Ruby Challenge

Description

This project is designed to test your knowledge of back-end web technologies, specifically in Ruby and assess your ability to create back-end products with attention to details, standards, and reusability.

Assignment

The goal of this exercise is to create a REST API for a library store.

Each book of this library will have at least the following attributes:

- Book id
- Book title
- Book author
- Book isbn
- Book price
- Book short description

Mandatory Features

1. Implement endpoints for:
    a. Add a new book
    b. Request information about a single specific book
    c. Request information about a collection of books
    d. Delete a book
    e. Search by a book title
    f. Search by a book isbn
2. Implement rate-limit
3. Add Unit tests.

Bonus (Optional)

- Use docker-compose to provide the environment.
- Include some integration tests

Considerations

- SRP: Single Responsibility Principle (Classes are self contained. They do the task they need to do and nothing else).
- Liskov's Substitution Principle: Interfaces (OOP, Swap principle... Makes the program able to be extended).
- Dependency Inversion Principle: Code should depend on interfaces, no concrete implementations.
- Coupled code: If code has too few classes we should not accept this candidate, even if the program works perfectly. Code violates SRP, classes demonstrate mixed concerns. Bad OOP in general.
- Abuse of class methods and singleton: Usually this indicates a junior candidate since this makes difficult to use the Substitution Principle.
- Duplicated code: Reused code should be encapsulated.
- Include well-known libraries.
- Packaging complete. Zip should contain a readme file containing all the necessary instructions about how the application be can set up, run and tested. Please include some examples of how to call the endpoints.
- Project structure: It should be the standard Ruby project layout, no IDE specific or custom.
- Keep your code versioned with Git locally.

Deliverables

When you finish the assignment, send a zip file (don't forget to include the .git/ folder.) or upload your project to your Git repo (Github, BitBucket, etc...) and share the repository link with your initial contact via email. Indicate which, if any, of the bonus tasks you completed.

If you didn't manage to finish everything, please tell us which parts you completed.