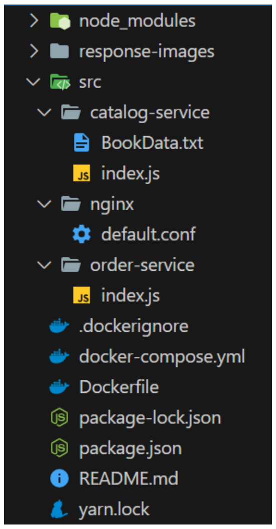
DOS Project Part 1

May Shelbayeh

Dema Khalili

The hierarchy for the project is structured as follows:



```
> node_modules
> response-images
v src
  v catalog-service
      BookData.txt
      index.js
  v nginx
      default.conf
  v order-service
      index.js
  .dockerignore
  docker-compose.yml
  Dockerfile
  package-lock.json
  package.json
  README.md
  yarn.lock
```

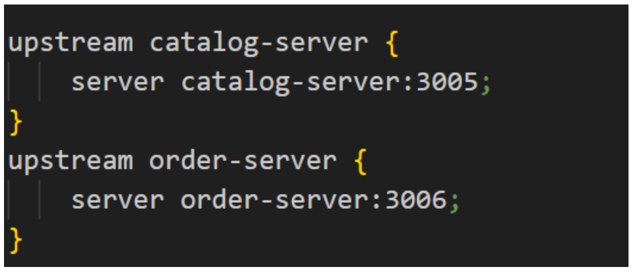
We create 2 services for catalog & order.

Write Dockerfile to create these containers, then for running the two containers we need a Docker compose file, which is a tool used for defining and running multi-container Docker applications, it's a yaml file.

For managing incoming request from client to backend servers using nginx, so it handles HTTP request then send it to right server.

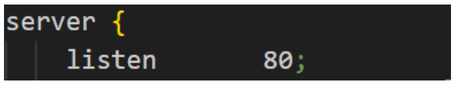
Giving the nginx configuration from default.config file.

default.config:



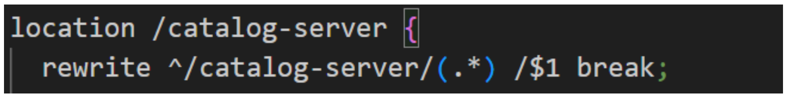
```
upstream catalog-server {
        server catalog-server:3005;
}
upstream order-server {
        server order-server:3006;
}
```

Two upstream blocks define the backend server, the first for catalog-server contains a single server listening at port 3005, the last is for order-service also with single server listening at port 3006.



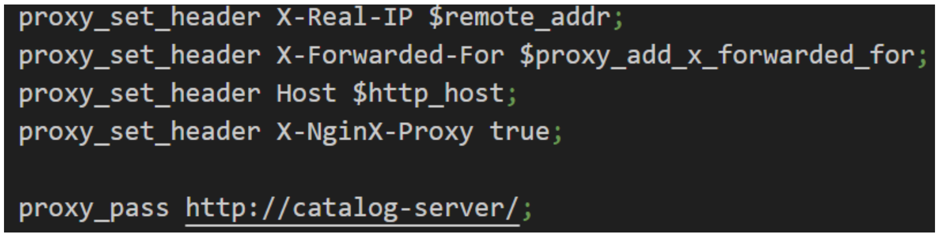
```
server {
    listen        80;
```

define a nginx port.



```
location /catalog-server {
  rewrite ^/catalog-server/(.*) /$1 break;
```

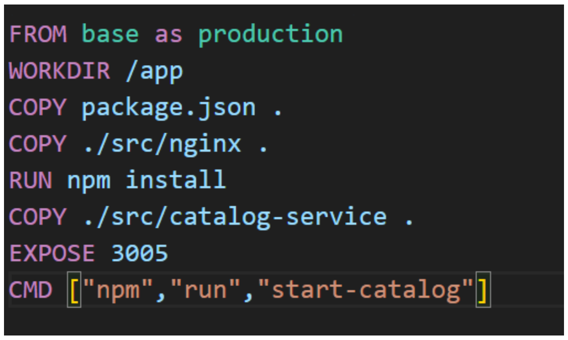
This location block handle the request started with /catalog-server, rewrite removes /catalog-server from the request URL.



