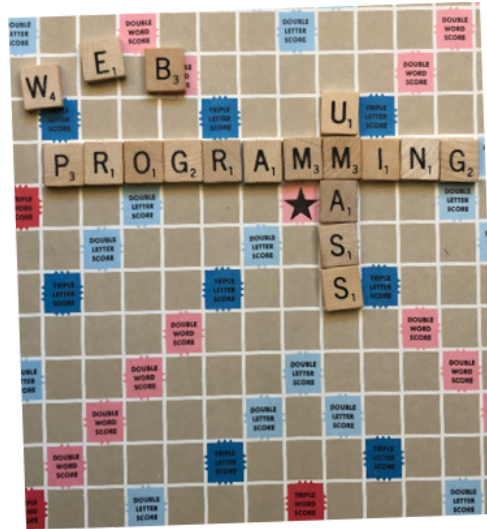


**COMPSCI 326 - Web Programming**  
**Homework 7 - Server Starter - *individual assignment***  
**due November 1, 2021, 11pm EDT**  
(GitHub classroom link: <https://classroom.github.com/a/HnDejV3> )



This is the seventh part of a series of assignments around the game of [Scrabble](#). We hope that it will be a fun experience in progressively learning all pieces of modern web development, so as to engineer a fully functional game. In this assignment, you will work on setting up a basic server capable of saving statistics about the game.

Please submit this assignment on GitHub Classroom. There will be an automated grader that will check the functionality of your submission. It will be helpful to come up with test cases, and we encourage you to share them amongst each other; this will make everyone's code better and is actually how Quality Assurance (QA) can work in practice. However, this is an individual assignment and you **cannot share code**; submissions will be run against plagiarism detection tools. Additionally, we will be spot checking the code for good coding practices. It is expected your code **does not** contain (1) extraneous variables/code, (2) missing semicolons, (3) missing curly braces, (4) use of double equals, (5) use of `let` when a `const` would suffice, (6) use of `var`. Furthermore, you should use whitespace consistently and to make the code legible. Now that you've learned how to use ESLint, it should be easy to satisfy these requirements.

You will find a template when you create your repository. You should import the files you previously used for homework 6 into a directory called `client`. **Please do not rename any of the existing files or further change the directory structure.** You are free to create more files and import them. However, you **cannot** use any external modules beyond those provided without prior permission.

# 1. A Basic Node.js Server

You will implement a basic REST API with 4 endpoints in `server/index.js` that we will use to track some statistics about the game. We will track the highest word scores as well as the highest game scores.

- **POST /wordScore**  
This endpoint will be used to save a word score on the server. The body of the request will be in JSON format, as an object with three keys: `name`, `word`, and `score`. The response should only consist of a 200 status code, with no body.
  - *Example request body:* `{ "name": "Asterix", "word": "test", "score": 8 }`
  - `curl -X POST -H 'Content-Type: application/json' -d '{ "name": "Asterix", "word": "test", "score": 8 }' localhost:8080/wordScore`
- **GET /highestWordScores**  
This endpoint will be used to get the top 10 word scores saved on the server (or top X scores if only  $X < 10$  scores are saved). It should return a JSON response, which will be an array of objects consisting of a word and a score.
  - *Example response body:* `[{ "name": "Asterix", "word": "test", "score": 8 }, ..., { "name": "Obelix", "word": "school", "score": 11 }]`
- **POST /gameScore**  
This endpoint will be used to save a game score for a single player on the server. The body of the request will be in JSON format, as an object with two keys: `name` and `score`. The response should only consist of a 200 status code, with no body.
  - *Example request body:* `{ "name": "Asterix", "score": 361 }`
- **GET /highestGameScores**  
This endpoint will be used to get the top 10 game scores saved on the server (or top X scores if only  $X < 10$  scores are saved). It should return a JSON response, which will be an array of objects consisting of a name and a score.
  - *Example response body:* `[{ "name": "Asterix", "score": 650 }, ..., { "name": "Obelix", "score": 513 }]`
  - `curl -H 'Content-Type: application/json' -d '{}'`  
`localhost:8080/highestGameScores`

Finally, the saved scores should be persistent: if the server is restarted, it should still have the previous scores saved. For now, you can just use a JSON file that you read and write to. You will need the following Node.js built-in modules: [http](#), [fs](#), [url](#).