An-Najah National University

College of Engineering and Information Technology
Computer Engineering Department

Distributed operating systems

Report: Bazar.com

Student names:

Ameen Abo Diak

Masa Zablah

Introduction:

The Bazar.com project is a two-tier web design for a small online bookstore. It employs microservices at each tier, with a front-end server handling user requests and initial processing, and backend servers managing the catalog and orders.

How It works:

- Front-End Server: This server receives requests from users and does initial work. Three functions are available: purchase, info, and search. Based on the user's request, these processes send queries to the order servers and catalog servers.
- Catalog Server: The catalog server keeps track of all the book information, such as the quantity of books available, their price, and the topic of each book. It allows updates for modifying the price or stock amount in addition to queries for searching by subject or item number.
- Order Server: The order server keeps track of every order that is placed. Upon receiving a purchase request, the system queries the catalog server to confirm the item's availability and, if it is, reduces the stock quantity.

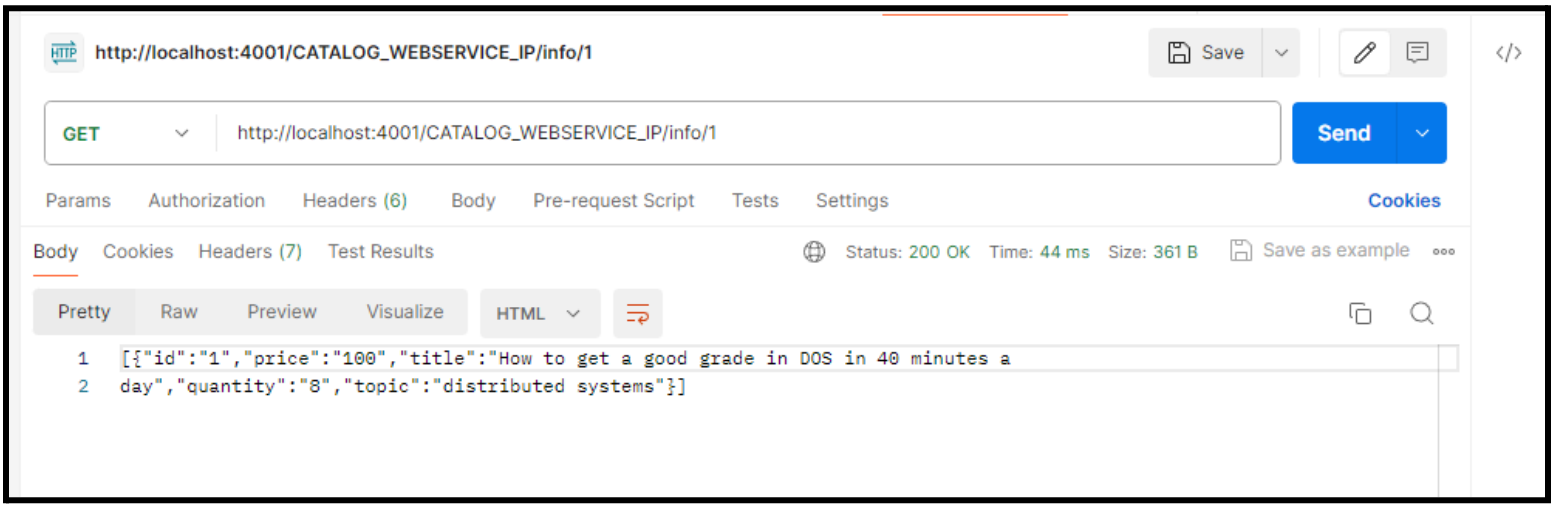
Design Tradeoffs:

CSV Files: Rather than using complex databases, CSV files were used to store catalog and order data persistently in order to maintain a basic implementation.

HTTP REST Interfaces: Using HTTP REST interfaces for allowing communication between services makes interface easier and makes it possible to test the connection with Postman .

