“An-Najah National University”
“Computer Engineering Department”

DOS Project Report

SUBMITTED TO:
Dr.Samer Arandi

Preparing By

Shahd Hennawi 12115159
Osamah Abdullah 12111983

❖**The services in our project:**

1) Front end service
2) Order service
3) Catalog service

1) Front-end service:

There are three operations in this server

• **1-Search**: the request is sent to catalog server then catalog return the item
axios. get(http://localhost:3001/search/distributed systems)

• **2-Info**: the request is sent to catalog server then catalog return the info
axios. get(http://catalog:3001/info/1)

• **3-Purchase**: The purchase order is sent to the order server
post(http://localhost:3002/purchase/1)

**** search and info requests can do it it by frontend server:**
We use axios to request the catalog server to give us the information but the request it self it sent to frontend server in port 3000

- axios. get(http://localhost:3000/search/distributed systems)
- axios. get(http://catalog:3000/info/1)

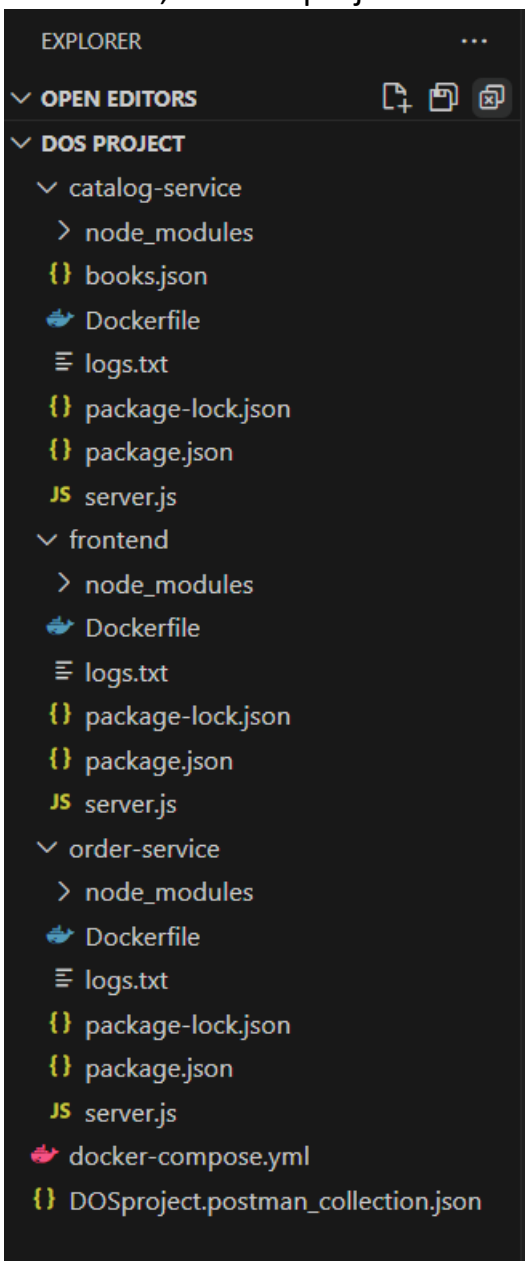
2) Order service:

Receives requests from the front-end and transfers them to catalog server When
it is sent to the catalog, the quantity of stock checked and modified.
http.get(http://catalog:3001/info/1)
axios.patch('http://catalog:3001/info/1)

3) Catalog service:

It receives requests from the order server, adjusts the quantity, sends the
response to the order, and also sends the response to search and info requests.

In the first, this our project Hierarchy :



-we create **3 Services** , 2 for Backend servers , Catalog & Order,
1 for Frontend Client Service.
-we create Dockerfile , to create our containers.
-we create container for each service , so we use docker-compose tool.
docker compose tool: is a tool for definig and running multi-container Docker Applications.
-so we create docker-compose.yml to write the configuration for our compose files

Now, Lets explain each part of our code:

{} books.json ×

catalog-service > {} books.json > {} 1

```
[
  {
    "id": 1,
    "title": "How to get a good grade in DOS in 40 minutes a day",
    "quantity": 5,
    "price": 30,
    "topic": "distributed systems"
  },
  {
    "id": 2,
    "title": "RPCs for Noobs",
    "quantity": 3,
    "price": 20,
    "topic": "distributed systems"
  },
  {
    "id": 3,
    "title": "Xen and the Art of Surviving Undergraduate School",
    "quantity": 7,
    "price": 25,
    "topic": "undergraduate school"
  },
  {
    "id": 4,
    "title": "Cooking for the Impatient Undergrad",
    "quantity": 4,
    "price": 15,
    "topic": "undergraduate school"
  }
]
```

