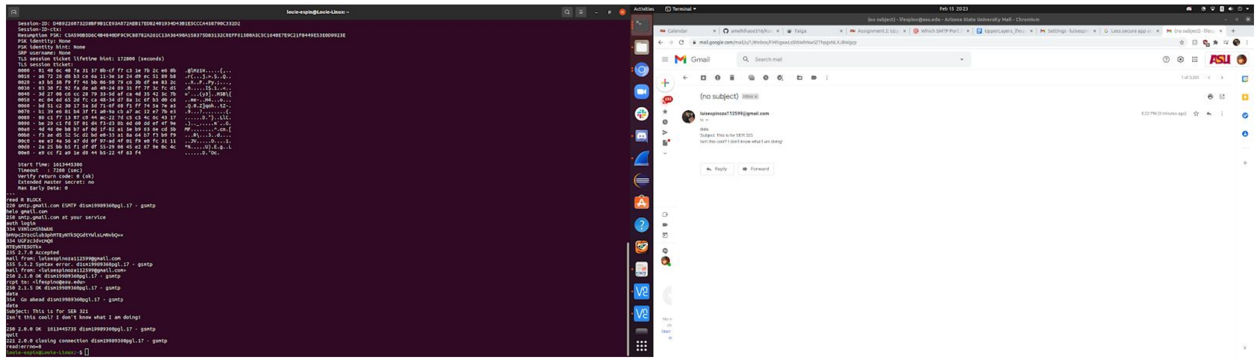
1. Sending emails and SMTP

Sending First Email:



```
    Session-ID: D4892260732D8BF9B1CE93A072AEB17ED82401934D43B1E5CCCA458790C332D2
    Session-ID-ctx:
    Resumption PSK: C5A590B5D6C4B4840DF9C9C88782A261C13A36490A158375D83132C0EFF01500A5C5C1640E7E9C21F8449E53E0D9923E
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 172800 (seconds)
    TLS session ticket:
    0000 - 01 40 6c 48 7a 31 57 8b-cf f7 c3 1e 7b 2c e0 8b   .@lHz1W.....{,..
    0010 - a6 72 28 d8 b3 ce 6a 11-3e 1e 24 d9 ec 51 89 b8   .r(...j.>.$..Q..
    0020 - a3 b5 58 f9 f7 46 bb 06-50 79 c6 3b df ae 83 2c   ..X..F..Py.;...,
    0030 - 83 30 f2 92 fa de a8 49-24 89 31 ff 7f 3c fc d5   .0.....I$.1..<..
    0040 - 3d 27 08 c6 cc 28 79 33-5d af ca 4d 35 42 5c 7b   ='...(y3]..M5B\{
    0050 - ec 04 6d 65 2d fc ca 48-34 d7 8a 1c 6f b3 d0 c6   ..me-..H4...o...
    0060 - bd 51 c2 30 17 5a 5d 71-6f d8 f1 ff 74 5a 7e a5   .Q.0.Z]qoh..tZ~.
    0070 - b1 39 e6 81 b4 3f f1 a0-9a cb a7 ac 12 e7 7b e3   .9...?........{.
    0080 - 86 c1 f7 13 07 c0 44 ac-22 7d c5 c5 4c 6c 43 17   ......D."}..LlC.
    0090 - be 29 c1 fd 5f 01 d4 f3-d3 8b 6d 60 dd ef 4f 9e   .)..._......m`..O.
    00a0 - 4d 46 0e b8 b7 af 0d 1f-82 a1 5e b9 63 6e cd 5b   MF........^.cn.[
    00b0 - f3 ae d5 52 5c d2 bd e0-33 a1 8a 64 b7 f3 b9 f9   ...R\...3..d....
    00c0 - ee e3 4a 56 a7 dd 0f 97-ad 4f 01 f9 e0 fc 31 11   ..JV.....O....1.
    00d0 - 2a 25 bb b5 f1 df df 55-29 00 45 e2 67 9e 0c 4c   *%.....U).E.g..L
    00e0 - e9 cc f2 a9 1e d0 44 b5-22 4f 63 f4               ......D."Oc.

    Start Time: 1613445306
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
220 smtp.gmail.com ESMTP d1sm19989360pgl.17 - gsmtp
helo gmail.com
250 smtp.gmail.com at your service
auth login
334 VXNlcm5hbWU6
bHVpc2VzcGlub3phMTEyNTk5QGdtYWlsLmNvbQ==
334 UGFzc3dvcmQ6
MTEyNTE5OTk=
235 2.7.0 Accepted
mail from: luisespinoza112599@gmail.com
555 5.5.2 Syntax error. d1sm19989360pgl.17 - gsmtp
mail from: <luisespinoza112599@gmail.com>
250 2.1.0 OK d1sm19989360pgl.17 - gsmtp
rcpt to: <lfespino@asu.edu>
250 2.1.5 OK d1sm19989360pgl.17 - gsmtp
data
354  Go ahead d1sm19989360pgl.17 - gsmtp
data
Subject: This is for SER 321
Isn't this cool? I don't know what I am doing!
.
250 2.0.0 OK  1613445735 d1sm19989360pgl.17 - gsmtp
quit
221 2.0.0 closing connection d1sm19989360pgl.17 - gsmtp
read:errno=0
louie-espin@Louie-Linux:~$
```

