Designing a REST API - Library

Roman Kuzmin
2024-08-03

Functional and non-functional requirements

1. Functional requirements

The reader registers in the system and then has the opportunity to:
- search (by author/title);
- place an order for a Book from the Catalogue.
An unregistered Reader cannot order a book.
For the catalog, implement the ability to sort books:
- by titel;
- by author;
- by publication;
- by the date of publication.
The librarian issues a book to the reader for subscription or to the reading room. The book is issued to the Reader for a certain period. If the book is not returned within the specified time, the reader will be charged a fine.
The book can be present in the library in one or more copies. The system keeps track of the available number of books.
Each user has a personal account in which registration information is displayed, as well as
1) for the reader:
- a list of books that are on the subscription and the date of possible return (if the date is overdue, the amount of the fine is displayed);
2) for the librarian:
- list of readers' orders;
- list of readers and their subscriptions.
The system administrator has the following rights:
- adding/deleting a book, editing information about a book;
- creating / deleting a librarian;
- blocking / unblocking the user.

2. Non-functional requirements

- Performance: Response time under 200 ms, handle 1000 requests per second.
- Scalability: Horizontal and vertical scalability.
- Security: JWT authentication, role-based access control, TLS/SSL encryption.
- Availability: 99.9% uptime, failover mechanisms.
- Maintainability: High code quality, error logging, comprehensive documentation.
- Usability: Consistent API design, detailed error messages.
- Reliability: Data consistency, backup, and recovery.
- Compliance: GDPR compliance, audit trails.
- Compatibility: API versioning.
- Efficiency: Optimal resource usage.