PER SCHOLAS

CP 325.4:

Pulling Data from an API

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

- Add asynchronous actions, like fetching data from an API, to the project.

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

You have learned some more advanced concepts such as object-oriented programming fundamentals, how JavaScript works in the background using the call stack and event loop, asynchronous JavaScript techniques using Promises, fetching data from external APIs, and using modules and imports to organize code.

At this stage, you have all of the JavaScript tools that you need to create a fully-functional front-end application, and have likely already done so on a smaller scale.

Your challenge for this assignment is to expand upon the functionality that you have added to your Capstone Project thus far:

- Use your knowledge of the event loop to fix old bugs, if they exist.
- Use async/await and Promises to tackle issues with synchronous code.
- Use fetch or Axios to interact with an external API and retrieve data for use throughout the application.
- Reorganize your code using modules.

All of these possibilities and more exist for you. Make a plan for how you would like to progress your project, and get coding!

While you are currently limited to using external APIs and the databases they interface with, you will be creating your *own* API and database for this application (soon)! You can use a placeholder API of your choosing to set up some of these interactions ahead of time, so that you are more comfortable with your approach when building your own systems.

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%

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%