Backend Challenge

CONTACTO

Morada

Edifício LACS,
Estrada Malveira da Serra, 920
2750-834 Cascais

Telefone & E-mail

(+351) 214 229 114
rh@devlop.systems

devlop.systems

General Instructions:

- Submission: Submit your solution via email.
- Tools: You can use any IDE, text editor to complete the exercises
- Use any language and framework you are comfortable with but we give preference to .NET.
- Document setup and any configuration required to run the API locally.
- Provide a README file detailing the API usage and how to run tests.
- Questions/Clarifications: If you have any questions or need clarifications, please feel free to contact us.

Challenge: Book Information Aggregator API

Scenario:

You are tasked with building a small REST API for a book information service. This service will allow users to search for books and view detailed information about them. The core functionality is to fetch book data from an external API, process it, and return the details in a well-structured response.

Requirements:

1. File Initialization
    - On application startup, the service should:
        1. Read the contents of books.json and load the data into memory.
        2. Ensure the data is available to endpoints like /books for listing and retrieval.
2. Endpoints
    - GET /books/search?query={query}: Searches for books by title or author. This endpoint will call an external API to retrieve book data based on the query parameter.
    - GET /books/{id}: Fetches detailed information about a specific book by ID, also from the external API.
    - POST /books: Accepts a JSON payload to add a new book to the system's local storage (simulated in-memory for this task).
    - GET /books: Returns all locally stored books added via the /books endpoint.
3. External API Integration
    - Use a public book API (Open Library Search API | Open Library) to fetch information about books.
4. Docker (bonus)
    - Being able to run the solution using a docker-compose file

Sample books.json File:

[{ "id": "1", "title": "To Kill a Mockingbird", "author": "Harper Lee", "description": "A novel about the serious issues of rape and racial inequality.", "published_year": 1960 }, { "id": "2", "title": "1984", "author": "George Orwell", "description": "A dystopian novel set in a totalitarian society under constant surveillance.", "published_year": 1949 }, { "id": "3", "title": "Moby Dick", "author": "Herman Melville", "description": "The narrative of Captain Ahab's obsessive quest to kill the white whale, Moby Dick.", "published_year": 1851 }]

Evaluation Criteria:

1. Code Quality
2. Error Handling
3. Testing
4. Best Practices in API Design
5. Documentation