{} books.json Dockerfile ×

catalog-service > Dockerfile

```
FROM node:14
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3001
CMD ["node", "server.js"]
```

inside our package.json file:

```
{} books.json   Dockerfile   {} package-lock.json   {} package.json ×
catalog-service > {} package.json > ...
1  {
2    "name": "catalog-service",
3    "version": "1.0.0",
4    "description": "catalog service",
5    "main": "server.js",
       Debug
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "Osamah Abdullah & Shahd Hennawi",
10   "license": "ISC",
11   "dependencies": {
12     "axios": "^1.7.7",
13     "express": "^4.21.1",
14     "fs-extra": "^11.2.0"
15   }
16 }
17
```

Docker-compose.yml file:

```
{} books.json   Dockerfile   {} package-lock.json   {} package.json   JS server.js   docker-compose.yml ×
docker-compose.yml
1  # docker-compose.yml
2
3  services:
4    frontend:
5      build: ./frontend
6      ports:
7        - "3000:3000"
8      depends_on:
9        - catalog-service
10       - order-service
11
12   catalog-service:
13     build: ./catalog-service
14     ports:
15       - "3001:3001"
16
17   order-service:
18     build: ./order-service
19     ports:
20       - "3002:3002"
21     depends_on:
22       - catalog-service
23
```

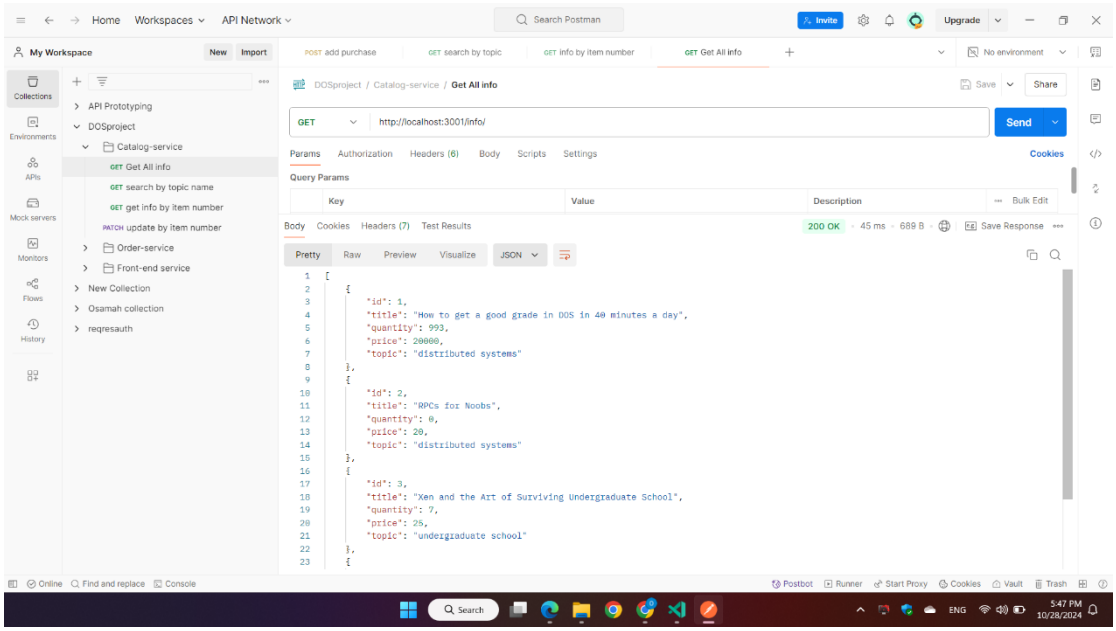
docker-compose up -d -- build

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                                    docker-compose

