fa683bfcde0feb65c38bd097bf265599617cb0",
    "node_id": "C_kwDOCc3JOtoAKDQ5ZmE2ODNiZmNkZTBmZWI2NWMzOGJkMDk3YmVyNjU1OTk2MTdjYjA",
    "commit": {
      "author": {
        "name": "amehlhase316",
        "email": "46384989+amehlhase316@users.noreply.github.com",
        "date": "2023-08-22T14:49:24Z"
      },
      "committer": {
        "name": "GitHub",
        "email": "noreply@github.com",
        "date": "2023-08-22T14:49:24Z"
      },
      "message": "Update DeliverableX.md\n\nMade checklist optional",
      "tree": {
        "sha": "7735c5477271fae444e7cd1ba6f760077c832e33",
        "url": "https://api.github.com/repos/amehlhase316/memoranda/git/trees/7735c5477271fae444e7cd1ba6f760077c832e33"
      },
      "url": "https://api.github.com/repos/amehlhase316/memoranda/git/commits/49fa683bfcde0feb65c38bd097bf265599617cb0",
      "comment_count": 0,
      "verification": {
        "verified": true,
        "reason": "valid",
        "signature": "-----BEGIN PGP SIGNATURE-----
\n\nwsBcBAABCAAQBQJk5Mr0CRBK7hj4Ov3rIwAAjAEIAIkEwgaRkrQRd0JsDUhup/ze\nTg8aeCFEiSEkNdY9C//dK/Cogwg5Q3H02+wxFhQ+rIAtWBQgmSeDSuPJl4gHA3oO\niNSZJE+xv8rBuVD1M+1bnmfniGwjDclcUDg1up
4hLXp/pnHh2TNnUgnHmsOV8QSvnF9D7zgHKgXStrHjdRxvRvB0otDadhsibM\nz2ZMt6wmiDdxuuTexzfeEutD9YxXRW8QV2iVHu3zrC83+s+whMipJ4zu1/kEJ6zB\nlmpu8kRzuh7xhX95cR4CMgP7XGcO5BDp7Uv4DULsV/Id/3
END PGP SIGNATURE-----\n",
        "payload": "tree 7735c5477271fae444e7cd1ba6f760077c832e33\nparent 25c65761c428ea91f640523ddce7de708c40238f\nauthor amehlhase316 <46384989+amehlhase316@users.noreply.g
-0700\ncommitter GitHub <noreply@github.com> 1692715764 -0700\n\nUpdate DeliverableX.md\n\nMade checklist optional",
        "verified_at": "2024-01-16T19:59:59Z"
      }
    },
    "url": "https://api.github.com/repos/amehlhase316/memoranda/commits/49fa683bfcde0feb65c38bd097bf265599617cb0",
