Saad S Khan

Professor Mehlhase

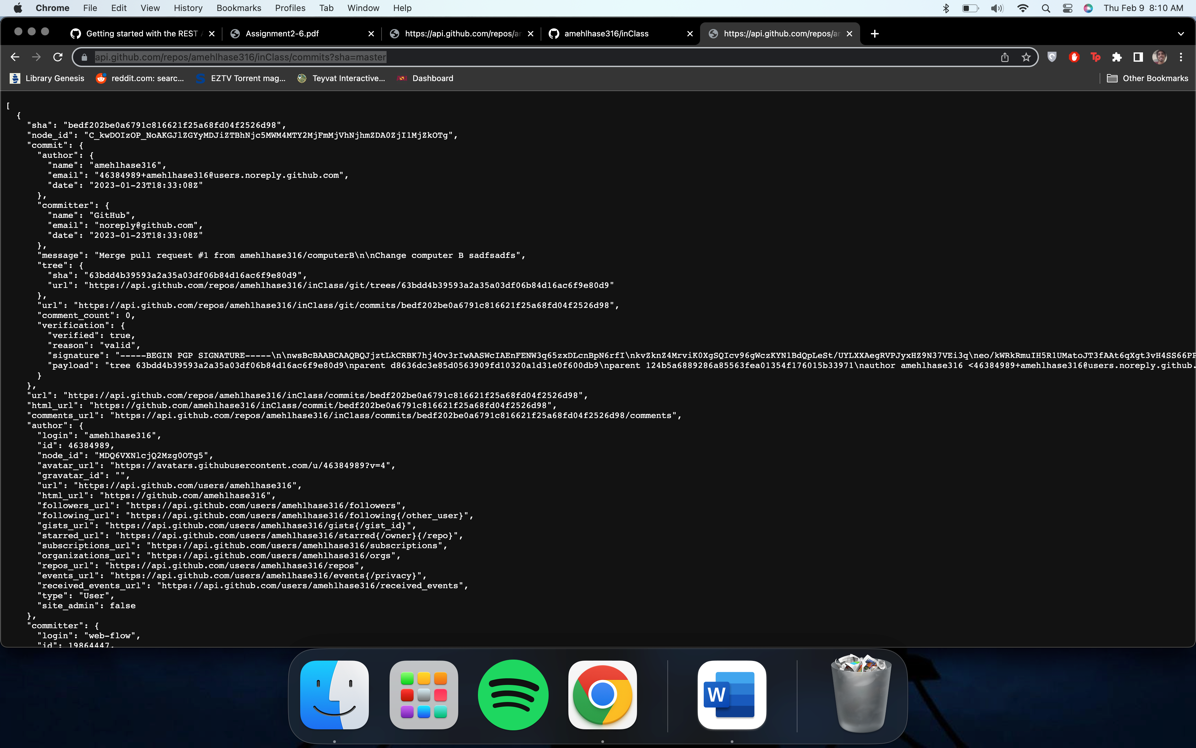
January 29th 2023

SER321

Assignment 2

Task 1:

URL for default branch: <https://api.github.com/repos/amehlhase316/inClass/commits?sha=master>



URL with parameters: <https://api.github.com/repos/amehlhase316/inClass/commits?sha=GitHubBranch&per_page=50>

A screenshot of a computer screen

Description automatically generated

Task 2:

Graphical user interface, text, application, chat or text message

Description automatically generated

1. I used the ip address filter to only see activity for the connection between my PC and the aws server the program is running on.
2. The browser sends an HTTP request to the server asking for a new random page when you are on the "/random" page and click the "Random" button. The browser refreshes the page by making another HTTP request to the server, which causes the current page to reload. These requests will appear in the command line output from the server.
3. We can get three response codes. The 200,400 and 404 codes.
4. The codes you can get are 200 ok code signaling the execution was successful, the 400 bad request code when request is not recognized and the 404 not found code when the file being called isn’t found.
5. Yes I can.
6. HTTPs encrypts your data so anyone tracking traffic cant gain access to your sensitive data while your PC is communicating with a website.
7. The server listens to port 9000, which isn’t the most common port for HTTP because the default HTTP port is 80.
8. Port 53789

Task 2.3

1. <http://35.88.123.125/>
2. The port being connected to is port 80. The ports changed because before we were specifying the port to connect to but now, we aren’t so it connected to the default http port, 80, and is then rerouted to port 9000.
3. It is still http because we did not change the protocol, we just changed the port its connecting to.
4. Graphical user interface, application

   Description automatically generated

Task 2.5.1:

I chose to put in error handling for when both the inputs aren’t provided and when one of the inputs is invalid. With this error handling in place, the server will be able to handle invalid or missing data provided by the client in a more graceful way, without crashing and while providing meaningful responses to the client.

Task 2.5.4:

Method 1: convert

Request: <http://localhost:9000/convert?value=15&from=miles&to=km>

Path: convert

Body parameter: from (string)

To (string)

val (double)

Response:

Ok (200)

Bad Request (400)

Internal Server Error (500)

Not Found (404)

Output: Conversion Result + value + “ “ + from + “ is equal to “ + result + ” “ + to

This method takes a number input along with two string inputs. The number input is the value to be converted while the two string inputs tell the program if the value needs to be converted from miles to km or km to miles.

Method 2: multiply squares

Request: http://localhost:9000/multiply\_squares?num1=2&num2=3

Path: multiply\_squares

Body parameter:

Num1 (int)

Num2 (int)

Response:

Ok (200)

Bad Request (400)

Internal Server Error (500)

Not Found (404)

Output: The closest perfect square to the sum of squares of " + num1 + " and " + num2 + " is: " + closestPerfectSquare + "."

This method takes in two integers as arguments, squares them, adds the squares and then finds the closest possible perfect square.