

Link: <https://github.com/pojha12/ser321-summer23-C-pojha1>

<https://api.github.com/repos/pojha12/assign1git/commits/main>

```
api.github.com/repos/pojha12/assign1git/commits/main

{
  "sha": "21867cbd97219369671318adb3f03925e912e907",
  "node_id": "C_kwDOIPEqrtoAKDIXODY3Y2JkOTcyMTkzNjZkZmZlZGZlZjAsOTI1ZTkxMmU5MDc",
  "commit": {
    "author": {
      "name": "pojha12@asu.edu",
      "email": "palak@MacBook-Pro.attlocal.net",
      "date": "2022-10-17T18:45:05Z"
    },
    "committer": {
      "name": "pojha12@asu.edu",
      "email": "palak@MacBook-Pro.attlocal.net",
      "date": "2022-10-17T18:45:05Z"
    },
    "message": "change on newbranch",
    "tree": {
      "sha": "2013e19af97afa66c51f316038e334bc5481ea41",
      "url": "https://api.github.com/repos/pojha12/assign1git/git/trees/2013e19af97afa66c51f316038e334bc5481ea41"
    },
    "url": "https://api.github.com/repos/pojha12/assign1git/git/commits/21867cbd97219369671318adb3f03925e912e907",
    "comment_count": 0,
    "verification": {
      "verified": false,
      "reason": "unsigned",
      "signature": null,
      "payload": null
    }
  },
  "url": "https://api.github.com/repos/pojha12/assign1git/commits/21867cbd97219369671318adb3f03925e912e907",
  "html_url": "https://github.com/pojha12/assign1git/commit/21867cbd97219369671318adb3f03925e912e907",
  "comments_url": "https://api.github.com/repos/pojha12/assign1git/commits/21867cbd97219369671318adb3f03925e912e907/comments",
  "author": null,
  "committer": null,
  "parents": [
    {
      "sha": "00f45a225ba4fcd616ac8de5c2234da05bfeb33a",
      "url": "https://api.github.com/repos/pojha12/assign1git/commits/00f45a225ba4fcd616ac8de5c2234da05bfeb33a",
      "html_url": "https://github.com/pojha12/assign1git/commit/00f45a225ba4fcd616ac8de5c2234da05bfeb33a"
    }
  ],
  "stats": {
    "total": 4,
    "additions": 3,
    "deletions": 1
  },
  "files": [
    {
      "sha": "056e396ea010862a25b2cee4c6e24a4d3b70d0db",
      "type": "file",
      "mode": "100644",
      "path": "newbranch"
    }
  ]
}
```

https://api.github.com/repos/pojha12/assign1git/branches/newbranch?per_page=50

```
api.github.com/repos/pojha12/assign1git/branches/newbranch?per_page=50

{
  "name": "newbranch",
  "commit": {
    "sha": "21867cbd97219369671318adb3f03925e912e907",
    "node_id": "C_kwDOIPEqrtoAKDIXODY3Y2JkOTcyMTkzNjZkZmZlZGZlZjAsOTI1ZTkxMmU5MDc",
    "commit": {
      "author": {
        "name": "pojha12@asu.edu",
        "email": "palak@MacBook-Pro.attlocal.net",
        "date": "2022-10-17T18:45:05Z"
      },
      "committer": {
        "name": "pojha12@asu.edu",
        "email": "palak@MacBook-Pro.attlocal.net",
        "date": "2022-10-17T18:45:05Z"
      },
      "message": "change on newbranch",
      "tree": {
        "sha": "2013e19af97afa66c51f316038e334bc5481ea41",
        "url": "https://api.github.com/repos/pojha12/assign1git/git/trees/2013e19af97afa66c51f316038e334bc5481ea41"
      },
      "url": "https://api.github.com/repos/pojha12/assign1git/git/commits/21867cbd97219369671318adb3f03925e912e907",
      "comment_count": 0,
      "verification": {
        "verified": false,
        "reason": "unsigned",
        "signature": null,
        "payload": null
      }
    },
    "url": "https://api.github.com/repos/pojha12/assign1git/commits/21867cbd97219369671318adb3f03925e912e907",
    "html_url": "https://github.com/pojha12/assign1git/commit/21867cbd97219369671318adb3f03925e912e907",
    "comments_url": "https://api.github.com/repos/pojha12/assign1git/commits/21867cbd97219369671318adb3f03925e912e907/comments",
    "author": null,
    "committer": null,
    "parents": [
      {
        "sha": "00f45a225ba4fcd616ac8de5c2234da05bfeb33a",
        "url": "https://api.github.com/repos/pojha12/assign1git/commits/00f45a225ba4fcd616ac8de5c2234da05bfeb33a",
        "html_url": "https://github.com/pojha12/assign1git/commit/00f45a225ba4fcd616ac8de5c2234da05bfeb33a"
      }
    ]
  },
  "links": {
    "self": "https://api.github.com/repos/pojha12/assign1git/branches/newbranch",
    "html": "https://github.com/pojha12/assign1git/tree/newbranch"
  }
}
```