    "html_url": "https://github.com/amehlhase316/memoranda/commit/49fa683bfcde0feb65c38bd097bf265599617cb0",
    "comments_url": "https://api.github.com/repos/amehlhase316/memoranda/commits/49fa683bfcde0feb65c38bd097bf265599617cb0/comments",
    "author": {
      "login": "amehlhase316",
      "id": 46384989,
      "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
      "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
      "gravatar_id": "",
```


https://api.github.com/repos/amehlhase316/memoranda/contributors




```
[
  {
    "login": "amehlhase316",
    "id": 46384989,
    "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
    "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amehlhase316",
    "html_url": "https://github.com/amehlhase316",
    "followers_url": "https://api.github.com/users/amehlhase316/followers",
    "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
    "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
    "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
    "repos_url": "https://api.github.com/users/amehlhase316/repos",
    "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
    "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false,
    "contributions": 49
  }
]
```

1. API call #1: List repositories for a user
    a. https://api.github.com/users/amehlhase316/repos
    b. Retrieves a list of all public repositories owned by the user and its metadata
    c. Requires: GitHub username
    d. https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#list-repositories-for-a-user

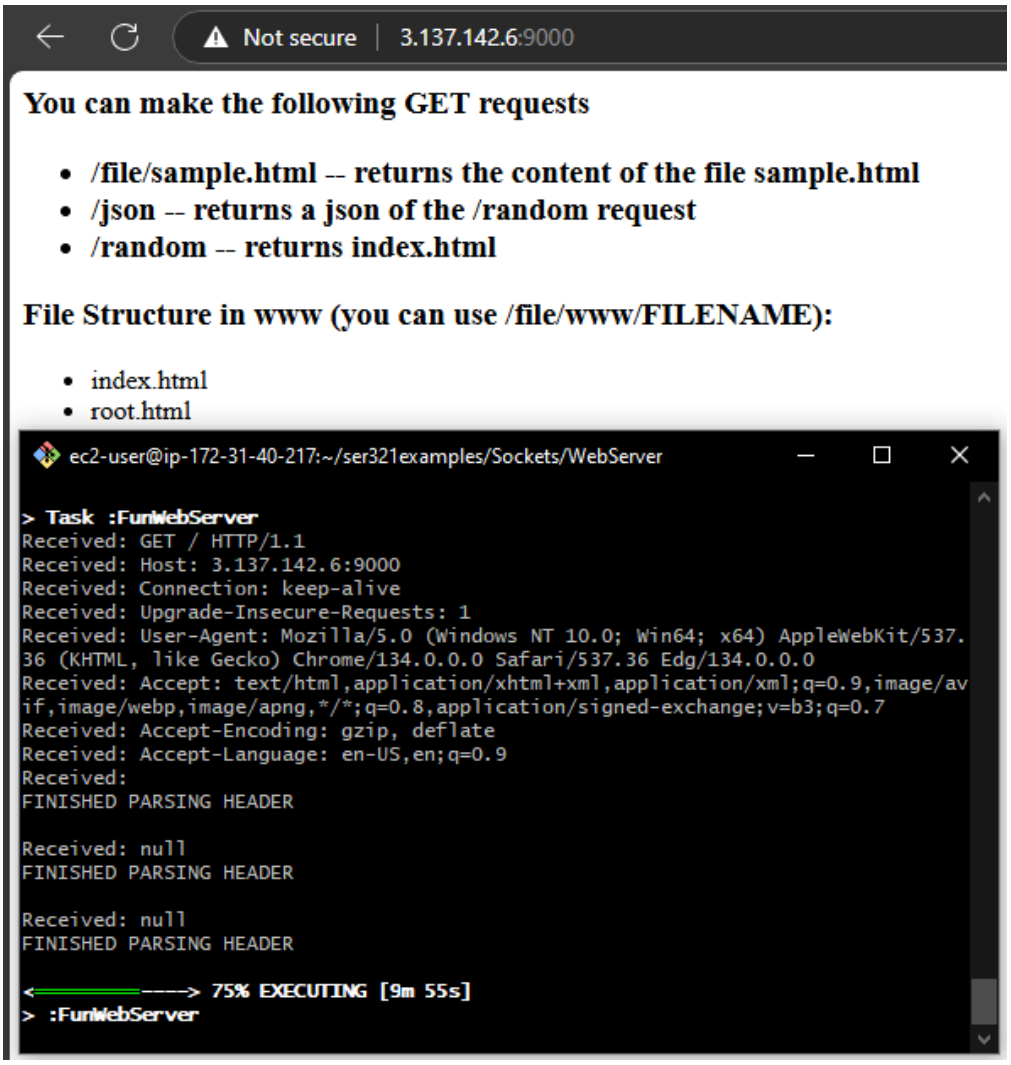
    API call #2: Get a specific repository
    e. https://api.github.com/repos/amehlhase316/memoranda
    f. Provides detailed metadata for a single public repository (name, owner, etc,..)
