DOS Project Part1
Bazar.com

Authors

Amr Ayman Kurdi (12027757)
Zaid Jamal Balout (12027806)

Affiliations

Instructor : Dr Samer Arandi

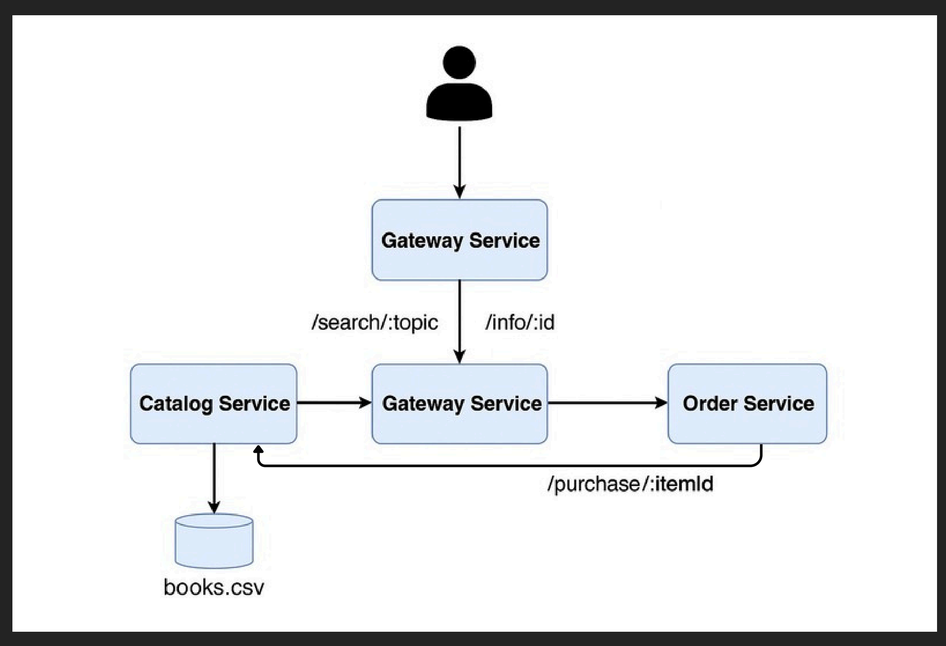
1 Abstract And Aims

This report covers the implementation of bazar.com, which consists of three main servers (Catalog, Order, and Gateway), each with a specific role, and shows how they were containerized using Docker and managed together using a docker-compose file.

you can find the project at : https://Github.com/amrkurdi12027757/DOS

2 The Task

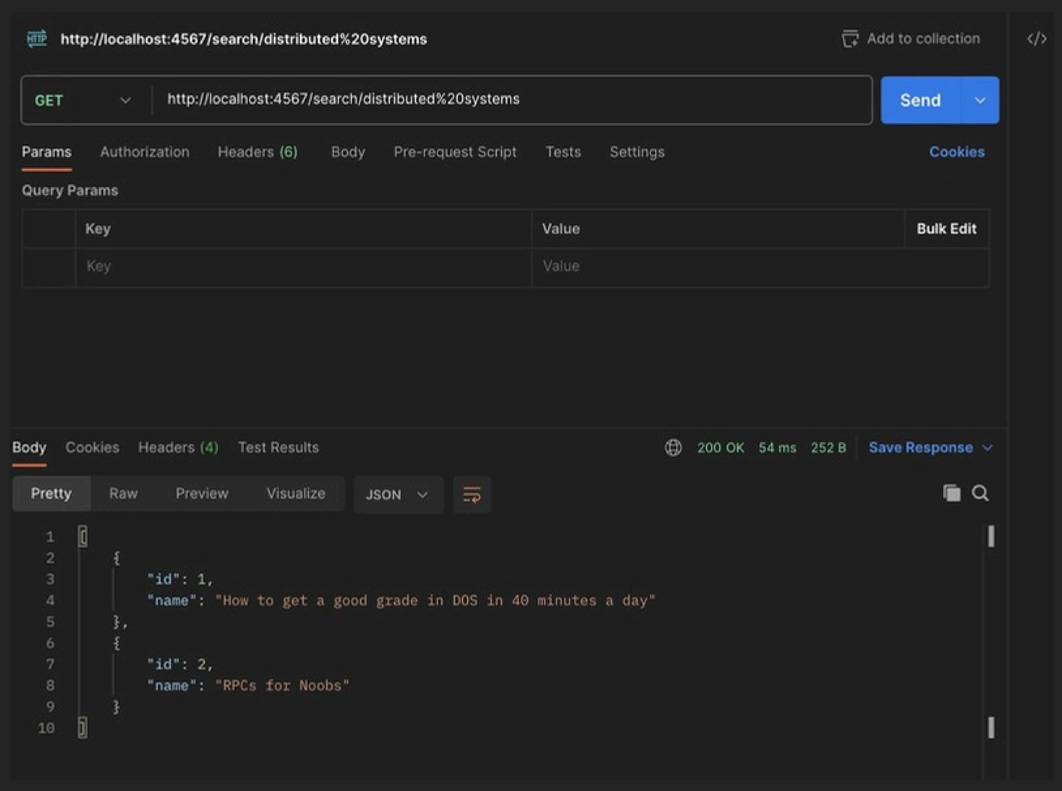
In this part of the project, we created bazar.com just like shown in the picture below:

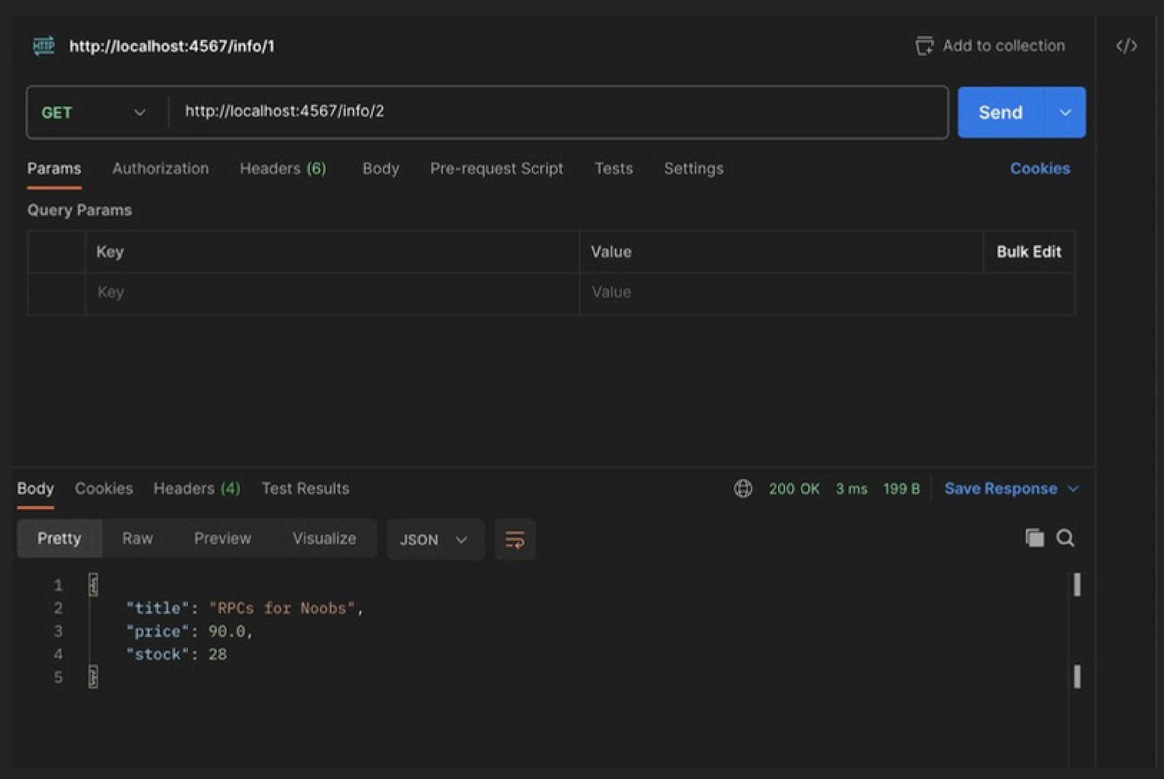


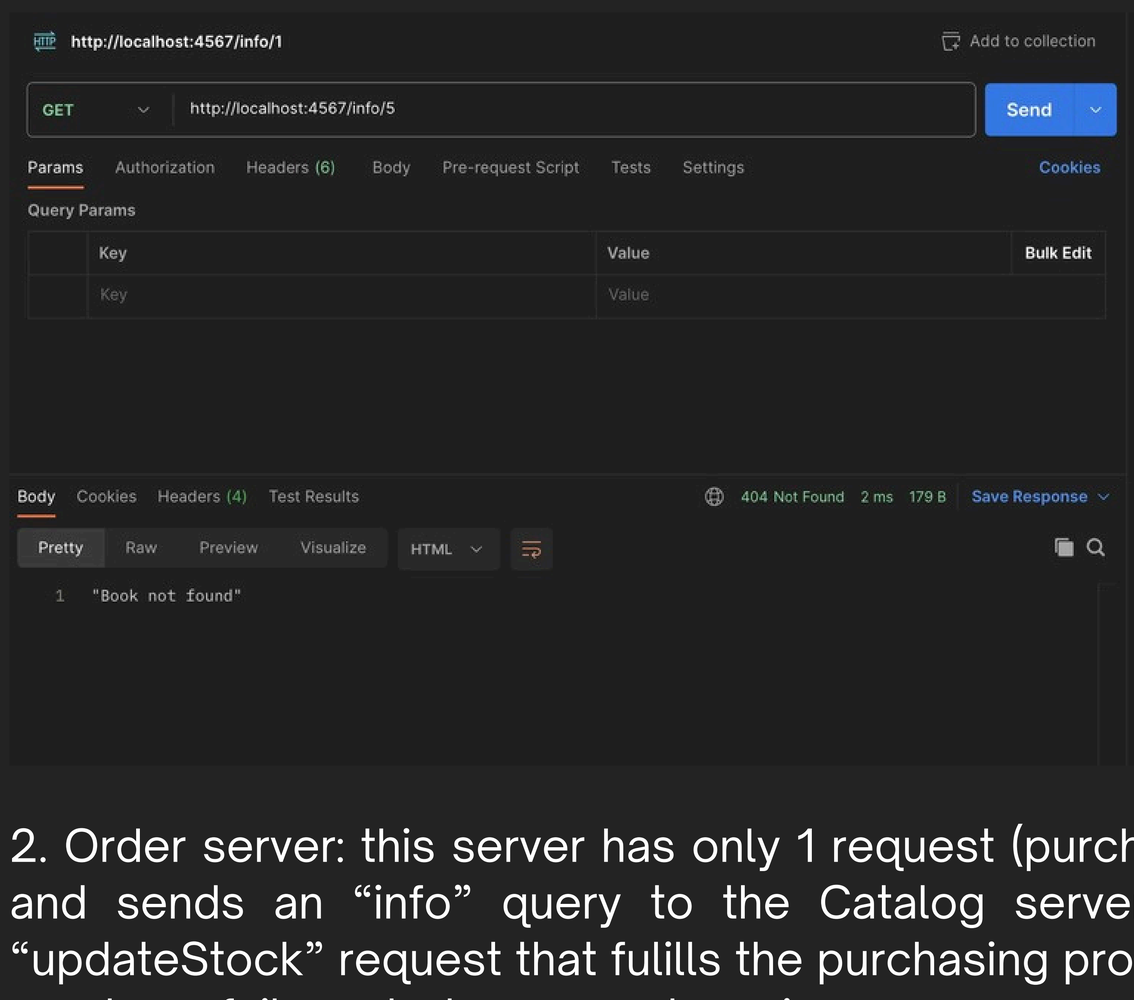
The project is made up of 3 parts:

1. Catalog server – This part handles communication with the book file. It includes two main queries:
   - "search": takes a topic and returns books under that topic.
   - "info": takes a book ID and returns the book's name, stock, and price.
2. There's also another function called "updateStock", used by the order server. It takes a book ID and reduces the stock by 1.