 => => transferring dockerfile: 153B                                                                          0.0s
 => [frontend internal] load .dockerignore                                                                    0.0s
 => => transferring context: 2B                                                                               0.0s
 => [frontend internal] load build context                                                                    0.3s
 => => transferring context: 47.74kB                                                                          0.3s
 => CACHED [frontend 3/5] COPY package*.json ./                                                               0.0s
 => CACHED [frontend 4/5] RUN npm install                                                                     0.0s
 => CACHED [frontend 5/5] COPY . .                                                                            0.0s
 => [frontend] exporting to image                                                                             0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:a638b585f1cdc97cff1a791ae9498fd7d991588c656361e7f6370ccbe94ec9ba                  0.0s
 => => naming to docker.io/library/dosproject-frontend                                                        0.0s
 => [frontend] resolving provenance for metadata file                                                         0.0s
[+] Running 3/0
 ✓ Container dosproject-catalog-service-1  Running                                                            0.0s
 ✓ Container dosproject-order-service-1    Running                                                            0.0s
 ✓ Container dosproject-frontend-1         Running                                                            0.0s
Attaching to catalog-service-1, frontend-1, order-service-1

 v View in Docker Desktop   o View Config   w Enable Watch
```

Catalog service server Requests and responses:



Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import

POST add purchase | GET search by topic | GET info by item number | GET Get All info | GET search by topic name | No environment

Collections | Environments | APIs | Mock servers | Monitors | Flows | History

- API Prototyping
- DOSproject
  - Catalog-service
    - GET Get All info
    - GET search by topic name
    - GET get info by item number
    - PATCH update by item number
  - Order-service
  - Front-end service
- New Collection
- Osamah collection
- reqresauth

DOSproject / Catalog-service / search by topic name

Save Share

GET http://localhost:3001/search/distributed systems Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results 200 OK 12 ms 454 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "id": 1,
        "title": "How to get a good grade in DOS in 40 minutes a day",
        "quantity": 993,
        "price": 20000,
        "topic": "distributed systems"
    },
    {
        "id": 2,
        "title": "RPCs for Noobs",
        "quantity": 0,
        "price": 20,
        "topic": "distributed systems"
    }
]
```


Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

Search ENG 5:47 PM 10/28/2024

Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import

POST add purchase | GET search by topic | GET info by item number | GET Get All info | GET search by topic name | No environment

Collections | Environments | APIs | Mock servers | Monitors | Flows | History

- API Prototyping
- DOSproject
  - Catalog-service
    - GET Get All info
    - GET search by topic name
    - GET get info by item number
    - PATCH update by item number
  - Order-service
  - Front-end service
- New Collection
- Osamah collection
- reqresauth

DOSproject / Catalog-service / search by topic name

Save Share

GET http://localhost:3001/search/distributed systemssssssssssssss Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

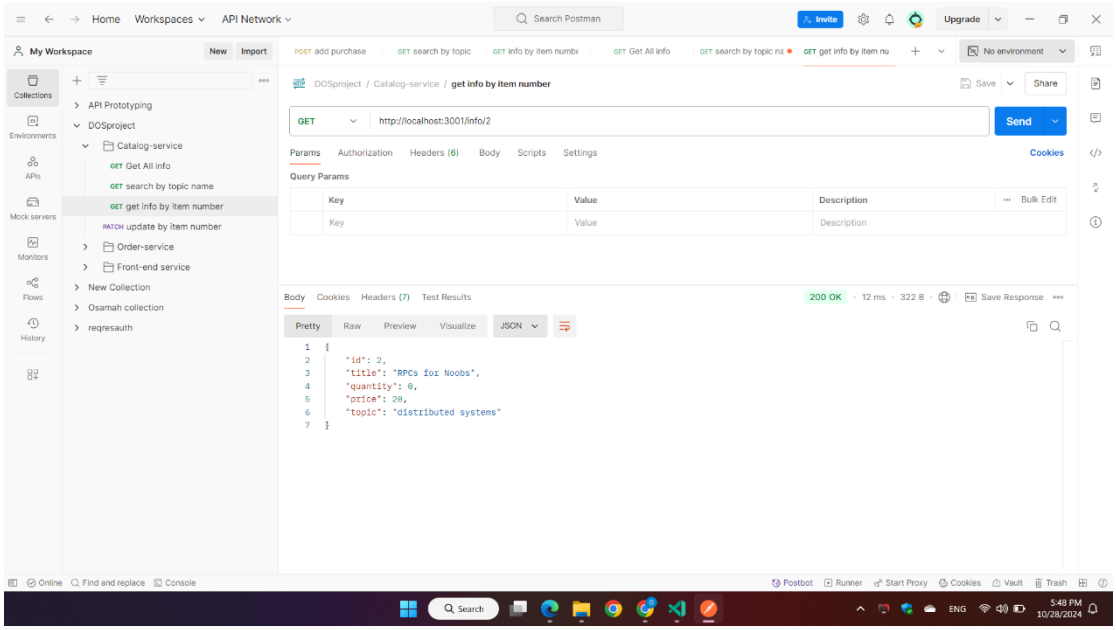
Body Cookies Headers (7) Test Results 404 Not Found 11 ms 271 B Save Response