    g. Requires: GitHub username and repository name
    h. https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#get-a-repository

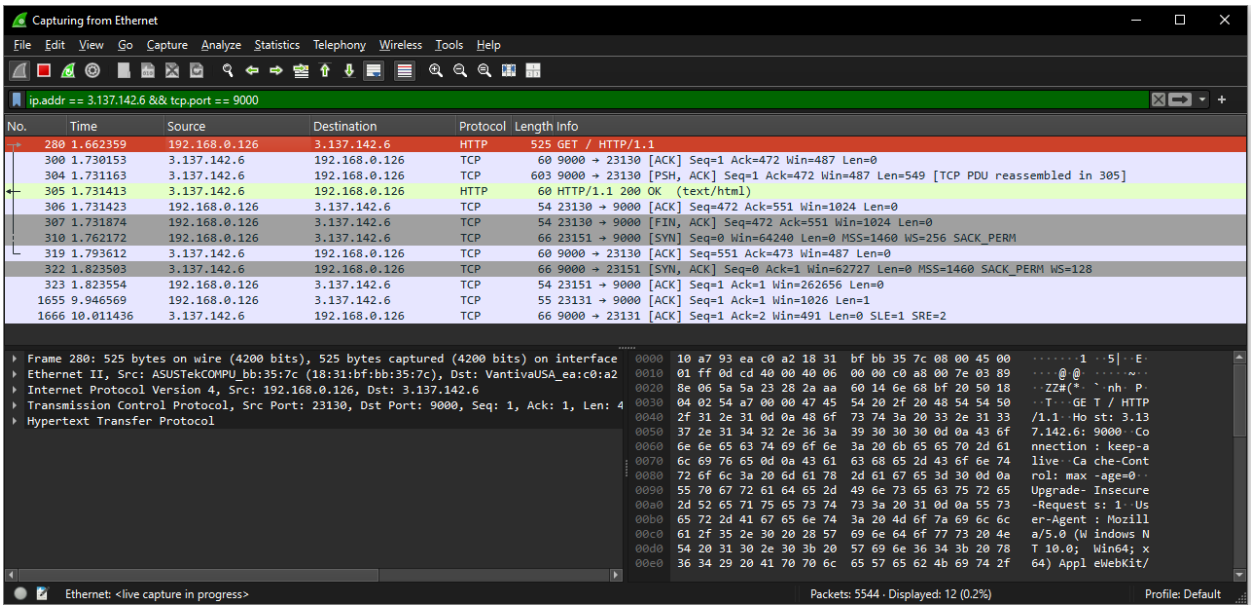
    API call #3: List commits on a repository
    i. https://api.github.com/repos/amehlhase316/memoranda/commits?sha=amehlhase316-patch-1&per_page=40
    j. Retrieves a list of commits from a specified branch with # of results per page.
    k. Required: GitHub username, repo name, branch (sha), and # of results (per_page).
    l. https://docs.github.com/en/rest/commits/commits?apiVersion=2022-11-28#list-commits

    API call #4: List contributors of a repository
    m. https://api.github.com/repos/amehlhase316/memoranda/contributors
    n. Returns a list of contributors to the specified repository, including the number of contributions made by each contributor.
    o. Required: GitHub username and repository name
    p. https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#list-repository-contributors
2. Stateless vs Stateful Communication

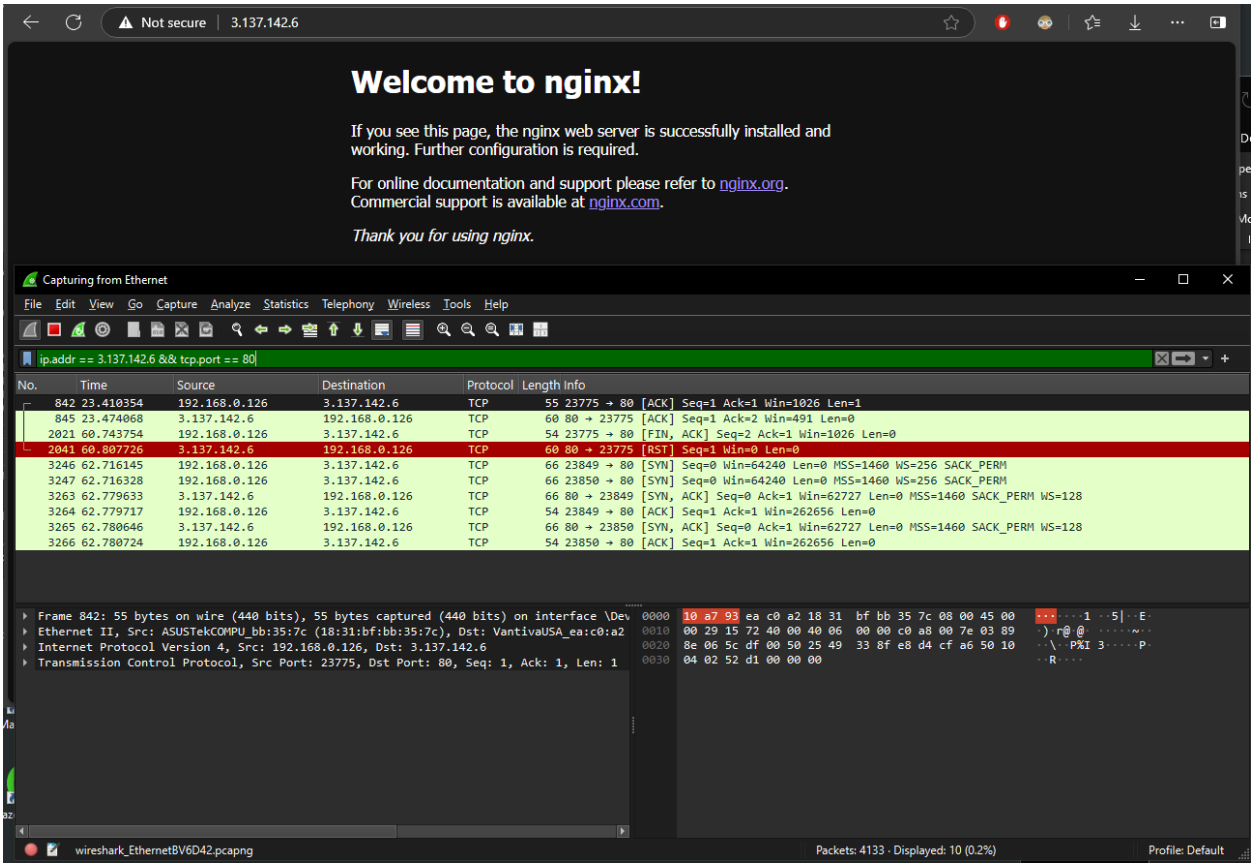
    In stateless communication, each request from a client to a server is treated as an independent transaction. The server does not store any information about previous requests. HTTP, which is used for web browsing and REST API calls like the ones in this assignment, is a stateless protocol. Each request contains all the information the server needs to process it (e.g., headers, query parameters). In contrast, stateful communication involves the server maintaining information (or "state") about the client between requests. Examples include protocols like FTP or Telnet, where the server keeps track of the session state throughout the interaction. Statelessness makes HTTP simpler and more scalable but also requires clients to resend context with every request.

Task 2

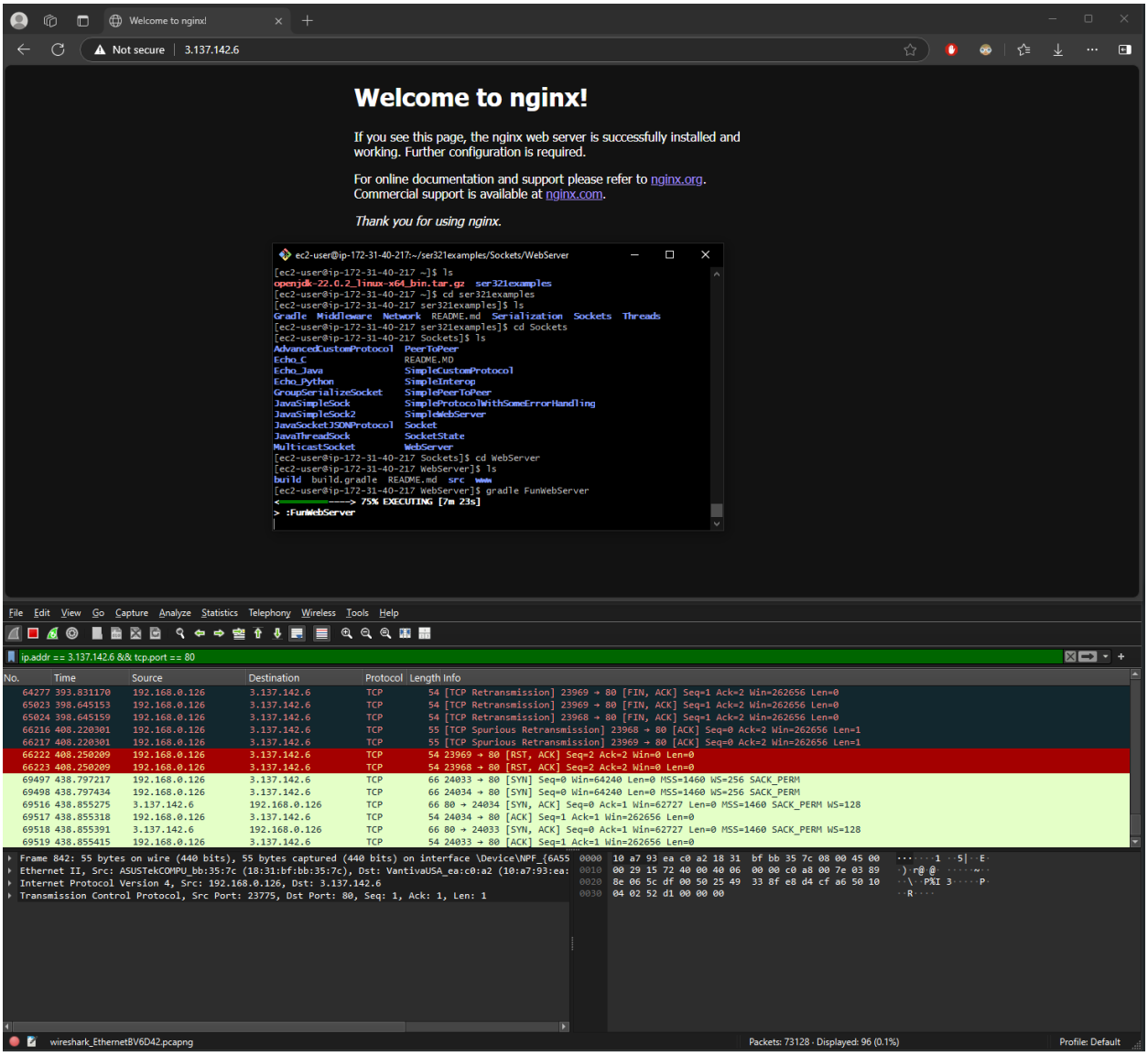

Not secure
3.137.142.6:9000
You can make the following GET requests
/file/sample.html -- returns the content of the file sample.html
/json -- returns a json of the /random request
/random -- returns index.html
File Structure in www (you can use /file/www/FILENAME):
index.html
root.html
ec2-user@ip-172-31-40-217:~/ser321examples/Sockets/WebServer
> Task :FunWebServer
Received: GET / HTTP/1.1
Received: Host: 3.137.142.6:9000
Received: Connection: keep-alive
Received: Upgrade-Insecure-Requests: 1
Received: User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.
36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 Edg/134.0.0.0
Received: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/av
if,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9
Received:
FINISHED PARSING HEADER
Received: null
FINISHED PARSING HEADER
Received: null
FINISHED PARSING HEADER
<----------> 75% EXECUTING [9m 55s]
> :FunWebServer


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 3.137.142.6 && tcp.port == 9000
