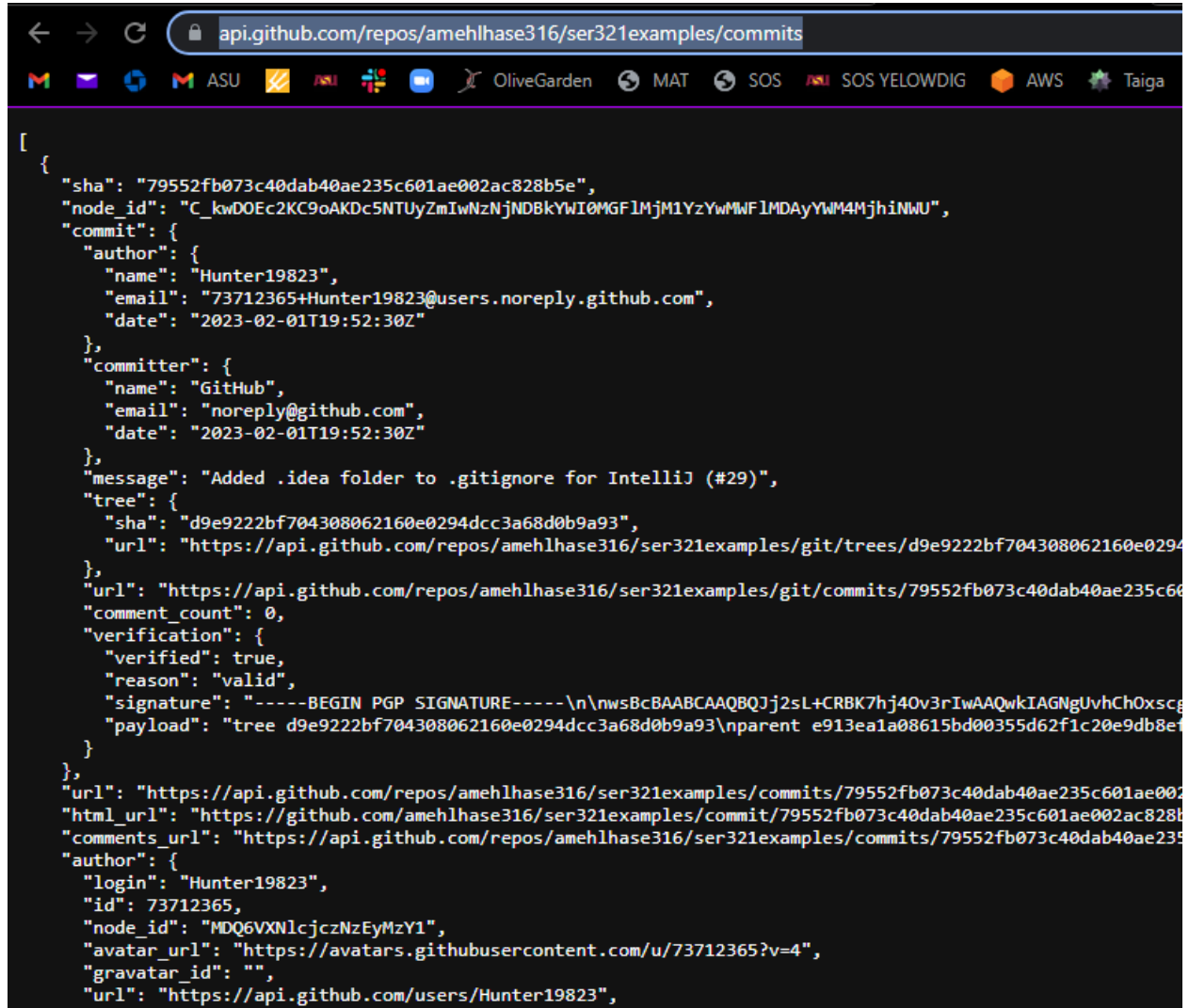


<https://docs.github.com/en/rest/search?apiVersion=2022-11-28#search-commits>  
<https://github.com/amehlhase316/ser321examples/tree/READMEupdates>  
<https://api.github.com/repos/amehlhase316/ser321examples/commits/READMEupdates>