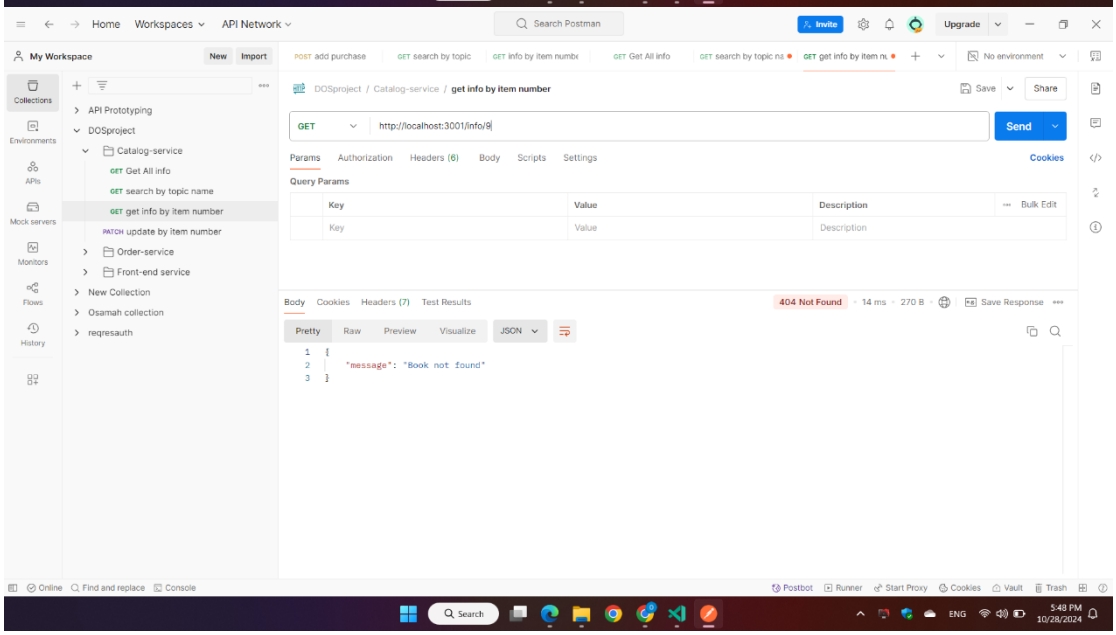
Pretty Raw Preview Visualize JSON

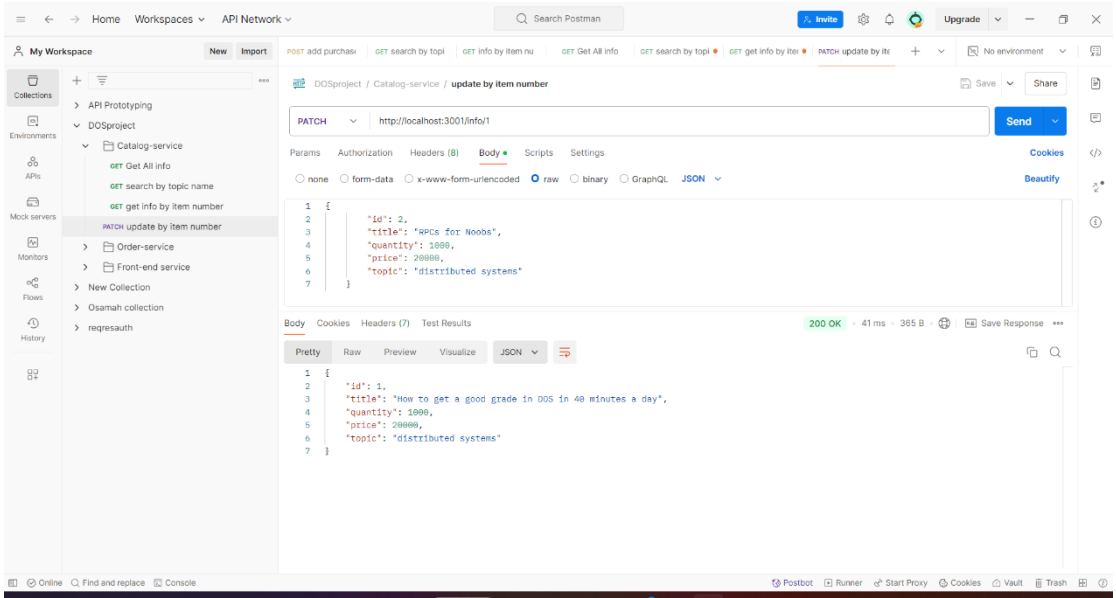

```
[
    "message": "Topic not found"
]
```