No. Time Source Destination Protocol Length Info
280 1.662359 192.168.0.126 3.137.142.6 HTTP 525 GET / HTTP/1.1
300 1.730153 3.137.142.6 192.168.0.126 TCP 60 9000 → 23130 [ACK] Seq=1 Ack=472 Win=487 Len=0
304 1.731163 3.137.142.6 192.168.0.126 TCP 603 9000 → 23130 [PSH, ACK] Seq=1 Ack=472 Win=487 Len=549 [TCP PDU reassembled in 305]
305 1.731413 3.137.142.6 192.168.0.126 HTTP 60 HTTP/1.1 200 OK (text/html)
306 1.731423 192.168.0.126 3.137.142.6 TCP 54 23130 → 9000 [ACK] Seq=472 Ack=551 Win=1024 Len=0
307 1.731874 192.168.0.126 3.137.142.6 TCP 54 23130 → 9000 [FIN, ACK] Seq=472 Ack=551 Win=1024 Len=0
310 1.762172 192.168.0.126 3.137.142.6 TCP 66 23151 → 9000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
319 1.793612 3.137.142.6 192.168.0.126 TCP 60 9000 → 23130 [ACK] Seq=551 Ack=473 Win=487 Len=0
322 1.823503 3.137.142.6 192.168.0.126 TCP 66 9000 → 23151 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128
323 1.823554 192.168.0.126 3.137.142.6 TCP 54 23151 → 9000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
1655 9.946569 192.168.0.126 3.137.142.6 TCP 55 23131 → 9000 [ACK] Seq=1 Ack=1 Win=1026 Len=1
1666 10.011436 3.137.142.6 192.168.0.126 TCP 66 9000 → 23131 [ACK] Seq=1 Ack=2 Win=491 Len=0 SLE=1 SRE=2
Frame 280: 525 bytes on wire (4200 bits), 525 bytes captured (4200 bits) on interface
Ethernet II, Src: ASUSTekCOMPU_bb:35:7c (18:31:bf:bb:35:7c), Dst: VantivaUSA_ea:c0:a2