(no subject)

luisespinoza112599@gmail.com

data
Subject: This is for SER 321
Isn't this cool? I don't know what I am doing!

Tracking with WireShark:

File added to github repository.

command	frames	Description
openssl s_client −crlf −ign_ eof −connect smtp.gmail.com:465	1-11	This opens an SSL connection to smtp.gmail.com at port 465 and prints the SSL certificate.
helo gmail.com	12-15	initiate the SMTP conversation with the gmail client
auth login	16-19	authenticate to the server with login
bHVpc2VzcGlub3phMTEyNTk5QGdtYWlsLmNvbQ==	20-23	My username
MTEyNTE5OTk=	24-27	My password
mail from: <luisespinoza112599@gmail.com>	28-31	Email of the sender
rcpt to: <lfespino@asu.edu>	32-35	email of the reciever
data	36-39	Starts a new message transfer
Subject:	40-41	not a command
Email text	42-43	not a command
<enter>	44-45	not a command
.	46-49	Marks the end of the new message
Other email	50-71	not a command

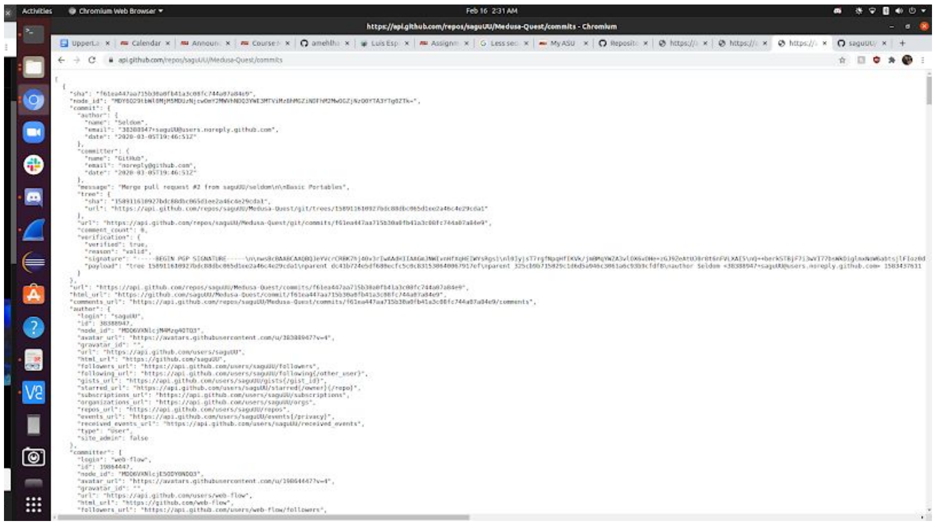
quit	71-80	Closes connection with server