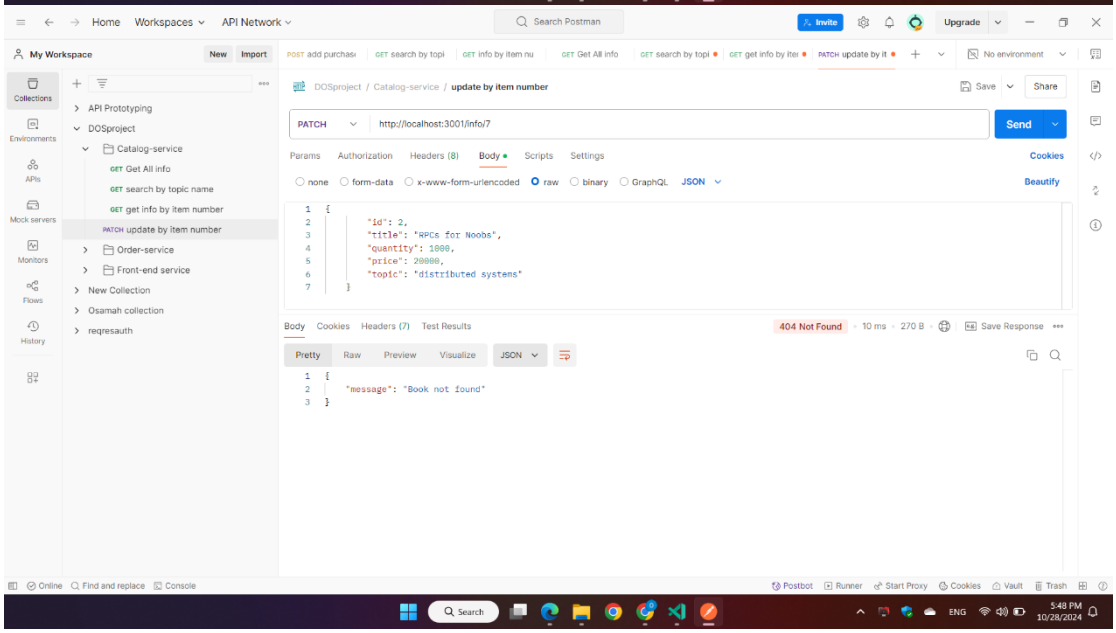

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

Search ENG 5:48 PM 10/28/2024

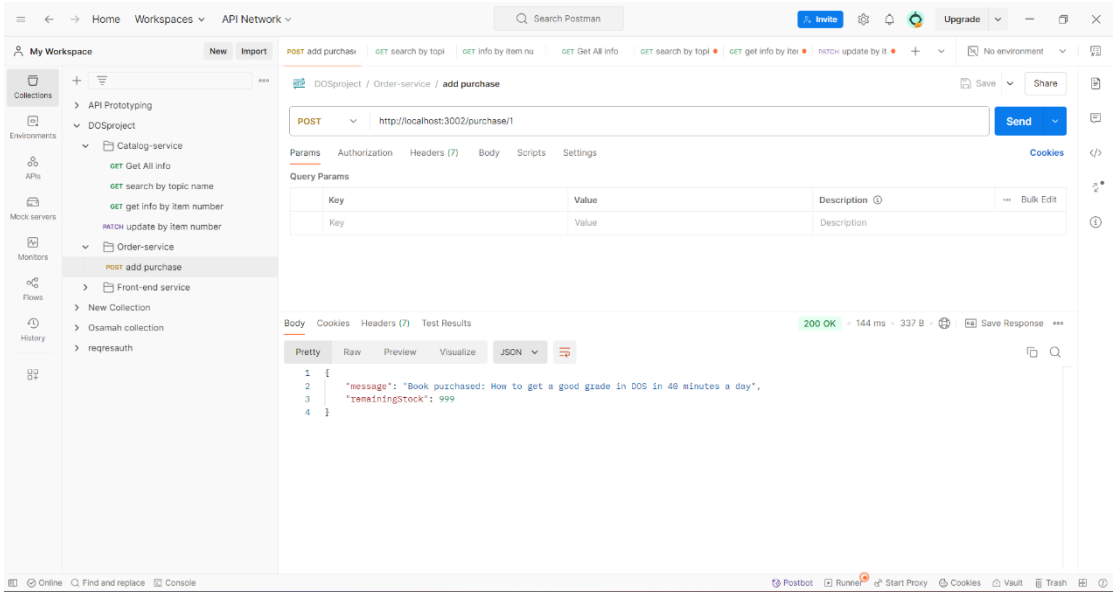

DOSproject / Catalog-service / get info by item number
GET
http://localhost:3001/info/2
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (7)
Test Results
200 OK
12 ms
322 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"id": 2,
"title": "RPCs for Noobs",
"quantity": 0,
"price": 20,
"topic": "distributed systems"
}