Internet Protocol Version 4, Src: 192.168.0.126, Dst: 3.137.142.6
Transmission Control Protocol, Src Port: 23130, Dst Port: 9000, Seq: 1, Ack: 1, Len: 4
Hypertext Transfer Protocol
0000 10 a7 93 ea c0 a2 18 31 bf bb 35 7c 08 00 45 00
0010 01 ff 0d cd 40 00 40 06 00 00 c0 a8 00 7e 03 89
0020 8e 06 5a 5a 23 28 2a aa 60 14 6e 68 bf 20 50 18
0030 04 02 54 a7 00 00 47 45 54 20 2f 20 48 54 54 50
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 33 2e 31 33
0050 37 2e 31 34 32 2e 36 3a 39 30 30 30 0d 0a 43 6f
0060 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61
0070 6c 69 76 65 0d 0a 43 61 63 68 65 2d 43 6f 6e 74
0080 72 6f 6c 3a 20 6d 61 78 2d 61 67 65 3d 30 0d 0a
0090 55 70 67 72 61 64 65 2d 49 6e 73 65 63 75 72 65
00a0 2d 52 65 71 75 65 73 74 73 3a 20 31 0d 0a 55 73
00b0 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c
00c0 61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 20 4e
00d0 54 20 31 30 2e 30 3b 20 57 69 6e 36 34 3b 20 78
00e0 36 34 29 20 41 70 70 6c 65 57 65 62 4b 69 74 2f
Ethernet: <live capture in progress>
Packets: 5544 · Displayed: 12 (0.2%)
Profile: Default

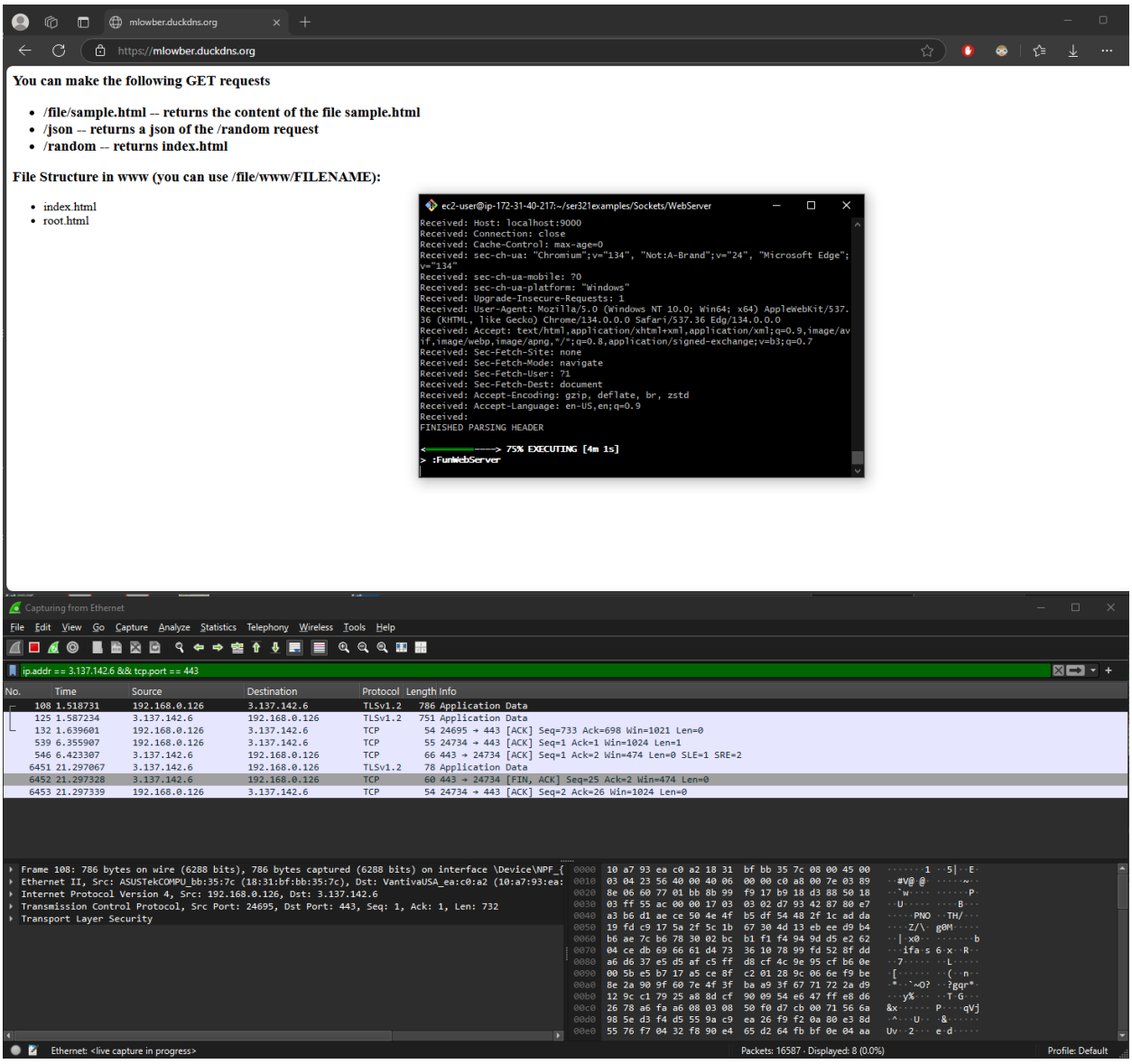
1. I used the wireshark filter: “ip.addr == 3.137.142.6 && tcp.port == 9000”
   This filter shows all traffic either to or from my EC2 server on the specified port. I chose this filter because it captures both the requests from my browser and the responses from the server, allowing me to observe the full HTTP conversation.