Questions

1. **What filter did you use to catch the traffic and explain why you chose that filter?**
   I used the TCP filter at port 465 because we are using an SSL encryption protocol, which uses TCP. This displays both TCP and TLS protocols.
2. **What is the standard port for SMTP and why do we use port 465 in the example?**
   The standard port is port 25. We are using port 465 because we are using SSL
3. **Explain each line used in the command line and what does it do?**
   Answered in the table above.
4. **How much back and forth communication do you see for establishing the connection?**
   Since we are using TCP, establishing our connection will be done through many packets. Opening an SSL connection, initiating our conversation with gmail.com and authenticating was done in a total of 27 frames.
5. **What is the port of your local machine between sending the two emails when communicating with the SMTP server?**
   In my wireshark file, the port is 38932.
6. **Explain who sends the first FIN flag and how the quitting process works.**
   In my wireshark file, the source port 465 sends the FIN flag first. This happens because "quit" asks the server to close the connection, and it responds with 221 if it is closed correctly.

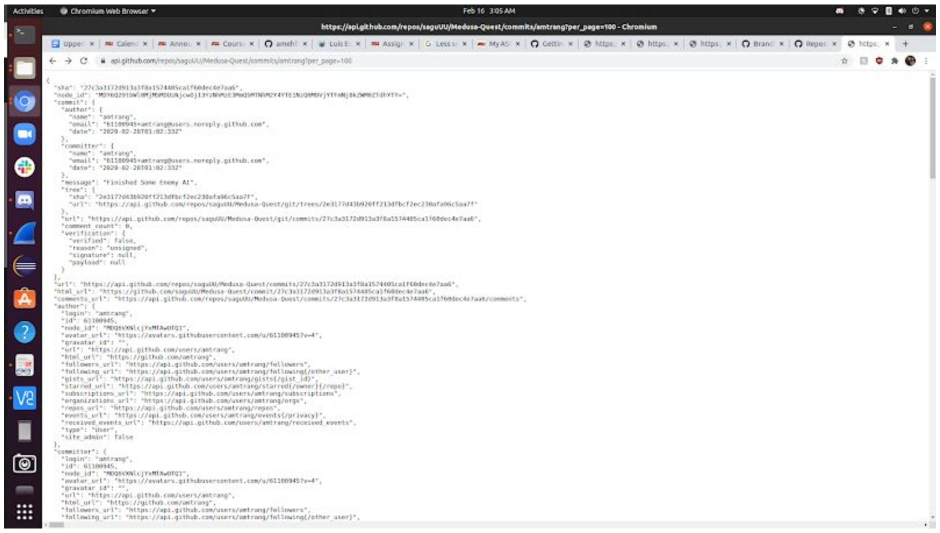
2. Understanding HTTP

Call that gets all the commits on the default branch of a repository
https://api.github.com/repos/saguUU/Medusa-Quest/commits