Task1:

1. <https://api.github.com/repos/amehlhase316/ser321examples/commits>



```
[
  {
    "sha": "79552fb073c40dab40ae235c601ae002ac828b5e",
    "node_id": "C_kwDOEc2KC9oAKDc5NTUyZmIwNzNjNDhkYWl0MGFIMjM1YzYwMWFIMDAyYWM4MjhiNWU",
    "commit": {
      "author": {
        "name": "Hunter19823",
        "email": "73712365+Hunter19823@users.noreply.github.com",
        "date": "2023-02-01T19:52:30Z"
      },
      "committer": {
        "name": "GitHub",
        "email": "noreply@github.com",
        "date": "2023-02-01T19:52:30Z"
      },
      "message": "Added .idea folder to .gitignore for IntelliJ (#29)",
      "tree": {
        "sha": "d9e9222bf704308062160e0294dcc3a68d0b9a93",
        "url": "https://api.github.com/repos/amehlhase316/ser321examples/git/trees/d9e9222bf704308062160e0294dcc3a68d0b9a93"
      },
      "url": "https://api.github.com/repos/amehlhase316/ser321examples/git/commits/79552fb073c40dab40ae235c601ae002ac828b5e",
      "comment_count": 0,
      "verification": {
        "verified": true,
        "reason": "valid",
        "signature": "-----BEGIN PGP SIGNATURE-----\n\nnwsBcBAABCAAQBQJj2sL+CRBK7hj40v3rIwAAQwkIAGNgUvhChOxscg\n\npayload: \"tree d9e9222bf704308062160e0294dcc3a68d0b9a93\\nparent e913ea1a08615bd00355d62f1c20e9db8e1\"",
        "payload": "tree d9e9222bf704308062160e0294dcc3a68d0b9a93\\nparent e913ea1a08615bd00355d62f1c20e9db8e1"
      }
    },
    "url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/79552fb073c40dab40ae235c601ae002ac828b5e",
    "html_url": "https://github.com/amehlhase316/ser321examples/commit/79552fb073c40dab40ae235c601ae002ac828b5e",
    "comments_url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/79552fb073c40dab40ae235c601ae002ac828b5e/comments",
    "author": {
      "login": "Hunter19823",
      "id": 73712365,
      "node_id": "MDQ6VXNlcjczNzEyMzY1",
      "avatar_url": "https://avatars.githubusercontent.com/u/73712365?v=4",
      "gravatar_id": "",
      "url": "https://api.github.com/users/Hunter19823",
      "html_url": "https://github.com/Hunter19823"
    }
  }
]
```

[https://api.github.com/repos/amehlhase316/ser321examples/commits/READMEupdates/search?per\\_page=10](https://api.github.com/repos/amehlhase316/ser321examples/commits/READMEupdates/search?per_page=10)