2. When I first visit the ‘/random’ page, I am met by “bread” and a picture of some beautiful bread. When clicking the random button, it cycles between “bread” and “streets” randomly. Refreshing the page visually has the exact same effect, but because it reloaded the entire page and re-fetched all resources, it resulted in more network traffic than just pressing the random button.
3. I observed the following response codes:
   a. ‘200 OK’ for successful page loads or valid requests.
   b. ‘404 Not Found’ when accessing a non-existent route.
   c. ‘400 Bad Request’ for malformed or missing parameters in requests like ‘/multiply’.
4. Explanations:
   a. Indicates a successful request where the server found and returned the expected resource.
   b. Occurs when I tried to access an endpoint or file that doesn’t exist on the server.
   c. Triggered when I tested endpoints like ‘/multiply’ with missing or invalid parameters. This is a good indicator of input validation working correctly.
5. Yes, I was able to find the server response in the Wireshark capture. By expanding the “Hypertext Transfer Protocol” section of the HTTP response packet, I could see the HTML content of the web page. It appeared under the “Line-based text data” section in plain text.
6. HTTP sends data in plain text, which means anyone monitoring the network (like with Wireshark) can see exactly what’s being transmitted — including sensitive information. In contrast, HTTPS encrypts the traffic, protecting both privacy and integrity. That’s why HTTPS is now the industry standard for secure communication on the web.
7. My server listens on port 9000, which is a custom port chosen for development. The most common/default port for HTTP is port 80. Port 9000 works as long as the client knows to request it specifically in the URL.
8. Each HTTP request from my local machine used what I found to be called ephemeral ports (high-numbered port like 51000–65000). I learned that these are dynamically assigned by the operating system to handle outgoing connections and are reused across different requests.



1. http://3.137.142.6 now works because nginx is forwarding port 80 traffic to the Java server running on port 9000
2. Traffic from the browser is going to port 80 (default HTTP). This is different from before, where it was going directly to port 9000. This is expected and intended because nginx listens on port 80 and proxies the request internally.
3. It's still using HTTP, not HTTPS. The browser shows 'http://' and traffic is unencrypted. I haven't yet configured TLS/SSL certificates for HTTPS, which seems to happen in the next section.
4. Yes. Now that the server is accepting traffic on port 80 instead of 9000, I could remove or restrict access to port 9000 in the AWS EC2 security group and allow only port 80 to open. This would make the server more secure by exposing fewer entry points.

5. 
Welcome to nginx!
Not secure 3.137.142.6
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.