1. Explain the specific API calls you used.

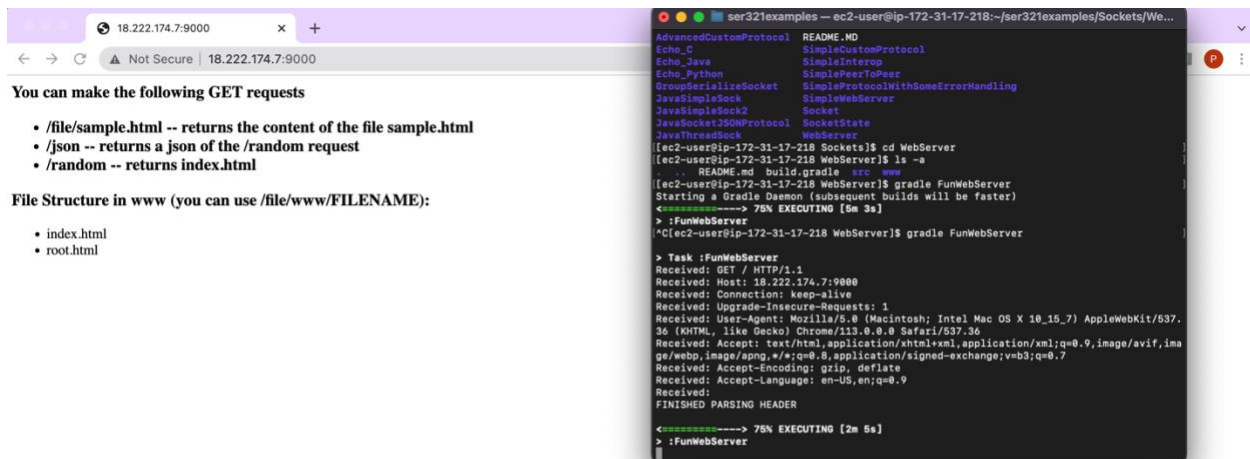
In the first call, I used main as the default branch. Commits/master gave me any commits the user had on the master branch. In the second call, I got the requests made in a specific branch. It was followed by the per_page argument which specifies the number of commits that are visible.

2. Explain the difference between stateless and a stateful communication.

Stateless communication is when the response of a request doesn't depend on a previous request. The protocol HTTP/HTTPs doesn't require the server to retain the session information.

Stateful communication is when the requests are dependent on the previous requests. The protocol requires the server to save the session information.