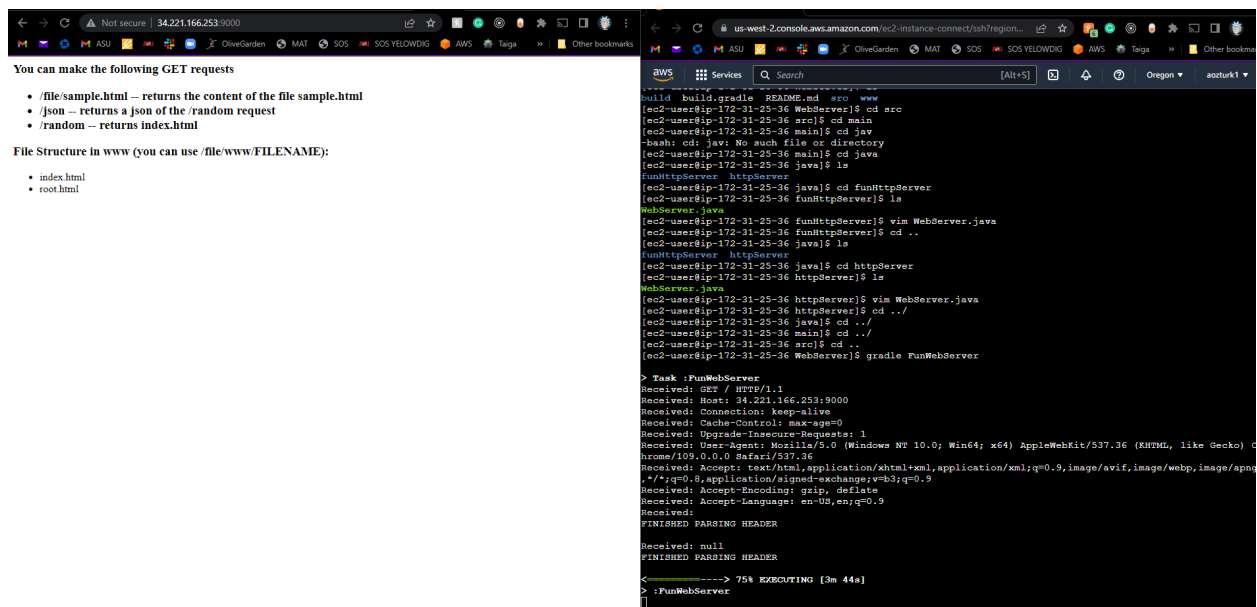
```

< → ↻ 🔒 api.github.com/repos/amehlhase316/ser321examples/commits/READMEupdates
📧 📬 🌐 ASU ☑️ 🏠 OliveGarden 🌍 MAT ⚙️ SOS 🏠 SOS YELOWDIG 🟡 AWS ✨ Taiga 🔄 GitHub

{
  "reason": "unsigned",
  "signature": null,
  "payload": null
},
"url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/a248cf5371226a401be4cf3aee78a4eeac7ebadd",
"html_url": "https://github.com/amehlhase316/ser321examples/commit/a248cf5371226a401be4cf3aee78a4eeac7ebadd",
"comments_url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/a248cf5371226a401be4cf3aee78a4eeac7ebadd/comments",
"author": null,
"committer": null,
"parents": [
  {
    "sha": "76fcc9249662abde309ea0d361f8493ad8bfca8",
    "url": "https://api.github.com/repos/amehlhase316/ser321examples/commits/76fcc9249662abde309ea0d361f8493ad8bfca8",
    "html_url": "https://github.com/amehlhase316/ser321examples/commit/76fcc9249662abde309ea0d361f8493ad8bfca8"
  }
],
"stats": {
  "total": 518,
  "additions": 51,
  "deletions": 467
},
"files": [
  {
    "sha": "8397f23736ada23cd03d741d3cb283cb283663c1",
    "filename": "Gradle/JavaGradle/build.gradle",
    "status": "modified",
    "additions": 1,
    "deletions": 1,
    "changes": 2,
    "blob_url": "https://github.com/amehlhase316/ser321examples/blob/a248cf5371226a401be4cf3aee78a4eeac7ebadd/Gradle%2FJavaGradle%2Fbuild.gradle",
    "raw_url": "https://github.com/amehlhase316/ser321examples/raw/a248cf5371226a401be4cf3aee78a4eeac7ebadd/Gradle%2FJavaGradle%2Fbuild.gradle",
    "contents_url": "https://api.github.com/repos/amehlhase316/ser321examples/contents/Gradle%2FJavaGradle%2Fbuild.gradle?ref=a248cf5371226a401be4cf3aee78a4eeac7ebadd",
    "patch": "@@ -54,7 +54,7 @@ task runFraction(type: JavaExec) {\n // Try:\n // 1. Change Fraction.main so it accepts\n}"
  },
  {
    "sha": "22c6980fc165b9a5c9c7a9f271b44feae4013ab3",
    "filename": "Middleware/G-RPC/README.md",
    "status": "modified",
    "additions": 3,
    "deletions": 0,
    "changes": 3,
    "blob_url": "https://github.com/amehlhase316/ser321examples/blob/a248cf5371226a401be4cf3aee78a4eeac7ebadd/Middleware%2FGRPC%2FREADME.md",
    "raw_url": "https://github.com/amehlhase316/ser321examples/raw/a248cf5371226a401be4cf3aee78a4eeac7ebadd/Middleware%2FGRPC%2FREADME.md",
    "contents_url": "https://api.github.com/repos/amehlhase316/ser321examples/contents/Middleware%2FGRPC%2FREADME.md?ref=a248cf5371226a401be4cf3aee78a4eeac7ebadd",
    "patch": "@@ 10,6 +10,9 @@ You do not need to install anything things will run through the gradle file.\n \\\n -"
  }
]