ec2-user@ip-172-31-40-217:~/ser321examples/Sockets/WebServer
[ec2-user@ip-172-31-40-217 ~]$ ls
openjdk-22.0.2_linux-x64_bin.tar.gz ser321examples
[ec2-user@ip-172-31-40-217 ~]$ cd ser321examples
[ec2-user@ip-172-31-40-217 ser321examples]$ ls
Gradle Middleware Network README.md Serialization Sockets Threads
[ec2-user@ip-172-31-40-217 ser321examples]$ cd Sockets
[ec2-user@ip-172-31-40-217 Sockets]$ ls
AdvancedCustomProtocol PeerToPeer
Echo_C README.MD
Echo_Java SimpleCustomProtocol
Echo_Python SimpleInterop
GroupSerializeSocket SimplePeerToPeer
JavaSimpleSock SimpleProtocolWithSomeErrorHandling
JavaSimpleSock2 SimpleWebServer
JavaSocketJSONProtocol Socket
JavaThreadSock SocketState
MulticastSocket WebServer
[ec2-user@ip-172-31-40-217 Sockets]$ cd WebServer
[ec2-user@ip-172-31-40-217 WebServer]$ ls
build build.gradle README.md src www
[ec2-user@ip-172-31-40-217 WebServer]$ gradle FunWebServer
<-------------> 75% EXECUTING [7m 23s]
> :FunWebServer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 3.137.142.6 && tcp.port == 80
No. Time Source Destination Protocol Length Info
64277 393.831170 192.168.0.126 3.137.142.6 TCP 54 [TCP Retransmission] 23969 → 80 [FIN, ACK] Seq=1 Ack=2 Win=262656 Len=0
65023 398.645153 192.168.0.126 3.137.142.6 TCP 54 [TCP Retransmission] 23969 → 80 [FIN, ACK] Seq=1 Ack=2 Win=262656 Len=0
65024 398.645159 192.168.0.126 3.137.142.6 TCP 54 [TCP Retransmission] 23968 → 80 [FIN, ACK] Seq=1 Ack=2 Win=262656 Len=0
66216 408.220301 192.168.0.126 3.137.142.6 TCP 55 [TCP Spurious Retransmission] 23968 → 80 [ACK] Seq=0 Ack=2 Win=262656 Len=1
66217 408.220301 192.168.0.126 3.137.142.6 TCP 55 [TCP Spurious Retransmission] 23969 → 80 [ACK] Seq=0 Ack=2 Win=262656 Len=1
66222 408.250209 192.168.0.126 3.137.142.6 TCP 54 23969 → 80 [RST, ACK] Seq=2 Ack=2 Win=0 Len=0
66223 408.250209 192.168.0.126 3.137.142.6 TCP 54 23968 → 80 [RST, ACK] Seq=2 Ack=2 Win=0 Len=0
69497 438.797217 192.168.0.126 3.137.142.6 TCP 66 24033 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
69498 438.797434 192.168.0.126 3.137.142.6 TCP 66 24034 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
69516 438.855275 3.137.142.6 192.168.0.126 TCP 66 80 → 24034 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128
69517 438.855318 192.168.0.126 3.137.142.6 TCP 54 24034 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
69518 438.855391 3.137.142.6 192.168.0.126 TCP 66 80 → 24033 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128
69519 438.855415 192.168.0.126 3.137.142.6 TCP 54 24033 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
Frame 842: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{6A55
Ethernet II, Src: ASUSTekCOMPU_bb:35:7c (18:31:bf:bb:35:7c), Dst: VantivaUSA_ea:c0:a2 (10:a7:93:ea:
Internet Protocol Version 4, Src: 192.168.0.126, Dst: 3.137.142.6
Transmission Control Protocol, Src Port: 23775, Dst Port: 80, Seq: 1, Ack: 1, Len: 1
0000 10 a7 93 ea c0 a2 18 31 bf bb 35 7c 08 00 45 00
0010 00 29 15 72 40 00 40 06 00 00 c0 a8 00 7e 03 89
0020 8e 06 5c df 00 50 25 49 33 8f e8 d4 cf a6 50 10
0030 04 02 52 d1 00 00 00
wireshark_EthernetBV6D42.pcapng
Packets: 73128 · Displayed: 96 (0.1%)
Profile: Default

2.5



1. Port 443, used for secure HTTPS communication.
2. No, HTTPS encrypts all data, so I can no longer see HTML or other content in Wireshark—only metadata.

2.6.1) For '/multiply', I added error handling to ensure the server responds well when clients submit incorrect or incomplete requests. I parsed the 'num1' and 'num2' query parameters from the URL. I checked if either parameter was missing, and responded with an error if so. I also used try/catch to handle cases where the inputs couldn't be parsed as integers. Finally, I ensured the server never crashes—every path results in a valid HTTP response.

Error codes used and why:

400 Bad Request: Appropriate because the client submitted invalid or incomplete input — a classic case of malformed request syntax or missing data.

200 OK: The standard success response for well-formed HTTP GET requests.

500 Internal Server Error: Ensures that even unexpected server-side issues don't crash the server and gives a clear indication that something went wrong internally.