CP 325.6:

Building a Server with Node and Express

Version 1.0, 08/14/23

Click here to open in a separate window.

Introduction

This document will challenge you with taking the next steps in your Capstone Project journey.

Assignment Objective

- Build a Node and Express server for your application.

Submission

This portion of the Capstone Project will not be submitted on its own; only the final project will be submitted. You are encouraged to seek feedback from instructors and peers along the way.

Instructions

The next big step for your application is to create a web server to handle server-side logic, data processing, and interfacing with your database (which will be the next step, soon)!

Now that you have an understanding of how to build REST APIs and serve content from server to client, you can begin to think about how you want your front-end to interact with your back-end.

The front-end is already in place! (Feel free to continue working on it.) Now, take the time to replace any API calls to third-party placeholder APIs with calls to your own API once you have created it. If you still want to use a third-party database for placeholder data, simply move those API calls to your server rather than your client application, and have the client request those resources from *your* server instead of the third party.

How your server interacts with the application, and the way your data is modeled, stored, and manipulated, *will* change over the course of time. Again, you still have time to perfect things, so for now, just experiment. Discover what works and what does not, and how the puzzle pieces of your application can fit together.

While testing your interactions, remember that your data is not being stored in a database yet, so every time you restart the server, it will reset to its default values! This is not a bad thing, but do not let it confuse you during testing. If you are using nodemon during development (highly recommended), this means that the data will reset any time a change is made to your files.

Partial Requirements

Let’s look at which Capstone Project requirements are related to this stage of development. Requirements not related to this stage of development have been omitted. For a full list of requirements, see CP 323.1 - Planning a Project, or CP 323.10 - Capstone Completion.

(20%) Project Structure, Standardization, and Convention	**Weight**
Project is organized into appropriate files and directories, following best practices.	**2%**
Project contains an appropriate level of comments.	**2%**
Project is pushed to GitHub, and contains a README file that documents the project, including an overall description of the project.	**5%**
Standard naming conventions are used throughout the project.	**2%**
Ensure that the program runs without errors (comment out things that do not work, and explain your blockers - you can still receive partial credit).	**4%**
Level of effort displayed in creativity, presentation, and user experience.	**5%**

(12%) Core JavaScript	**Weight**
Demonstrate proper usage of ES6 syntax and tools.	**2%**
Use functions and classes to adhere to the DRY principle.	**2%**
Use Promises and async/await, where appropriate.	**2%**
Use Axios or fetch to retrieve data from an API.	**2%**
Use sound programming logic throughout the application.	**2%**
Use appropriate exception handling.	**2%**

(19%) Server	**Weight**
Create a RESTful API using Node and Express. * For the purposes of this project, you may forgo the HATEOAS aspect of REST APIs.	**7%**
Include API routes for all four CRUD operations.	**5%**
Include at least one form of user authentication/authorization within the application.	**2%**

(35%) Front-End Development	**Weight**
Interface directly with the server and API that you created.	**5%**

The following section is NOT included in the requirements for this project. Completing this section is NOT required. This section will NOT negatively impact your grade if left unfinished.

This section is intended for learners looking to go the extra mile by showcasing additional skills such as project management, and optional technologies like TypeScript.

You must complete ALL other requirements to receive credit for this section; however, this extra credit will not be included if you have already received the maximum 100% grade. The extra credit can only offset points lost elsewhere.

(5%) Extra Credit	**Weight**
Adhere to Agile principles and the Scrum framework. Perform stand-up sessions (with an instructor) when possible.	**1%**
Successfully track your project using a software like Jira.	**1%**
Build your application primarily with TypeScript.	**3%**