Activities
Chromium Web Browser
Feb 16 2:31 AM
https://api.github.com/repos/saguUU/Medusa-Quest/commits - Chromium
api.github.com/repos/saguUU/Medusa-Quest/commits

```
[
  {
    "sha": "f61ea447aa715b38a0fb41a3c08fc744a07a84e9",
    "node_id": "MDY6Q29tbWl0MjM5MDUzNjcwOmY2MWVhNDQ3YWE3MTViMzBhMGZiNDFhM2MwOGZjNzQ0YTA3YTg0ZTk=",
    "commit": {
      "author": {
        "name": "Seldom",
        "email": "38388947+saguUU@users.noreply.github.com",
        "date": "2020-03-05T19:46:51Z"
      },
      "committer": {
        "name": "GitHub",
        "email": "noreply@github.com",
        "date": "2020-03-05T19:46:51Z"
      },
      "message": "Merge pull request #2 from saguUU/seldom\n\nBasic Portables",
      "tree": {
        "sha": "15891161892 7bdc88dbc065d1ee2a46c4e29cda1",
        "url": "https://api.github.com/repos/saguUU/Medusa-Quest/git/trees/158911618927bdc88dbc065d1ee2a46c4e29cda1"
      },
      "url": "https://api.github.com/repos/saguUU/Medusa-Quest/git/commits/f61ea447aa715b30a0fb41a3c08fc744a07a84e9",
      "comment_count": 0,
      "verification": {
        "verified": true,
        "reason": "valid",
        "signature": "-----BEGIN PGP SIGNATURE-----\n\nwsBcBAABCAAQBQJeYVcrCRBK7hj40x3rIwAAdHIIAAGaJNWIvnHfXgHEIWYsRgsl\nl0JyjsT7rgfNpqHfIKVk/jWBMqYWZA3vlOX6vOHe+zGJ9ZeAtU38r0t6nFVLXAIS\nQ++berkSTBjF7J3wVI77bsWKDiglmxNmW6abtsjlFIoz0d",
        "payload": "tree 158911618927bdc88dbc065d1ee2a46c4e29cda1\nparent dc41b724e5df680ecfc5c0c83153864008 7917ef\nparent 325cb0b715829c1d6d5a946c3061a6c93b3cfdf8\nauthor Seldom <38388947+saguUU@users.noreply.github.com> 1583437611"
      }
    },
    "url": "https://api.github.com/repos/saguUU/Medusa-Quest/commits/f61ea447aa715b30a0fb41a3c08fc744a07a84e9",
    "html_url": "https://github.com/saguUU/Medusa-Quest/commit/f61ea447aa715b30a0fb41a3c08fc744a07a84e9",
    "comments_url": "https://api.github.com/repos/saguUU/Medusa-Quest/commits/f61ea447aa715b30a0fb41a3c08fc744a07a84e9/comments",
    "author": {
      "login": "saguUU",
      "id": 38388947,
      "node_id": "MDQ6VXNlcjM4Mzg4OTQ3",
      "avatar_url": "https://avatars.githubusercontent.com/u/38388947?v=4",
      "gravatar_id": "",
      "url": "https://api.github.com/users/saguUU",
      "html_url": "https://github.com/saguUU",
      "followers_url": "https://api.github.com/users/saguUU/followers",
      "following_url": "https://api.github.com/users/saguUU/following{/other_user}",
      "gists_url": "https://api.github.com/users/saguUU/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/saguUU/starred{/owner}{/repo}",
      "subscriptions_url": "https://api.github.com/users/saguUU/subscriptions",
      "organizations_url": "https://api.github.com/users/saguUU/orgs",
      "repos_url": "https://api.github.com/users/saguUU/repos",
      "events_url": "https://api.github.com/users/saguUU/events{/privacy}",
      "received_events_url": "https://api.github.com/users/saguUU/received_events",
      "type": "User",
      "site_admin": false
    },
    "committer": {
      "login": "web-flow",
      "id": 19864447,
      "node_id": "MDQ6VXNlcjE5ODY0NDQ3",
      "avatar_url": "https://avatars.githubusercontent.com/u/19864447?v=4",
      "gravatar_id": "",
      "url": "https://api.github.com/users/web-flow",
      "html_url": "https://github.com/web-flow",
      "followers_url": "https://api.github.com/users/web-flow/followers",
```

Call that that adds GET parameters that specifies a specific branch and sets page limit to 100
https://api.github.com/repos/saguUU/Medusa-Quest/commits/amtrang?per_page=100