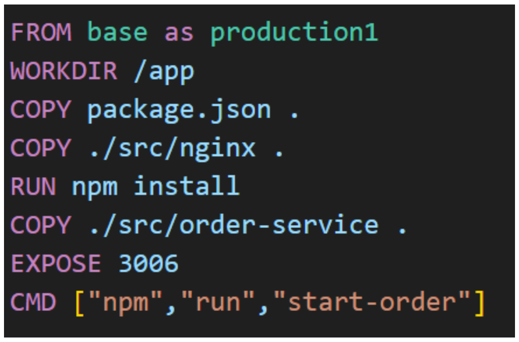
```
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header Host $http_host;
 proxy_set_header X-NginX-Proxy true;

 proxy_pass http://catalog-server/;
```

Proxy-pass forward the request to the ‘catalog-service’ upstream.
the same thing for order-service.

Dokerfile:



```
FROM base as production
WORKDIR /app
COPY package.json .
COPY ./src/nginx .
RUN npm install
COPY ./src/catalog-service .
EXPOSE 3005
CMD ["npm","run","start-catalog"]
```

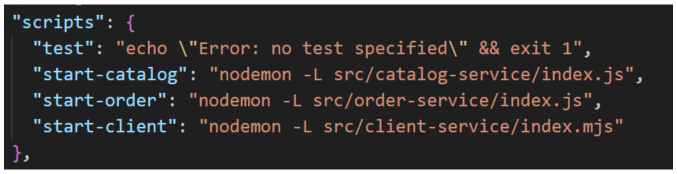


```
FROM base as production1
WORKDIR /app
COPY package.json .
COPY ./src/nginx .
RUN npm install
COPY ./src/order-service .
EXPOSE 3006
CMD ["npm","run","start-order"]
```

WORKDIR /app: refers to the directory which our files for containers will be existing in.

COPY package.json .: to install all dependencies we needed we run the containers.

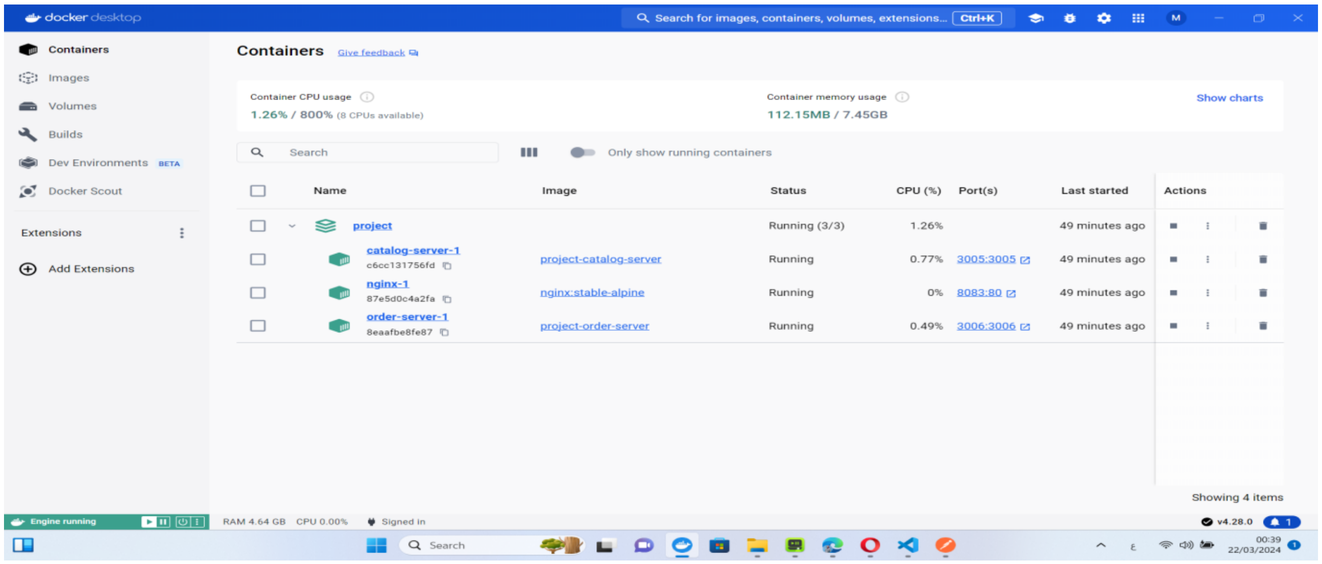
Inside package.json:



```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start-catalog": "nodemon -L src/catalog-service/index.js",
  "start-order": "nodemon -L src/order-service/index.js",
  "start-client": "nodemon -L src/client-service/index.mjs"
},
```

COPY ./src/nginx .: copy nginx config inside containers

Expose #port: the port that the server running on it.