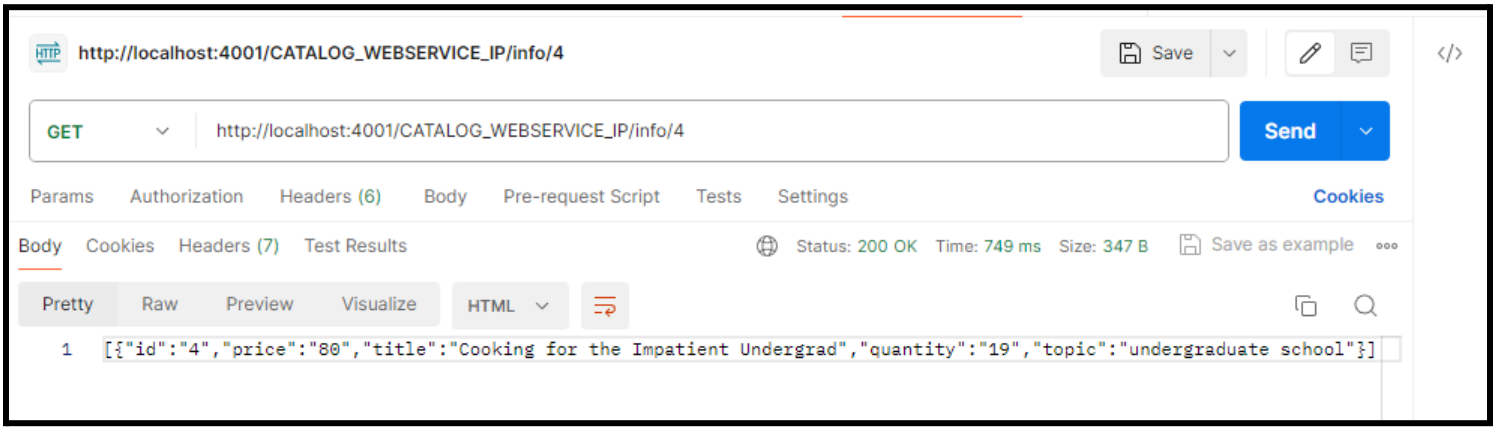
We used postman to test our project, and to package our program into containers, we went with Docker. Every service (front-end, order server, and catalog) operates within a separate container.

This is the search operation with the item number:

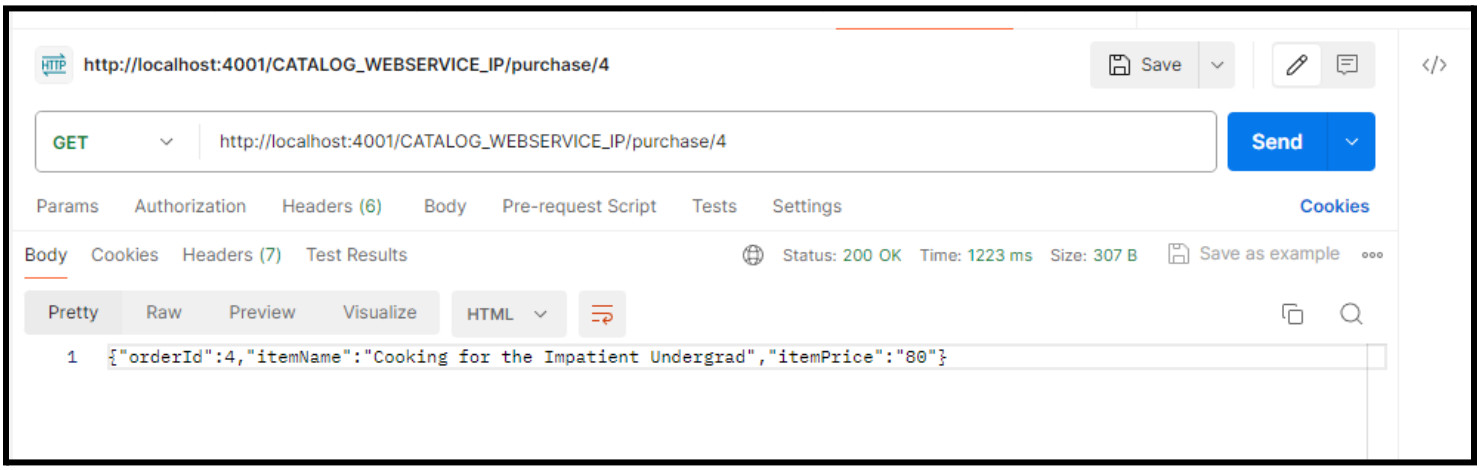


For the purchase operation:

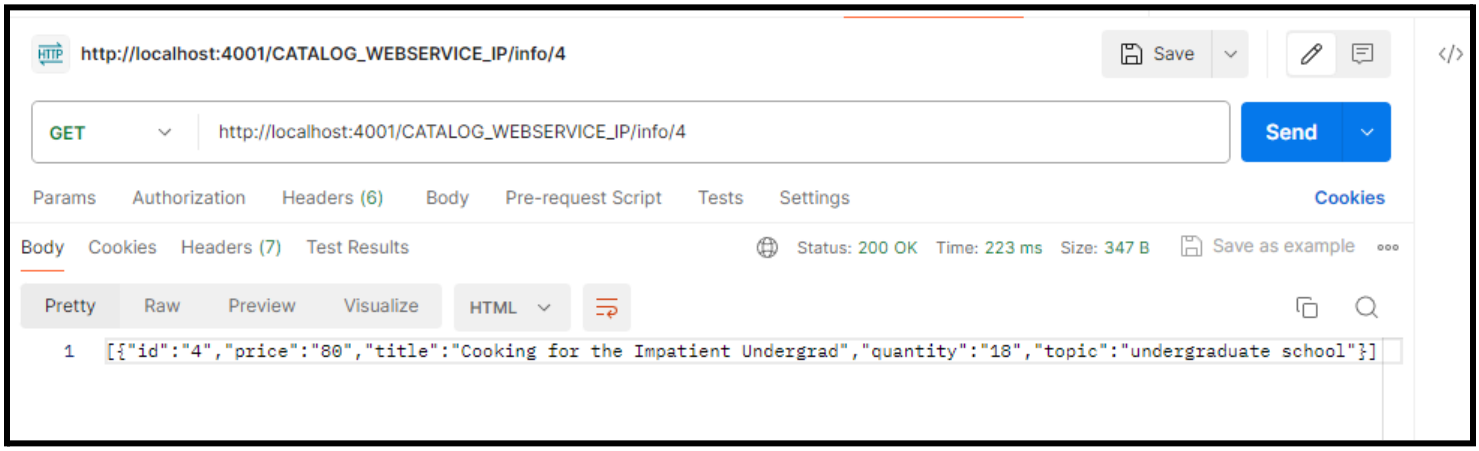
This is a screenshot of the info of the item with id=4, the quantity is 19:



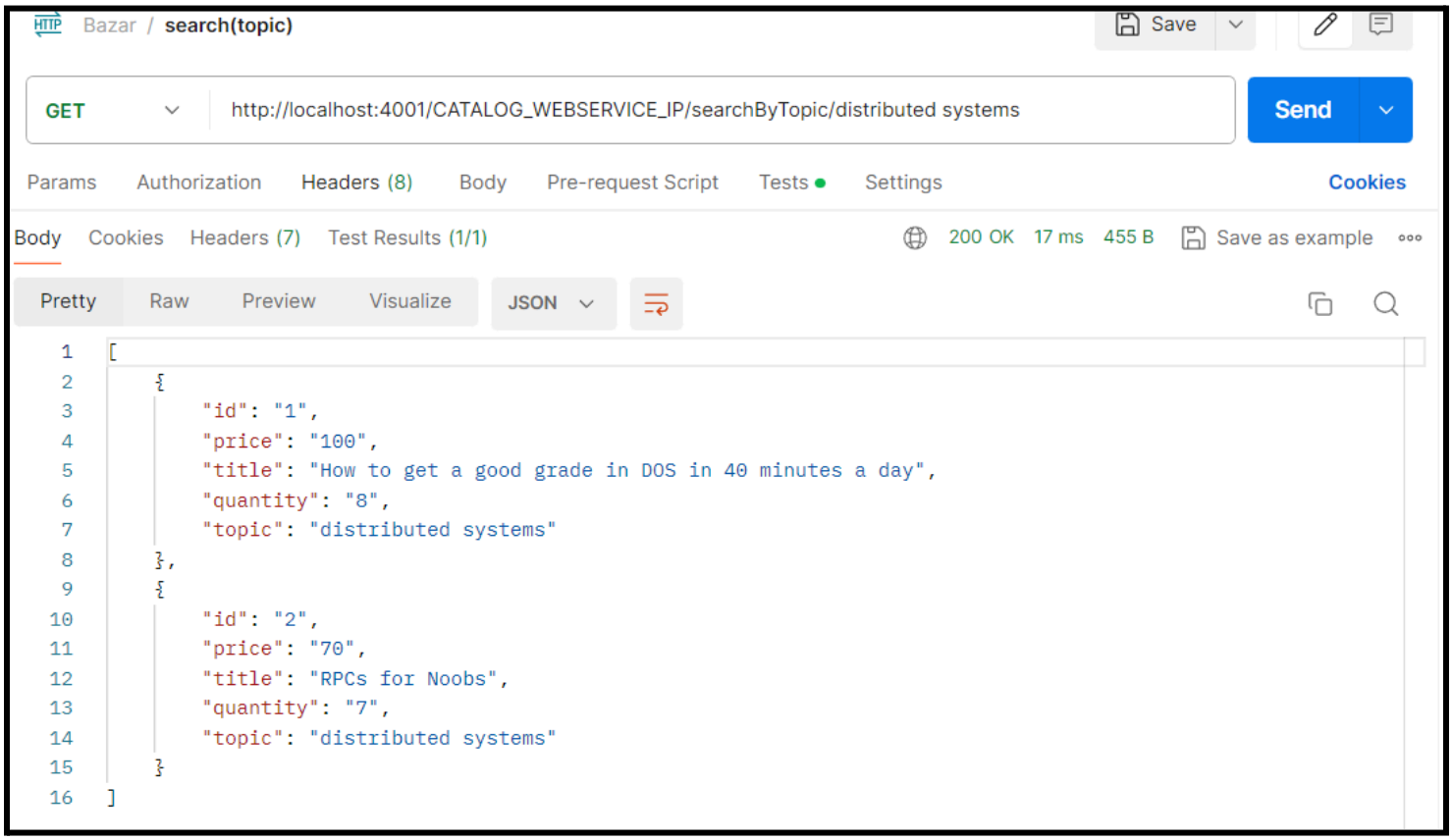
When we purchase this item using this http request:



The quantity if this item in stock will be decremented to 18 as shown:



Users can search by topic to retrieve all books associated with that topic.



Future Enhancements

Front-End Development

To significantly enhance user engagement and accessibility, a user-friendly front-end will be developed. This interface will be built using modern web technologies such as React, Vue, or Angular to provide a seamless and intuitive user experience. Responsive design principles will be applied to ensure the application is fully accessible across various devices and screen sizes.

Admin Dashboard

An administrative dashboard will be implemented, offering comprehensive CRUD (Create, Read, Update, Delete) capabilities for book management. This feature will include secure authentication and authorization to ensure that only authorized admin users can access and perform these operations.

Real-Time Stock Alerts for Admin

The system will incorporate real-time notifications to alert administrators when books are nearing low stock levels or are out of stock. This enhancement aims to facilitate prompt stock management decisions, utilizing WebSockets or server-sent events for instantaneous communication.

Online Payment Integration

The system will integrate a secure online payment gateway, such as Stripe or PayPal, to facilitate online transactions. This integration will focus on secure data handling, compliance with payment industry standards, and providing a smooth payment process for users.