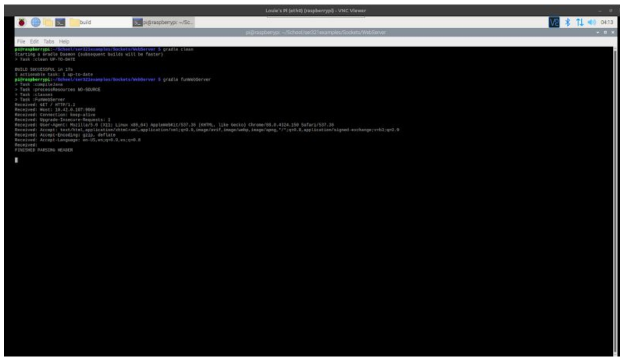
```
{
  "sha": "27c3a3172d913a3f8a1574405ca1f60dec4e7aa6",
  "node_id": "[illegible]",
  "commit": {
    "author": {
      "name": "amtrang",
      "email": "61100945+amtrang@users.noreply.github.com",
      "date": "2020-02-28T01:02:33Z"
    },
    "committer": {
      "name": "amtrang",
      "email": "61100945+amtrang@users.noreply.github.com",
      "date": "2020-02-28T01:02:33Z"
    },
    "message": "Finished Some Enemy AI",
    "tree": {
      "sha": "2e3177d43b920ff213dfbcf2ec230afa96c5aa7f",
      "url": "https://api.github.com/repos/saguUU/Medusa-Quest/git/trees/2e3177d43b920ff213dfbcf2ec230afa96c5aa7f"
    },
    "url": "https://api.github.com/repos/saguUU/Medusa-Quest/git/commits/27c3a3172d913a3f8a1574405ca1f60dec4e7aa6",
    "comment_count": 0,
    "verification": {
      "verified": false,
      "reason": "unsigned",
      "signature": null,
      "payload": null
    }
  },
  "url": "https://api.github.com/repos/saguUU/Medusa-Quest/commits/27c3a3172d913a3f8a1574405ca1f60dec4e7aa6",
  "html_url": "https://github.com/saguUU/Medusa-Quest/commit/27c3a3172d913a3f8a1574405ca1f60dec4e7aa6",
  "comments_url": "https://api.github.com/repos/saguUU/Medusa-Quest/commits/27c3a3172d913a3f8a1574405ca1f60dec4e7aa6/comments",
  "author": {
    "login": "amtrang",
    "id": 61100945,
    "node_id": "MDQ6VXNlcjYxMTAwOTQ1",
    "avatar_url": "https://avatars.githubusercontent.com/u/61100945?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amtrang",
    "html_url": "https://github.com/amtrang",
    "followers_url": "https://api.github.com/users/amtrang/followers",
    "following_url": "https://api.github.com/users/amtrang/following{/other_user}",
    "gists_url": "https://api.github.com/users/amtrang/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/amtrang/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/amtrang/subscriptions",
    "organizations_url": "https://api.github.com/users/amtrang/orgs",
    "repos_url": "https://api.github.com/users/amtrang/repos",
    "events_url": "https://api.github.com/users/amtrang/events{/privacy}",
    "received_events_url": "https://api.github.com/users/amtrang/received_events",
    "type": "User",
    "site_admin": false
  },
  "committer": {
    "login": "amtrang",
    "id": 61100945,
    "node_id": "MDQ6VXNlcjYxMTAwOTQ1",
    "avatar_url": "https://avatars.githubusercontent.com/u/61100945?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amtrang",
    "html_url": "https://github.com/amtrang",
    "followers_url": "https://api.github.com/users/amtrang/followers",
    "following_url": "https://api.github.com/users/amtrang/following{/other_user}",
```

Questions

1. **Explain the specific api calls you used.**
   I used two REST API calls from github. The first call specified the repository name and its owner, then I used "/commits" to display all the commits of the repository's default branch. The next api call was similar to the last one, except I added "/amtrang" after the commits. This gave me the commits of a specific branch, called amtrang. Next I added "?per_page=100" which makes the call display 100 up to commits per page.
2. **Explain the difference between stateless and state-full communication**
   State-full communication is different from stateless communication in that it stores previous context and history and uses that to deliver data. In a stateless communication, requests and responses happen in isolation from any previous history or state, what you ask for is what you get.

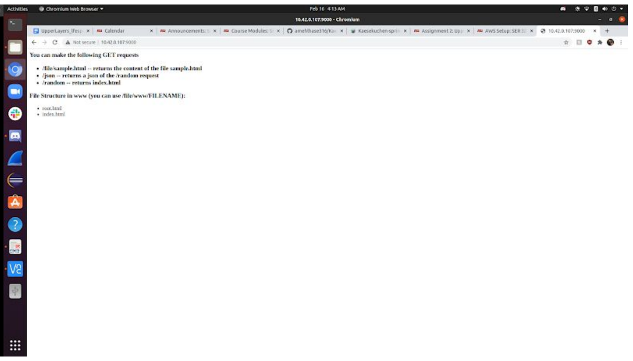
3. Setup your second system and run Server on it