```

## Task2:



1. I used the filter `tcp.port==9000`
2. Random will give you a new json with the image header but the refresh gives you HTML code with the json and jpg as well.
3. 200 with the /random, 400 with github?, 404 with file/
4. 200 is if it works ok, 400 for when request not recognized, 404 for when request not found
5. Yes you can see some in plain English text
6. Because you can see what people are sending on the internet so HTTPs doesn't allow for that.
7. We are listening to 9000. The most common port for HTTP 80
8. Local port 11006, 11007 and up in range about 100

## 2.3

1. `ipOfSecondMachine:80`
2. The traffic now is going through port 80 rather than 9000. I don't think it should be the same as it is supposed to direct everything from 9000 to 80
3. Still HTTP. Because we can see the request in plain English text

[illegible]

I tried to do the extra credit multiple times following the duckdns instructions but just didnt' work

first step - choose a domain.

linux cron

if your linux install is running a cronstab, then you can use a cron job to keep updated we can see this with

```
ps -ef | grep cr[on]
```

if this returns nothing - then go and read up how to install cron for your distribution of linux also confirm that you have curl installed, test this by attempting to run curl

```
curl
```

if this returns a command not found like error - then find out how to install curl for your distribution. otherwise lets get started and make a directory to put your files in, move into it and make our main script

```
mkdir duckdns
cd duckdns
vi duck.sh
```

now copy this text and put it into the file (in vi you hit the i key to insert, ESC then u to undo) The example below is for the domain aozturk

if you want the configuration for a different domain, use the drop down box above

you can pass a comma separated (no spaces) list of domains

you can if you need to hard code an IP (best not to - leave it blank and we detect your remote ip)

hit ESC then use arrow keys to move the cursor x deletes, i puts you back into insert mode

```
echo url="https://www.duckdns.org/update?domains=aozturk&token=ec79aff3-a3eb-4d54-919a-d16252e819a3&ip" | curl -k -o ->/duckdns/duck.log -K -
```

now save the file (in vi hit ESC then :wq! then ENTER)

this script will make a https request and log the output in the file duck log

now make the duck.sh file executable

```
chmod 700 duck.sh
```

next we will be using the cron process to make the script get run every 5 minutes

```
crontab -e
```

aws

Services

Search

[Alt+T]

Go to Settings or a Terms

http:// aozturk .duckdns.org

[ec2-user@ip-172-31-25-36 ~]\$ crontab -e

crontab: installing new crontab

[ec2-user@ip-172-31-25-36 ~]\$ crontab -e

[1]+ Stopped crontab -e

You have new mail in /var/spool/mail/ec2-user

[ec2-user@ip-172-31-25-36 ~]\$ crontab -e

crontab: installing new crontab

[ec2-user@ip-172-31-25-36 ~]\$ ./duck.sh

% Total % Received % Xferd Average Speed Time Time Time Current

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the file /home/ec2-user/duckdns/duck.log: No such

Warning: file or directory

100 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

curl: (23) Failure writing output to destination

[ec2-user@ip-172-31-25-36 ~]\$ cat duck.log

cat: duck.log: No such file or directory

[ec2-user@ip-172-31-25-36 ~]\$ sudo ./duck.sh

% Total % Received % Xferd Average Speed Time Time Time Current

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the file /root/duckdns/duck.log: No such file or

Warning: directory

100 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

curl: (23) Failure writing output to destination

[ec2-user@ip-172-31-25-36 ~]\$ echo url="https://www.duckdns.org/update?domains=aozturk&token=ec79aff3-a3eb-4d54-919a-d16252e819a3&ip" | curl -k -o ->/duckdns/duck.log -K -