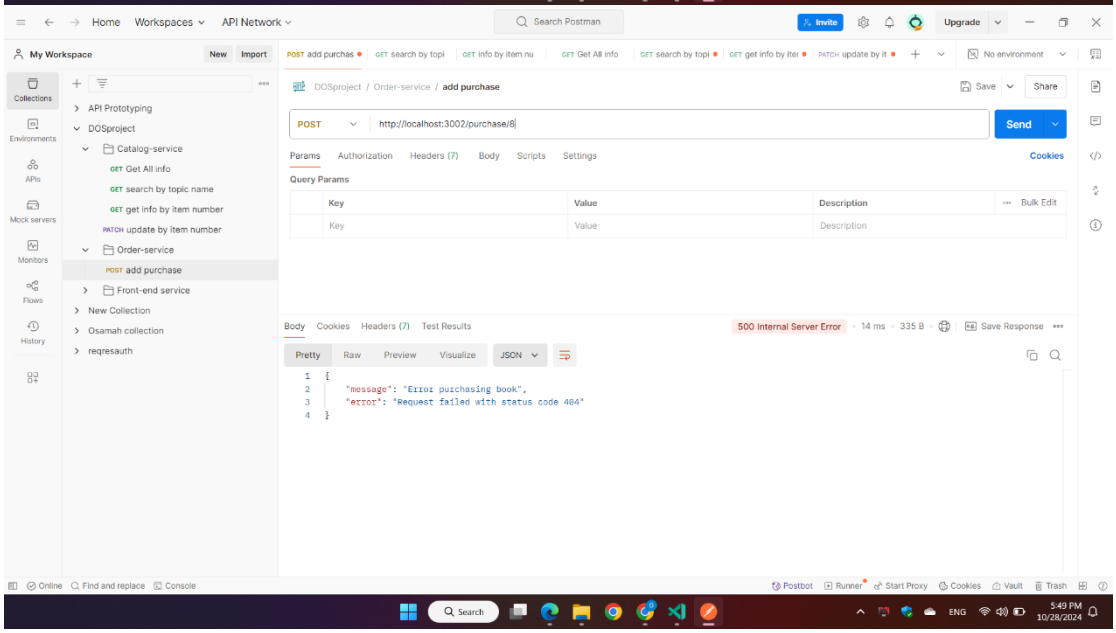

DOSproject / Catalog-service / get info by item number
GET
http://localhost:3001/info/9
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (7)
Test Results
404 Not Found
14 ms
270 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Book not found"
}


DOSproject / Catalog-service / update by item number
PATCH
http://localhost:3001/info/1
Send
Body
raw
JSON
{
"id": 2,
"title": "RPCs for Noobs",
"quantity": 1000,
"price": 20000,
"topic": "distributed systems"
}
200 OK
41 ms
365 B
{
"id": 1,
"title": "How to get a good grade in DOS in 40 minutes a day",
"quantity": 1000,
"price": 20000,
"topic": "distributed systems"
}