Running a simple Java WebServer:



```
pi@raspberrypi:~/School/ser321examples/Sockets/WebServer $ gradle clean
Starting a Gradle Daemon (subsequent builds will be faster)
> Task :clean UP-TO-DATE

BUILD SUCCESSFUL in 17s
1 actionable task: 1 up-to-date
pi@raspberrypi:~/School/ser321examples/Sockets/WebServer $ gradle FunWebServer
> Task :compileJava
> Task :processResources NO-SOURCE
> Task :classes
> Task :FunWebServer
Received: GET / HTTP/1.1
Received: Host: 10.42.0.107:9000
Received: Connection: keep-alive
Received: Upgrade-Insecure-Requests: 1
Received: User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 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=0.9
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9,es;q=0.8
Received:
FINISHED PARSING HEADER
```



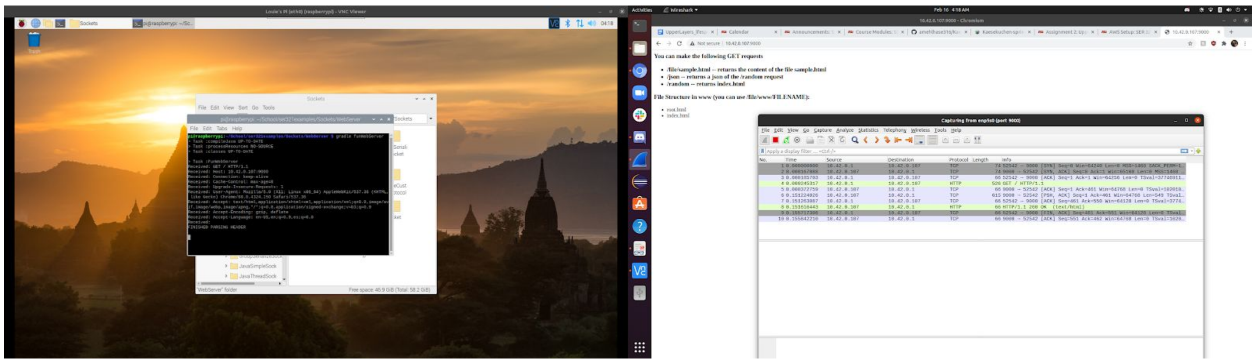
You can make the following GET requests

- /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):

- root.html
- index.html

Analyze what happens



```
pi@raspberrypi:~/School/ser321examples/Sockets/WebServer $ gradle FunWebServer
> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :FunWebServer
Received: GET / HTTP/1.1
Received: Host: 10.42.0.107:9000
Received: Connection: keep-alive
Received: Cache-Control: max-age=0
Received: Upgrade-Insecure-Requests: 1
Received: User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/88.0.4324.150 Safari/537.36
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=0.9
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9,es;q=0.8
Received:
FINISHED PARSING HEADER
```

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):

- root.html
- index.html

Capturing from enp5s0 (port 9000)

```
No.  Time         Source       Destination  Protocol Length Info
   1 0.000000000  10.42.0.1    10.42.0.107  TCP        74 52542 → 9000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1…
   2 0.000167888  10.42.0.107  10.42.0.1    TCP        74 9000 → 52542 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 …
   3 0.000185703  10.42.0.1    10.42.0.107  TCP        66 52542 → 9000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=37746911…
   4 0.000245317  10.42.0.1    10.42.0.107  HTTP      526 GET / HTTP/1.1
   5 0.000372759  10.42.0.107  10.42.0.1    TCP        66 9000 → 52542 [ACK] Seq=1 Ack=461 Win=64768 Len=0 TSval=102010…
   6 0.151224826  10.42.0.107  10.42.0.1    TCP       615 9000 → 52542 [PSH, ACK] Seq=1 Ack=461 Win=64768 Len=549 TSval…
   7 0.151263087  10.42.0.1    10.42.0.107  TCP        66 52542 → 9000 [ACK] Seq=461 Ack=550 Win=64128 Len=0 TSval=3774…
   8 0.151616443  10.42.0.107  10.42.0.1    HTTP       66 HTTP/1.1 200 OK  (text/html)
   9 0.155717306  10.42.0.1    10.42.0.107  TCP        66 52542 → 9000 [FIN, ACK] Seq=461 Ack=551 Win=64128 Len=0 TSval…
  10 0.155842210  10.42.0.107  10.42.0.1    TCP        66 9000 → 52542 [ACK] Seq=551 Ack=462 Win=64768 Len=0 TSval=1020…
```

