

Bookstore application

- The api should allow users to manage a collection of books:
- Each book have following attributes:
 - title
 - author
 - ISBN (International Standard Book Number) -> ID
 - price

Operations:

Create operation:

- Implement an endpoint to add a new book to the bookstore's inventory.
- Ensure that the endpoint accepts JSON data with the book's attributes (Title, Author, ISBN, Price).
- Generate a unique identifier for each book entry.

Read operation:

- Implement an endpoint to retrieve information about a specific book using its unique identifier (e.g., by ISBN).
- Implement an endpoint to retrieve a list of all books in the inventory.

Update operation:

- Implement an endpoint to update the information of an existing book using its unique identifier (e.g., by ISBN).
- Allow users to update any of the book's attributes (Title, Author, ISBN, Price).

Delete operation:

- Implement an endpoint to delete a book from the inventory using its unique identifier (e.g., by ISBN).

Validation:

- Implement validation checks to ensure that book data is consistent and complete (e.g., required fields are present, ISBN is valid, Price is a non-negative number).

Persistence:

- Store the book data in a database of your choice (in this case use JSON files).

Submission:

- Provide the codebase with clear instructions on how to set up and run the application
- create Operation (C):
 - Request:
 - HTTP Method: POST
 - Endpoint: `/api/books`
 - Request Body: JSON data with the book's attributes (Title, Author, ISBN, Price).
 - Response:
 - HTTP Status Code 201 (Created) if the book is successfully added.
 - HTTP Status Code 400 (Bad Request) if the request data is invalid or incomplete.
 - HTTP Status Code 500 (Internal Server Error) for server-related errors.
- Read Operation (R):
 - Request to Retrieve a Specific Book:

- HTTP Method: GET
 - Endpoint: `/api/books/{ISBN}`
 - Replace `{ISBN}` with the ISBN of the book to retrieve.
- Response:
 - HTTP Status Code 200 (OK) if the book is found, along with the book's data in the response body.
 - HTTP Status Code 404 (Not Found) if the book with the specified ISBN is not found.
 - HTTP Status Code 500 (Internal Server Error) for server-related errors.
- Request to Retrieve All Books:
 - HTTP Method: GET
 - Endpoint: `/api/books`
- Response:
 - HTTP Status Code 200 (OK) along with a JSON array containing the details of all books in the inventory.
 - HTTP Status Code 500 (Internal Server Error) for server-related errors.
- Update Operation (U):
 - Request:
 - HTTP Method: PATCH
 - Endpoint: `/api/books/{ISBN}`
 - Replace `{ISBN}` with the ISBN of the book to update.
 - Request Body: JSON data with the book's attributes to be updated (Title, Author, ISBN, Price).
 - Response:
 - HTTP Status Code 200 (OK) if the book is successfully updated.
 - HTTP Status Code 400 (Bad Request) if the request data is invalid or incomplete.
 - HTTP Status Code 404 (Not Found) if the book with the specified ISBN is not found.
 - HTTP Status Code 500 (Internal Server Error) for server-related errors.
- Delete Operation (D):
 - Request:

- HTTP Method: DELETE
- Endpoint: `/api/books/{ISBN}`
- Replace `{ISBN}` with the ISBN of the book to delete.

- Response:

- HTTP Status Code 204 (No Content) if the book is successfully deleted.
- HTTP Status Code 404 (Not Found) if the book with the specified ISBN is not found.
- HTTP Status Code 500 (Internal Server Error) for server-related errors.