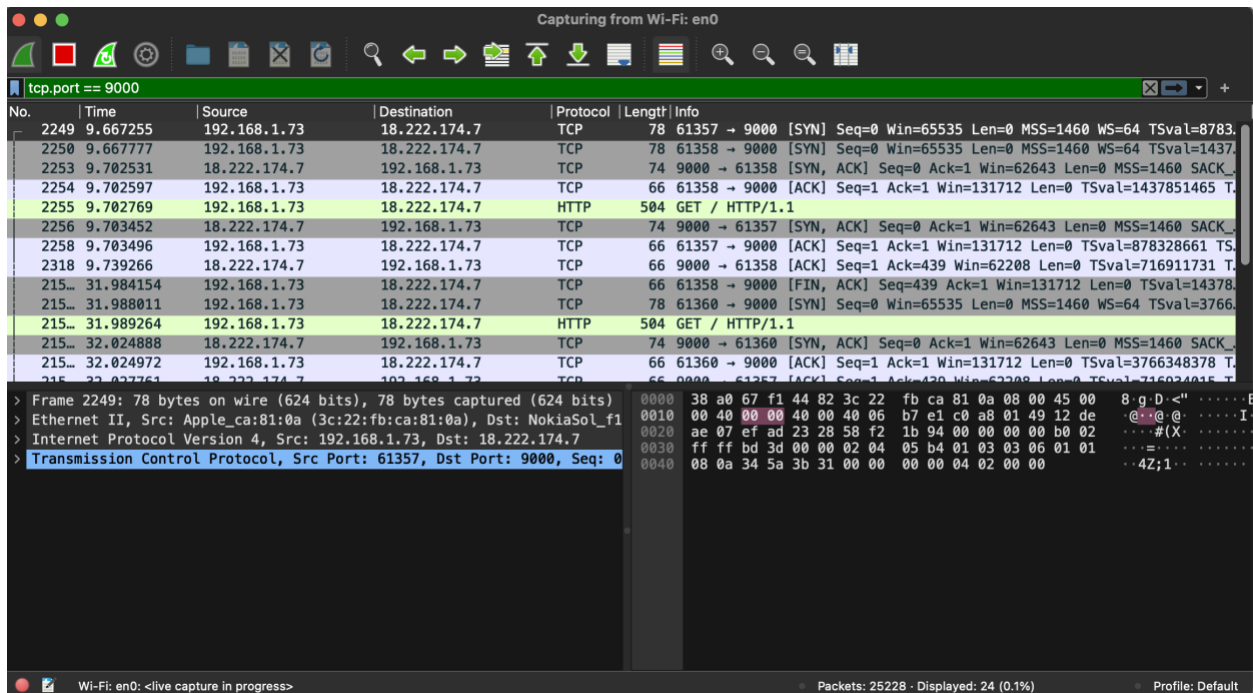
2.2



The screenshot shows a web browser on the left and a terminal window on the right. The browser displays a page with the URL 18.222.174.7:9000 and the text "You can make the following GET requests". Below this, there is a list of requests: /file/sample.html, /json, and /random. The terminal window shows the command prompt for a web server, with the following output:

```
AdvancedCustomProtocol README.MD
Echo_C SimpleCustomProtocol
Echo_Java SimpleInterop
Echo_Python SimplePeerToPeer
GroupSerializeSocket SimpleProtocolWithSomeErrorHandling
JavaSimpleSock SimpleWebServer
JavaSimpleSock2 Socket
JavaSocket2SOMProtocol SocketState
JavaThreadSock WebServer
[ec2-user@ip-172-31-17-218 Sockets]$ cd WebServer
[ec2-user@ip-172-31-17-218 WebServer]$ ls -la
total 12
-rw-r--r-- 1 ec2-user ec2-user 4096 Nov 14 14:14 README.md
-rw-r--r-- 1 ec2-user ec2-user 112 Nov 14 14:14 build.gradle
-rw-r--r-- 1 ec2-user ec2-user 112 Nov 14 14:14 src
[ec2-user@ip-172-31-17-218 WebServer]$ gradle FunWebServer
Starting a Gradle Daemon (subsequent builds will be faster)
<=====--> 75% EXECUTING [5m 3s]
> :FunWebServer
^C[ec2-user@ip-172-31-17-218 WebServer]$ gradle FunWebServer
> Task :FunWebServer
Received: GET / HTTP/1.1
Received: Host: 18.222.174.7:9000
Received: Connection: keep-alive
Received: Upgrade-Insecure-Requests: 1
Received: User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
Received: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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
<=====--> 75% EXECUTING [2m 5s]
> :FunWebServer
```

2.3



1. What filter did you use? Explain why you chose that filter.
The filter I used was `tcp.port == 9000`. We are dealing with a tcp connection and 9000 is the port that I was connecting to.
2. What happens when you are on /random and click the "Random" button compared to the browser refresh (you can also use the command line output that the WebServer generates to answer this)?
When the random button is clicked, the server receives a `GET /json HTTP/1.1`. When you refresh, the server receives a `GET /random HTTP/1.1` and then a `GET /json HTTP/1.1`.
3. What kinds of response codes are you able to get through different requests to your server?
I got 200, 400, and 404.
4. Explain the response codes you get and why you get them.
Response code of 200 means that the request was successful. Response code of 400 means that the request was bad on the client side. Response code of 404 means that the server couldn't find the requested resource.
5. When you do a `ipOfSecondMachine:9000` take a look what Wireshark generates as a server response. Are you able to find the data that the server sends back to you?
Yes I am able to see.
6. Based on the above question explain why HTTPS is now more common than HTTP.
HTTPS is more secure than HTTP.

7. What port does the server listen to for HTTP requests in our case and is that the most common port for HTTP?
The server listened for port 9000 and the most common port is 80.
8. What local port is used when sending different requests to the WebServer? How does it differ to the traffic to your SMTP server from part 1?
The port is 59948.

2.4

1. What is the URL that you can use now to reach the main page?
<http://3.23.99.252:9000/>
2. Check your traffic to your Webserver. What port is the traffic going to now? Is it the same as previously used or is it and should it be different?
The traffic is going through port 80 now. It was using port 9000 before.
3. Is it still HTTP or is it now HTTPs? Why?
It's still HTTP.

4.