Questions

1. **What filter did you use? Explain why you chose that filter.**
   I simply chose to use “port 9000” since that is the port I am using and my second system is a pi that is connected to my computer through an ethernet cable.
2. **What happens when you are on /random and click the refresh button compared to the browser refresh (you can also use the command line output that the WebServer generates to answer this)?**
   When pressing the “random” button I get the following command line output:

```
Received: GET /json HTTP/1.1
Received: Host: 10.42.0.107:9000
Received: Connection: keep-alive
Received: User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4324.150 Safari/537.36
Received: Accept: */*
Received: Referer: http://10.42.0.107:9000/random
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9,es;q=0.8
Received:
FINISHED PARSING HEADER
```

   When reloading in the browser I get this instead:

```
Received: GET /random HTTP/1.1
Received: Host: 10.42.0.107:9000
Received: Connection: keep-alive
Received: Cache-Control: max-age=0
Received: Upgrade-Insecure-Requests: 1
Received: User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4324.150 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-e
xchange;v=b3;q=0.9
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9,es;q=0.8
Received:
FINISHED PARSING HEADER

Received: GET /json HTTP/1.1
Received: Host: 10.42.0.107:9000
Received: Connection: keep-alive
Received: User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4324.150 Safari/537.36
Received: Accept: */*
Received: Referer: http://10.42.0.107:9000/random
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9,es;q=0.8
```

```
Received:
FINISHED PARSING HEADER
```

3. **What kinds of response codes are you able to get through different requests to your server?**
   When looking through the code, we can see the following codes:
   - 200: OK
   - 404: Not Found
   - 400: Bad Request
4. **Explain the response codes you get and why you get them?**
   - 200: When everything goes well
   - 404: When we make a file GET request and the file does not exist
   - 400: When the request is not recognized at all
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, because this is HTTP and the information is not encrypted.
6. **Based on the above question explain why HTTPs is now more common than HTTP.**
   With all the sensitive information that is transferred, we need the encryption that HTTPs provides us.
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 listens to port 9000. The most common port for HTTP 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?**
   35712. The traffic differs in that my SMTP server's traffic was encrypted.

Creating a Real Web Server

1. **Check your traffic to your Webserver now. What port is the traffic going to now? Is it the same as previously used or is it and should it be different?**
   It is now going to port 80, the default HTTP port. It is different because we are using nginx to display our webserver program.
2. **Is it still HTTP or is it now HTTPs? Why?**
   It is still HTTP. We get to set up HTTPs in the extra credit section.

Multiply

The multiply case did not have error handling. I decided to first check if the request contains "num1=" and "num2=" then check if the parameters are integers. This my code, using code 400 as Bad Request:

```
else if (request.contains("multiply?")) {
        // This multiplies two numbers, there is NO error handling, so when
        // wrong data is given this just crashes
// checks if there are mun1 and num2 parameters
if (request.contains("num1=") && request.contains("num2=")) {
  Map<String, String> query_pairs = new LinkedHashMap<String, String>();
  // extract path parameters
  query_pairs = splitQuery(request.replace("multiply?", ""));

  // check if parameters are integers
  if (isStringInt(query_pairs.get("num1")) && isStringInt(query_pairs.get("num2"))) {

    // extract required fields from parameters
    Integer num1 = Integer.parseInt(query_pairs.get("num1"));
    Integer num2 = Integer.parseInt(query_pairs.get("num2"));

    // do math
    Integer result = num1 * num2;

    // Generate response
    builder.append("HTTP/1.1 200 OK\n");
    builder.append("Content-Type: text/html; charset=utf-8\n");
    builder.append("\n");
    builder.append("Result is: " + result);
  } else { // failure; the request does not contain two integers
    builder.append("HTTP/1.1 400 Bad Request\n");
    builder.append("Content-Type: text/html; charset=utf-8\n");
    builder.append("\n");
    builder.append("failure; the request does not contain two integers");
  }

} else { // failure; the request does not contain two paramters
  builder.append("HTTP/1.1 400 Bad Request\n");
  builder.append("Content-Type: text/html; charset=utf-8\n");
  builder.append("\n");
  builder.append("failure; the request does not contain two paramters");
}
```

GitHub

WebServer updated coded added to repo.