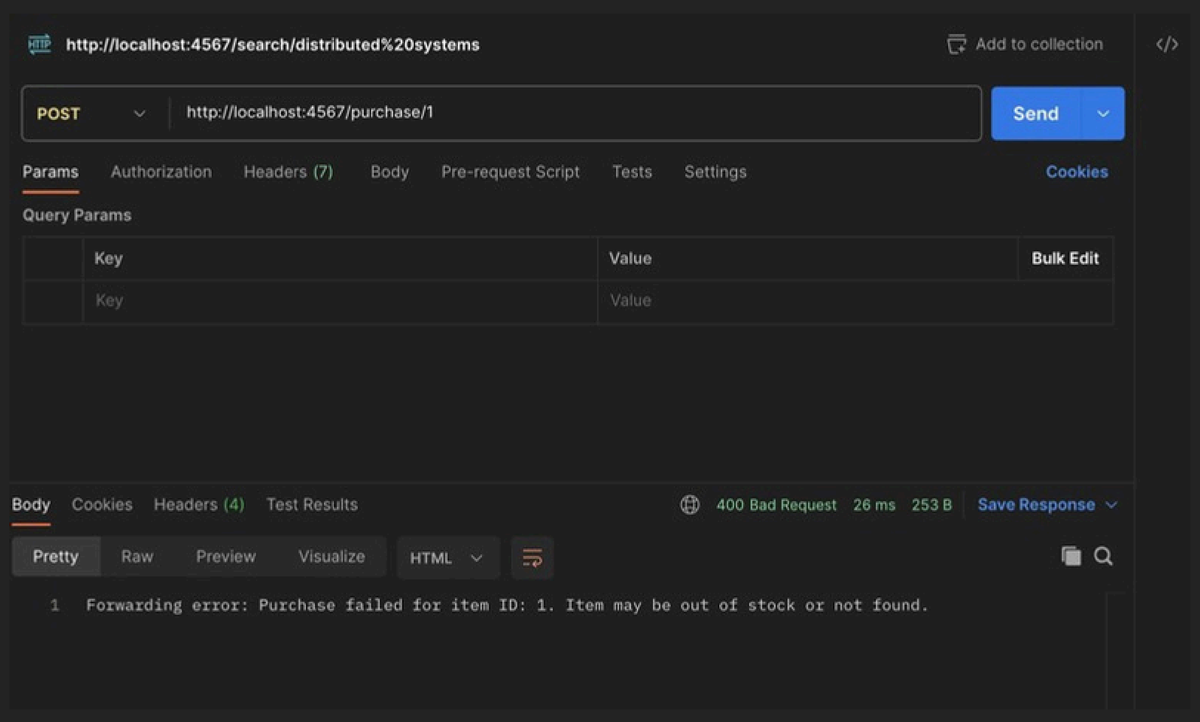
You can see the "search" and "info" queries shown below:

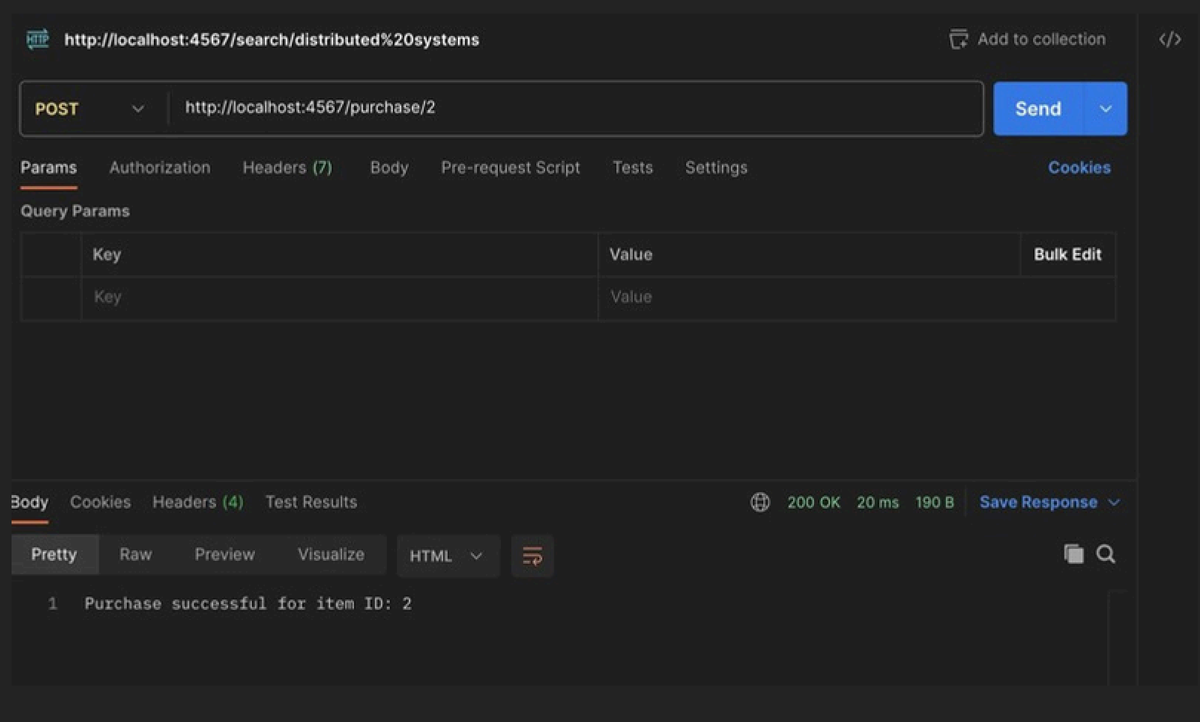






2. Order server: this server has only 1 request (purchase) that takes the book ID as a parameter and sends an “info” query to the Catalog server, if the stock is not 0, then it sends a “updateStock” request that fulills the purchasing process. Shown below are examples of when a purchase fails and when a purchase is a success:





3. Gateway server – This server’s job is to simply pass requests to the Catalog and Order servers, and then send back the responses to the client. All the requests mentioned above go through the Gateway server (you can tell by the same port number).

3 Dockerization

We created 3 Docker containers, one for each server. To make things easier and show how useful Docker is, we also added a docker-compose file. When you run it, all three servers start at the same time with their needed settings already defined.
We used the same Dockerfile for each server, with the only difference being the name of the JAR file.
Below are the Dockerfiles for the three servers and the docker-compose file:

```dockerfile
FROM maven:3.9.6-eclipse-temurin-17

WORKDIR /app

COPY . .

RUN mvn -pl order-module -am package

CMD ["java", "-jar", "order-module/target/order-1.0.jar"]
```

```dockerfile
FROM maven:3.9.6-eclipse-temurin-17

WORKDIR /app

COPY . .

RUN mvn -pl gateway-module -am package

CMD ["java", "-jar", "gateway-module/target/gateway-1.0.jar"]
```

```dockerfile
FROM maven:3.9.6-eclipse-temurin-17

WORKDIR /app

COPY . .

RUN mvn -pl catalog-module -am package

CMD ["java", "-jar", "catalog-module/target/catalog-1.0.jar"]
```

```yaml
version: '3.8'

services:
  catalog:
    build:
      context: .
      dockerfile: catalog-module/Dockerfile
    ports:
      - "4575:4575"
    networks:
      - microservicesNetwork

  order:
    build:
      context: .
      dockerfile: order-module/Dockerfile
    ports:
      - "3300:3300"
    depends_on:
      - catalog
    networks:
      - microservicesNetwork

  gateway:
    build:
      context: .
      dockerfile: gateway-module/Dockerfile
    ports:
      - "4567:4567"
    depends_on:
      - catalog
      - order
    networks:
      - microservicesNetwork

networks:
  microservicesNetwork:
    driver: bridge
```