Containers Give feedback

Container CPU usage
1.26% / 800% (8 CPUs available)

Container memory usage
112.15MB / 7.45GB

Only show running containers

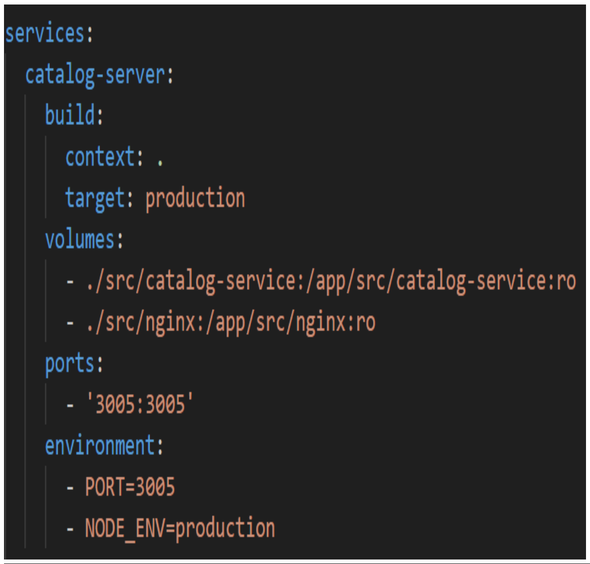
Name	Image	Status	CPU (%)	Port(s)	Last started
project		Running (3/3)	1.26%		49 minutes ago
catalog-server-1 c6cc131756fd	project-catalog-server	Running	0.77%	3005:3005	49 minutes ago
nginx-1 87e5d0c4a2fa	nginx:stable-alpine	Running	0%	8083:80	49 minutes ago
order-server-1 8eaafbe8fe87	project-order-server	Running	0.49%	3006:3006	49 minutes ago

Showing 4 items

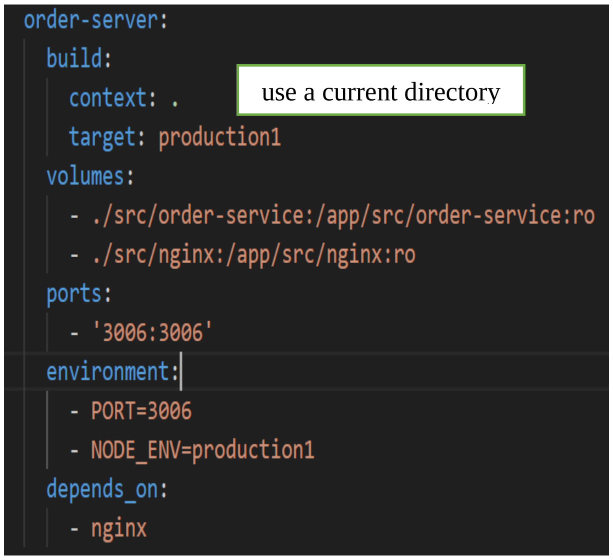
Engine running   RAM 4.64 GB   CPU 0.00%   Signed in   v4.28.0

1- three containers in Docker 'catalog-service, nginx, order-service'

docker-compose.yml file: define three services

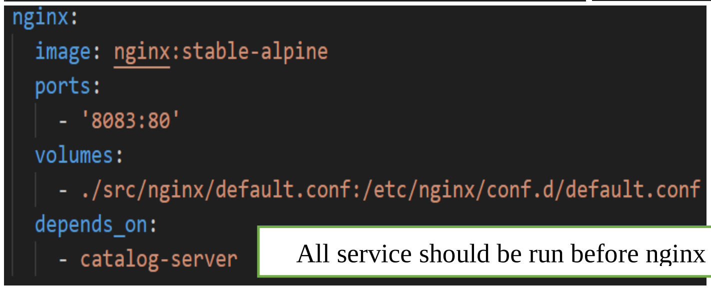


```yaml
services:
  catalog-server:
    build:
      context: .
      target: production
    volumes:
      - ./src/catalog-service:/app/src/catalog-service:ro
      - ./src/nginx:/app/src/nginx:ro
    ports:
      - '3005:3005'
    environment:
      - PORT=3005
      - NODE_ENV=production
```



```yaml
  order-server:
    build:
      context: .
      target: production1
    volumes:
      - ./src/order-service:/app/src/order-service:ro
      - ./src/nginx:/app/src/nginx:ro
    ports:
      - '3006:3006'
    environment:
      - PORT=3006
      - NODE_ENV=production1
    depends_on:
      - nginx
```

use a current directory



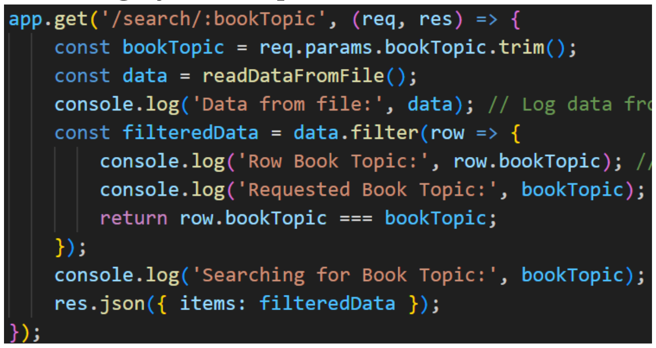
```yaml
  nginx:
    image: nginx:stable-alpine
    ports:
      - '8083:80'
    volumes:
      - ./src/nginx/default.conf:/etc/nginx/conf.d/default.conf
    depends_on:
      - catalog-server
```