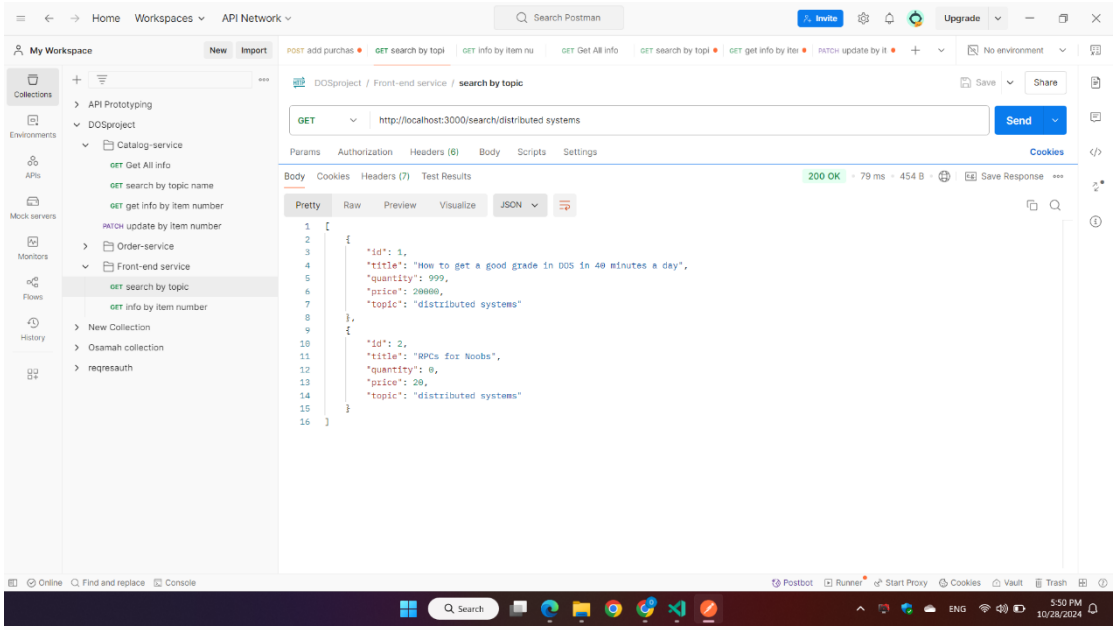

DOSproject / Catalog-service / update by item number
PATCH
http://localhost:3001/info/7
Send
Body
raw
JSON
{
"id": 2,
"title": "RPCs for Noobs",
"quantity": 1000,
"price": 20000,
"topic": "distributed systems"
}
404 Not Found
10 ms
270 B
{
"message": "Book not found"
}

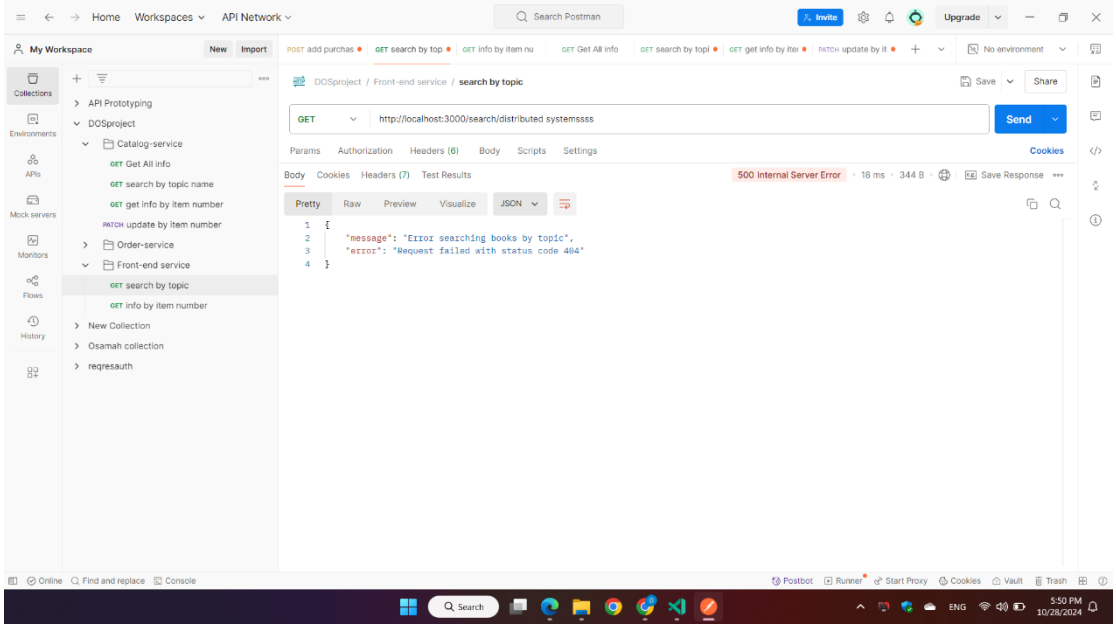
Order-service server requests and responses:

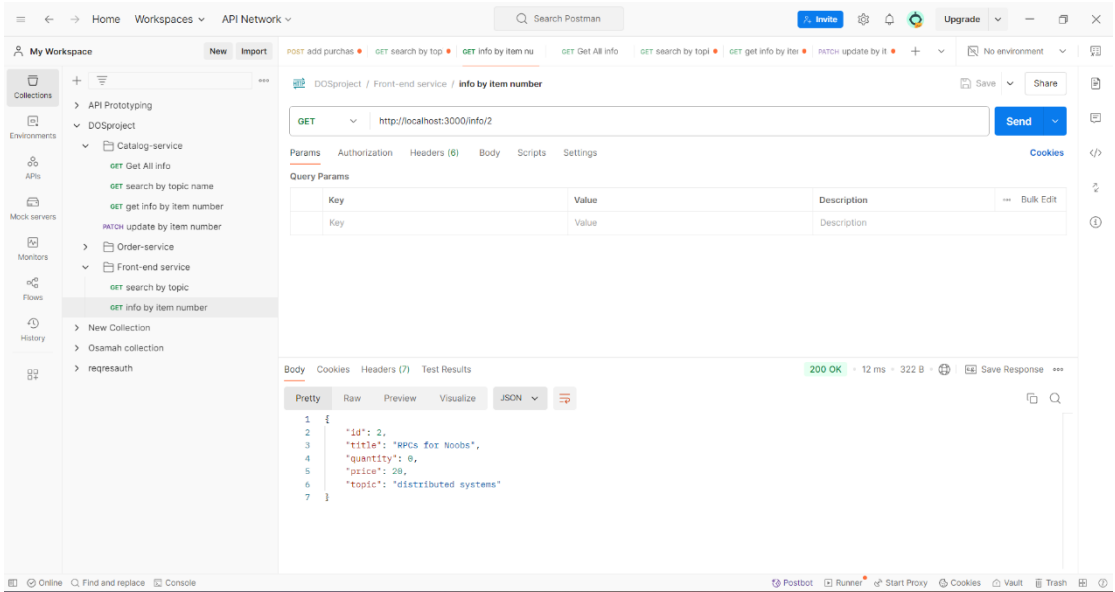


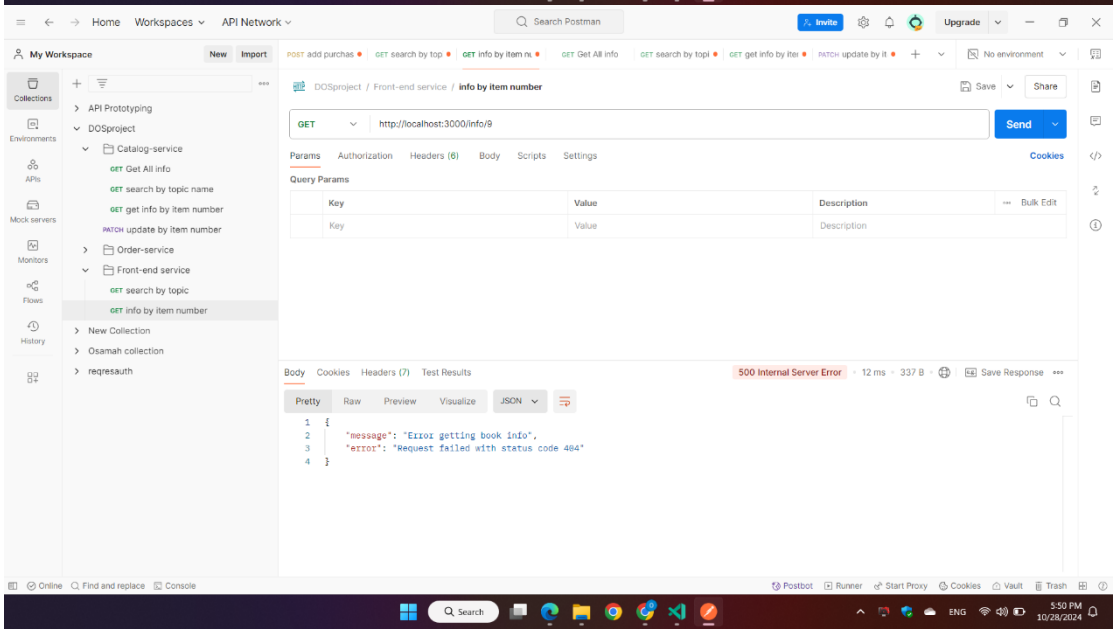


Frontend server requests and responses:









We use Node.js (Java Runtime Enviroment) and Express.js as framework because it provides powerful methods that are asynchronous and lightweight also it is scalable (in Node.js threadpool by default 4 threads but we can make it up to 1024 if we want more than that we should change system kernel) they are backed by a large community and require specific utilization and expertise.