% Total % Received % Xferd Average Speed Time Time Time Current

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the file /home/ec2-user/duckdns/duck.log: No such

Warning: file or directory

100 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

curl: (23) Failure writing output to destination

[ec2-user@ip-172-31-25-36 ~]\$ chmod 700 duck.sh

[ec2-user@ip-172-31-25-36 ~]\$ echo url="https://www.duckdns.org/update?domains=aozturk&token=ec79aff3-a3eb-4d54-919a-d16252e819a3&ip" | curl -k -o ->/duckdns/duck.log -K -

% Total % Received % Xferd Average Speed Time Time Time Current

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

the file /home/ec2-user/duckdns/duck.log: No such

Warning: file or directory

100 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

curl: (23) Failure writing output to destination

[ec2-user@ip-172-31-25-36 ~]\$ []

Feedback

Language

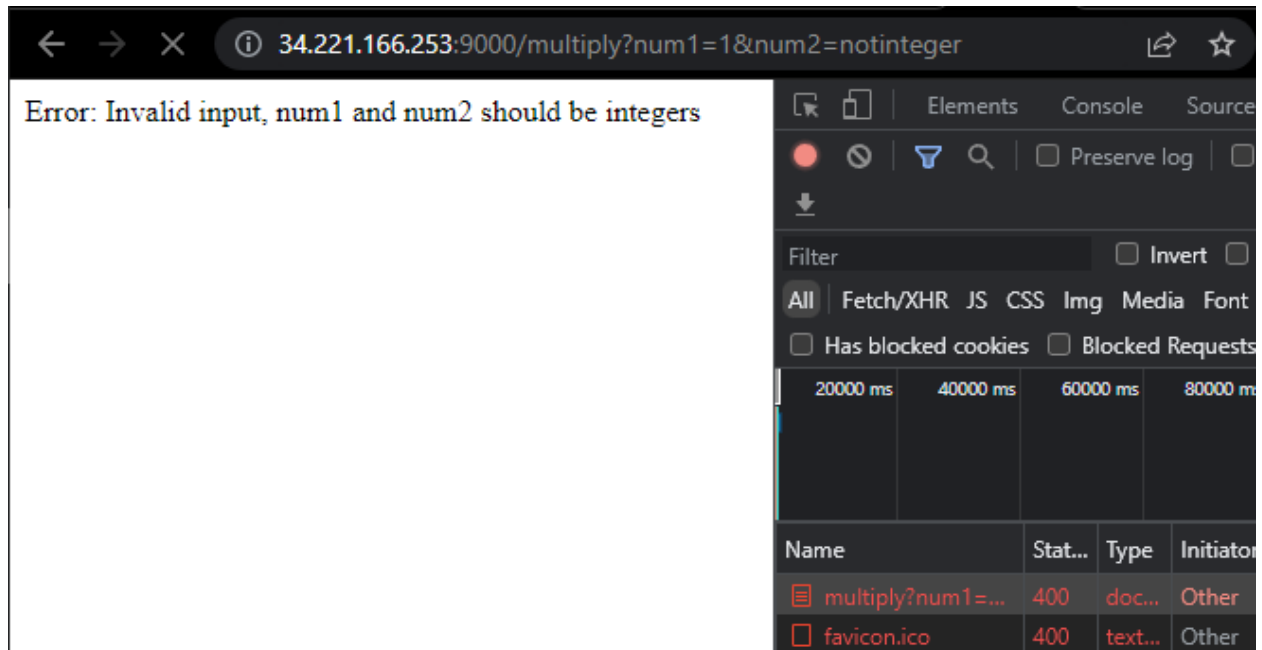
Activate Windows

Go to Settings to activate Windows.

