**Distributed and Operating Systems Spring 2019**

**Lab 1: Bazar.com: A Multi-tier Online Book Store**

**Manar Jber + Rahaf Nasralleh 11924677 + 11923488**

# Multi-tier-Online-Book-Store

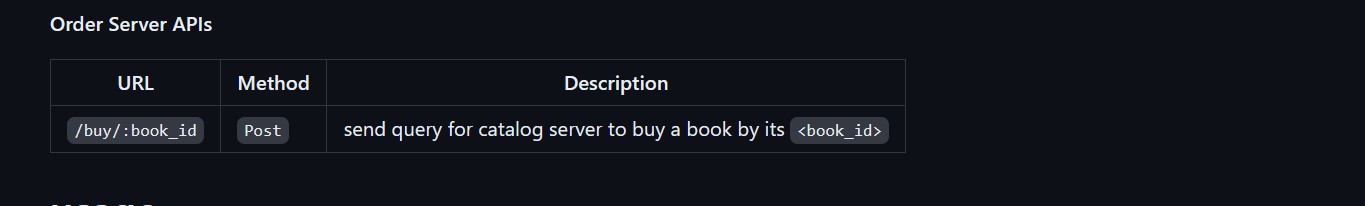
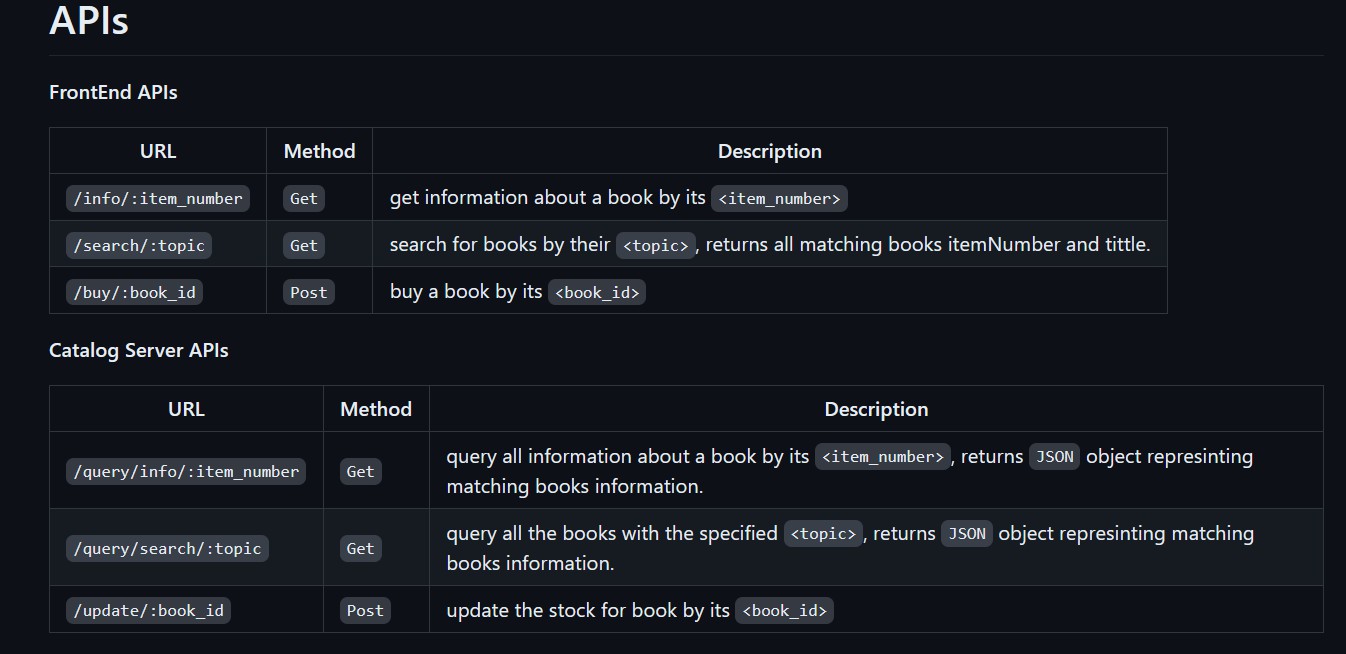
The store will employ a two-tier web design - a front-end and a backend - and use microservices at each tier. The front-end tier will accept user requests and perform initial processing. The backend consists of two components: a catalog server and an order server.

