

Fullstack QAP 2 – February 14th, 2023

Completed by: Rob Vatcher

Git Hub Repo link: <https://github.com/RobV34/fullStack2/blob/main/server.js>

1. How many hours did it take you to complete this assessment?

10-12 hours

2. What online resources you have used? (My lectures, YouTube, Stack overflow etc.)

Lectures, YouTube, and google.

3. Did you need to ask any of your friends in solving the problems.

I did not.

4. Did you need to ask questions to any of your instructors? If so, how many questions did you ask (or how many help sessions did you require)?

I did not.

5. Rate (subjectively) the difficulty of each question from your own perspective, and whether you feel confident that you can solve a similar but different problem requiring some of the same techniques in the future now that you've completed this one.

It was hard to get my head around at first but it all came together as I dug into it.

Task 3:

Step 2: Description of the events scenario.

The server utilizes a custom EventEmitter class, MyEmitter, to handle asynchronous events. The event scenarios captured and logged in the server are as follows:

HTTP Status Codes:

200 OK: Every time an HTML file is successfully served to the client, an event is emitted with a 200 OK status code.

404 Not Found: When a requested file is not found in the server's views directory, a 404 Not Found event is emitted.

500 Internal Server Error: Server errors, such as filesystem read errors not related to missing files, emit a 500 Internal Server Error event.

An event is emitted every time a specific route is accessed.

A separate event is emitted for any route that is not the home route (/). This helps to specifically track the usage of additional features of the website beyond the landing page.

Successful file reads emit an event confirming that the content has been successfully retrieved and is being served.

When the server fails to read a file because it doesn't exist, an event is emitted, which allows us to track missing files or incorrect route accesses.

The server writes all events to both the console and a log file (server.log). The log file is appended with new events each time they occur, providing a persistent record that can be reviewed later for debugging.

Bonus: Three user stories included. Branches created and merged back to the main.