Privacy

Terms

### 2.5.1 Multiply



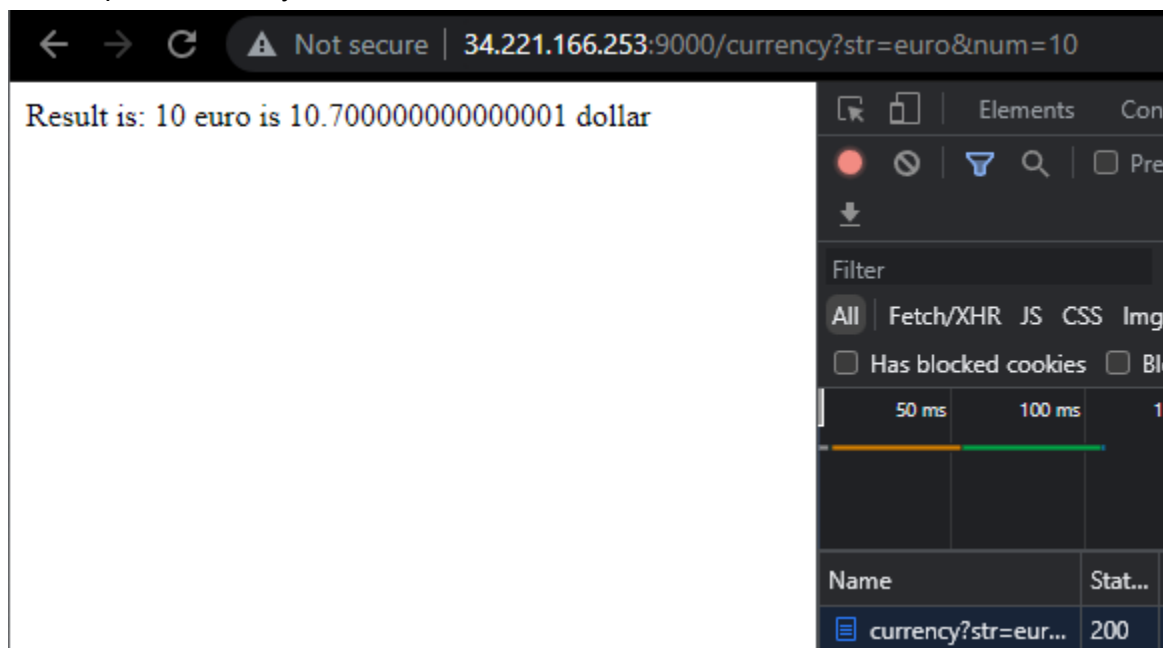
I used the `NumberFormatException` in a try and catch for the `num1` and `num2` instantiation which throws a 400 for an invalid request because not putting an integer is the wrong request format/attempt with "multiply?".

## 2.5.2 Github

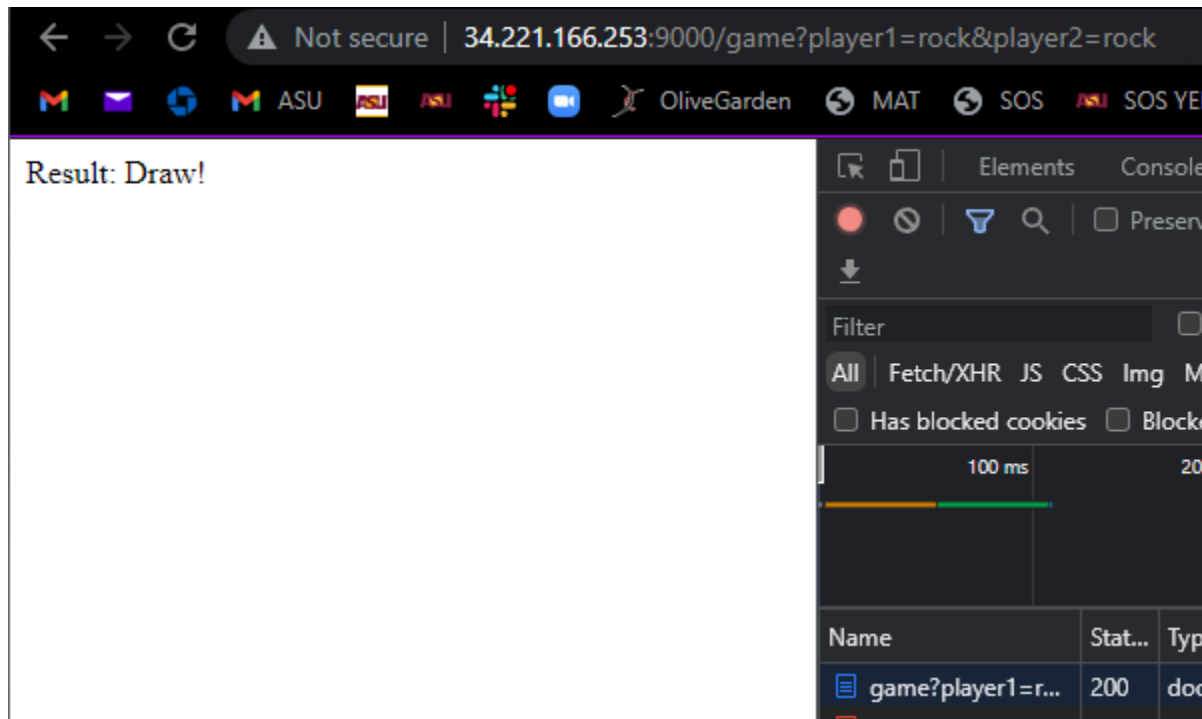
...