Wi-Fi: en0

tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
360	25.220341	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	TCP	98	65180 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1440 WS=64 TSval=3079920989
361	25.220432	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	TCP	98	65181 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1440 WS=64 TSval=1288552741
362	25.241406	2620:149:a32:f100:...	2600:1700:3fe3:ed1...	TCP	94	80 → 65181 [SYN, ACK] Seq=0 Ack=1 Win=64260 Len=0 MSS=1220 SACK_PE
363	25.241407	2620:149:a32:f100:...	2600:1700:3fe3:ed1...	TCP	94	80 → 65180 [SYN, ACK] Seq=0 Ack=1 Win=64260 Len=0 MSS=1220 SACK_PE
364	25.262007	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	TCP	86	65180 → 80 [ACK] Seq=1 Ack=1 Win=2432 Len=0 TSval=3079921028 TSecr
365	25.262016	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	HTTP	239	CONNECT proxy-safebrowsing.googleapis.com:443 HTTP/1.1
366	25.262088	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	TCP	86	65181 → 80 [ACK] Seq=1 Ack=1 Win=2432 Len=0 TSval=1288552780 TSecr
367	25.262091	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	HTTP	239	CONNECT proxy-safebrowsing.googleapis.com:443 HTTP/1.1
368	25.285938	2620:149:a32:f100:...	2600:1700:3fe3:ed1...	TCP	86	80 → 65181 [ACK] Seq=1 Ack=154 Win=64256 Len=0 TSval=3609459940 TS
369	25.286671	2620:149:a32:f100:...	2600:1700:3fe3:ed1...	HTTP	312	HTTP/1.1 200 OK
370	25.286671	2620:149:a32:f100:...	2600:1700:3fe3:ed1...	TCP	86	80 → 65180 [ACK] Seq=1 Ack=154 Win=64256 Len=0 TSval=3228876375 TS
371	25.287725	2620:149:a32:f100:...	2600:1700:3fe3:ed1...	HTTP	312	HTTP/1.1 200 OK
372	25.317560	2600:1700:3fe3:ed1...	2620:149:a32:f100:...	TCP	86	65180 → 80 [ACK] Seq=154 Ack=227 Win=2432 Len=0 TSval=3079921078 T

> Frame 369: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bit

> Ethernet II, Src: NokiaSol_f1:44:82 (38:a0:67:f1:44:82), Dst: Apple_ca

> Internet Protocol Version 6, Src: 2620:149:a32:f100::234, Dst: 2600:17

> Transmission Control Protocol, Src Port: 80, Dst Port: 65181, Seq: 1,

> Hypertext Transfer Protocol

0000 7d 66 06 60 bd 9b 00 50 fe 9d 89 50 2f c6 20 5a }f...P...P/.

0040 13 c6 80 18 00 fb cc 84 00 00 01 01 08 0a d7 23

0060 fc e4 4c cd c1 4e 48 54 54 50 2f 31 2e 31 20 32 ..L..NHT TP/1.1

0080 30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 46 72 69 00 OK ·D ate: F

00a0 2c 20 32 36 20 4d 61 79 20 32 30 32 33 20 32 30 , 26 May 2023

00c0 3a 35 36 3a 32 30 20 47 4d 54 0d 0a 50 72 6f 78 :56:20 G MT·Pr

00e0 79 2d 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 y-Connec tion:

0100 65 70 2d 61 6c 69 76 65 0d 0a 56 69 61 3a 20 68 ep-alive ·Via:

0120 74 74 70 2f 31 2e 31 20 75 73 64 61 6c 34 2d 65 ttp/1.1 usdaL4

0140 64 67 65 2d 67 65 74 2d 30 30 38 2e 74 73 2e 61 dge-get- 008.ts

0160 70 70 6c 65 2e 63 6f 6d 20 28 61 63 64 6e 2f 38 pple.com (acdn

0180 37 2e 31 34 33 30 31 29 0d 0a 58 2d 43 61 63 68 7.14301) ·X-Ca

01a0 65 3a 20 73 6b 69 70 70 65 64 0d 0a 43 44 4e 55 e: skip ed·CD

01c0 55 49 44 3a 20 34 66 65 30 62 32 65 36 2d 61 61 UID: 4fe 0b2e6-

01e0 35 65 2d 34 38 63 32 2d 62 33 39 34 2d 63 63 32 5e-48c2- b394-c

0200 61 33 64 61 37 62 38 32 30 2d 32 39 37 35 38 37 a3da7b82 0-2975

0220 36 30 34 30 0d 0a 0d 0a 6040...

Transmission Control Protocol (tcp), 32 bytes

Packets: 43033 · Displayed: 260 (0.6%)

Profile: Default

← → ↻ 🔒 Not Secure | 3.23.99.252:9000

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

2.6.1

I used 206 for content missing. This is to ensure that there are 2 arguments when the request is made since multiplication cannot be done without 2 numbers. I also used 400 for bad request.