CMPS 350 Web Development - Assignment 4: WebAPI

Deadline - Thursday, April 25 @ 11:59 PM

Objective

This assignment aims to evaluate your proficiency in creating a Web API with Node.js and NextJS. You will extend the LibraryLocus App to be accessible via a Web API and conduct tests using Postman, Mocha-Chai and a user interface crafted in vanilla JavaScript. To understand more about the requirements of the app, watch this Video.

Setup Instructions

- **Open Base Code**: Open the provided LibraryLocus Base Code, which contains all the dependencies necessary for completing the project.
- **Install Dependencies**: After opening the project, run ***npm i*** in the terminal to install all necessary dependencies.
- **Run the Application**: Execute ***npm run dev*** to start the application in development mode.

Assignment Description

Continue developing the LibraryLocus app by implementing the following functionalities in a repository class that manages reading and writing operations:

BookRepo class Functionalities

- **addBook**: Allows adding a new book
- **updateBook**: Allows updating an existing book using ISBN.
- **deleteBook**: Allows deleting a book using its ID or ISBN.
- **getBook(name)**: Retrieves a book by its name. Returns the book object or a "Not found" exception if the book does not exist.
- **getBooksByPageCount(pageCount)**: Retrieves books with pages greater than or equal to the specified pageCount (e.g., pageCount=200 retrieves all books with 200 pages or more).
- **getBooksByAuthor(author)**: Retrieves all books authored by the specified author, including books co-authored.
- **getBooksByCategory(category)**: Retrieves books from a specific category (e.g., category = "Programming" retrieves all programming books).
- **getAuthorsBookCount()**: Generates a map listing authors alongside the count of books they have authored.

Web API Endpoints

HTTP Verb	URL	Functionality
GET	**/api/books**	Returns all the books
GET	**/api/books?name=**	Returns one book by the given name
GET	**/api/books?pageCount=**	Returns books with pages >= the specified pageCount
GET	**/api/books?author=**	Returns all books authored by the specified author
GET	**/api/books?category=**	Returns books from a specified category
GET	**/api/books/summary**	Returns a map of author names with the count of books they have authored
POST	**/api/books/**	Adds a book
PUT	**/api/books/:isbn**	Updates a book
DELETE	**/api/books/:isbn**	Deletes a book

Testing

- **Postman**: Use Postman to manually test the functionality of your Web API.
- **Automated Testing**: Create a **book-service.spec.js** file to test the Web API using **chai-http**.
- Test using the **vanilla JS UI implementation** found in the public folder.

Submission Guidelines

After completing the assignment, submit your project along with the completed Testing-Sheet.docx. Save both items in the Assignments/assignment4 folder. Finally, sync your repository to upload your work to GitHub.