## 2.5.3

First request: currency?



Second request: /game?player1=rock&player2=rock



2.5.4

**First request:**

/currency?str=dollar&num=10

Path:

currency

Body Parameter:

str (String) - required //dollar or euro

num (int) - required //any whole number

Response:

OK 200 - Success

"Result is: " + num + " " + str + " is " + result + " " + strng

400 - Bad Request

Error: Invalid parameters. str and num are required.

401 - Bad Request

Error: Invalid currency type. str must be either 'dollar' or 'euro'.

**Second request:**

/game?player1=rock&player2=rock

Path:

game

Body Parameter:

player1 (String) - rock/paper/scissor

player2 (String) - rock/paper/scissor

Response:

OK 200 - Success

"Player 1 wins!" or "Player 2 wins!"

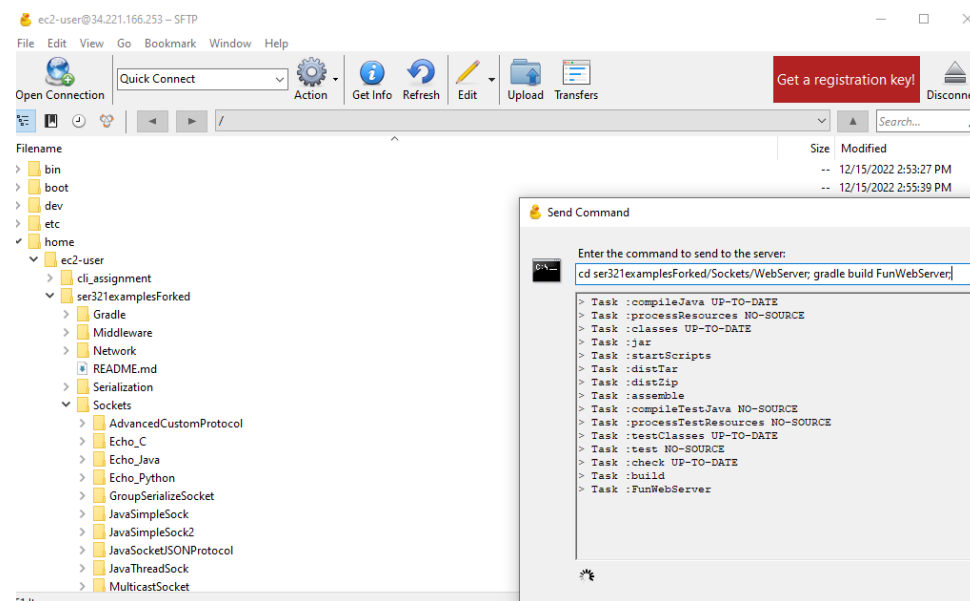
400 - Bad Request

Error: player1 must be either 'rock', 'paper', or 'scissors'.

401 - Bad Request

Error: Invalid parameters. player1 and player2 are required.

## 2.5.5



Running through Cyberduck without the AWS