# Architecture

1. Catalog Server: It contains database containing information about the available books. This information includes details such as the book title, quantity in stock, price, and the topic of the book. functions as a microservice dedicated to managing and providing information about the available books. Implemented as an independent process, it offers a RESTful API supporting operations such as querying books by subject or item number and updating book details.
2. Order Server: Responsible for managing customer orders. Implemented as a microservice, the order server communicates with catalog server and front-end server, to facilitate the purchase process. This microservice is designed to handle the purchase operation efficiently and provides a single operation endpoint, typically through a RESTful API.
3. Frontend Server: Initial point of contact for user requests and orchestrates interactions between the user interface and the backend services. Implemented as a microservice, the front-end server supports a variety of user operations through a RESTful API.

**Implementation**

This project was developed using Node.js and the Express framework, leveraging its support for building lightweight microservices. The choice of the SQLite database was intentional, as it provides a simpler and more lightweight solution for data storage, aligning with the goal of creating a streamlined and efficient program architecture.



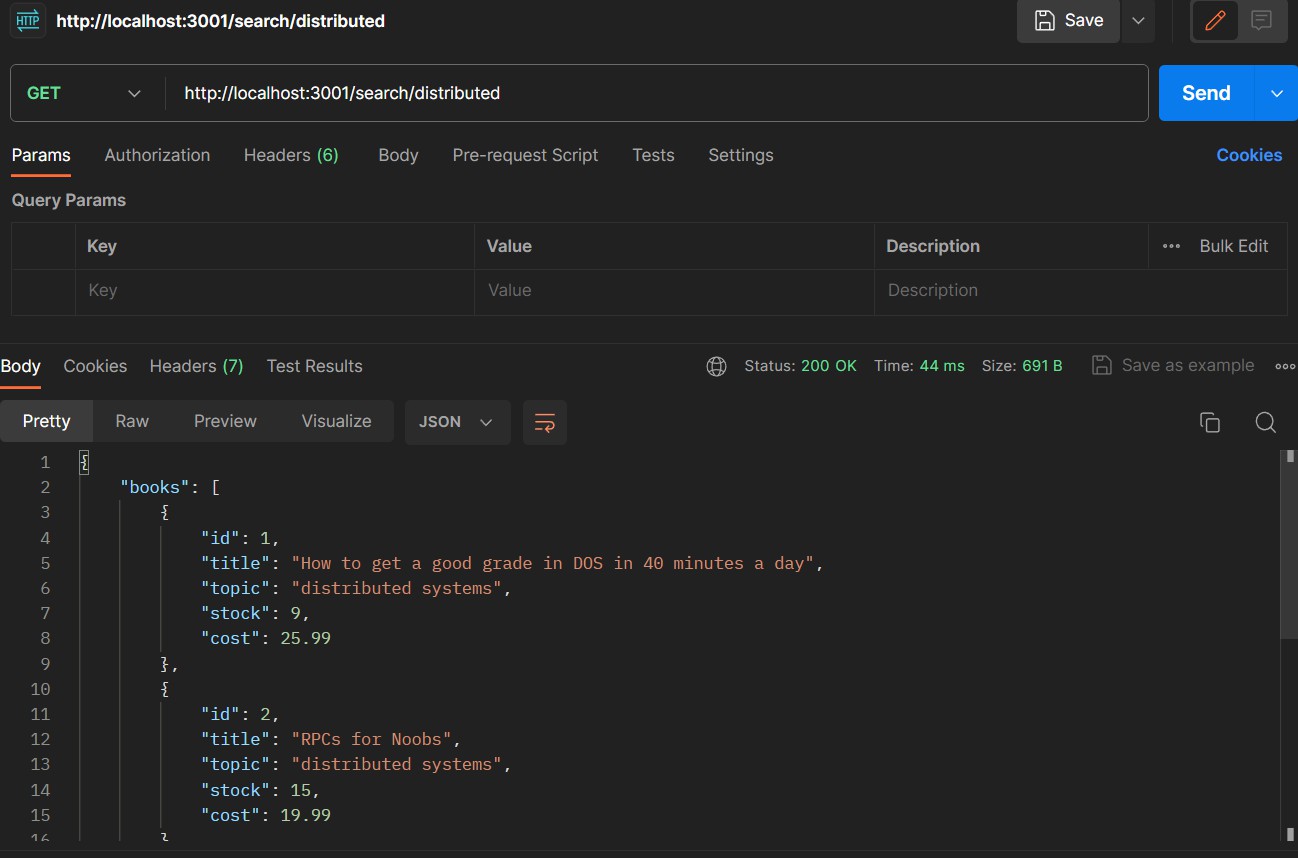
**usage**

The project deals with Database sqlite3, so you must download this dependency on your device Download the file and place each server in a different project, configuring it for Node.js and Framework Express, to run the front-end server. Type node app.js It will work on port 3001. for catalog-server do same thing It will work on port 3004. for order- server It will work on port 3007.

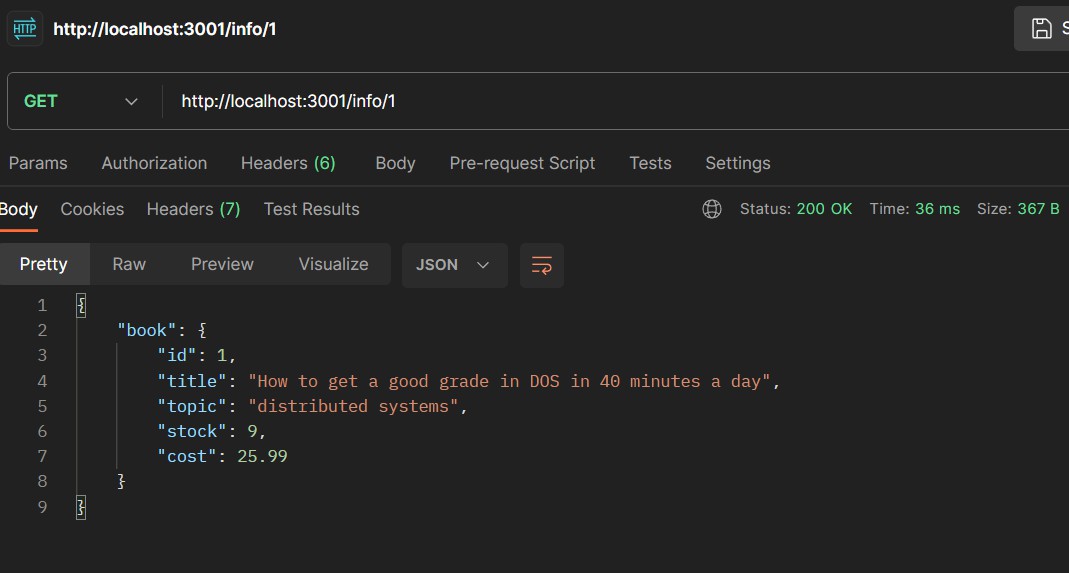
**Testing**

Frontend address: http://localhost:3001. catalog-server address: http://localhost:3004. order-server address: http://localhost:3007.

**Search for books by topic** http://localhost:3001/search/distributed%20systems



**Ask for information about a selected book** http://localhost:3001/info/1



**Buy Books** http://localhost:3001/buy/1