All service should be run before nginx

After services 'keyword' begin to define each service by written service name, then we can write **docker run compose up** command for building our containers.

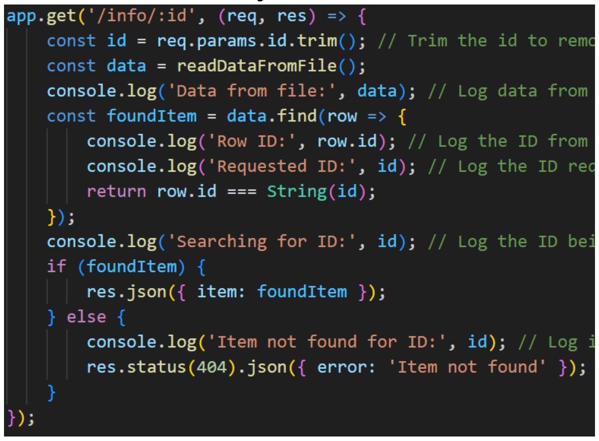
Catalog-service:

- Searching by book topic. localhost:3005/search/:bookTopic



```
app.get('/search/:bookTopic', (req, res) => {
    const bookTopic = req.params.bookTopic.trim();
    const data = readDataFromFile();
    console.log('Data from file:', data); // Log data fro
    const filteredData = data.filter(row => {
        console.log('Row Book Topic:', row.bookTopic); //
        console.log('Requested Book Topic:', bookTopic);
        return row.bookTopic === bookTopic;
    });
    console.log('Searching for Book Topic:', bookTopic);
    res.json({ items: filteredData });
});
```

- Book information by book id. localhost:3005/info/:id

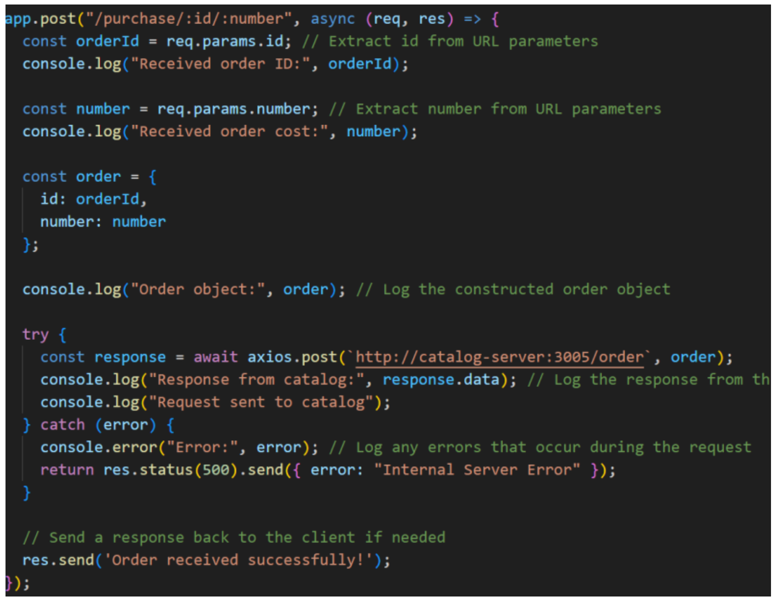


```
app.get('/info/:id', (req, res) => {
    const id = req.params.id.trim(); // Trim the id to remo
    const data = readDataFromFile();
    console.log('Data from file:', data); // Log data from
    const foundItem = data.find(row => {
        console.log('Row ID:', row.id); // Log the ID from
        console.log('Requested ID:', id); // Log the ID rec
        return row.id === String(id);
    });
    console.log('Searching for ID:', id); // Log the ID bei
    if (foundItem) {
        res.json({ item: foundItem });
    } else {
        console.log('Item not found for ID:', id); // Log i
        res.status(404).json({ error: 'Item not found' });
    }
});
```

- For purchase it will be a request from order service

order-service:

- Purchase:
  Received book id & number from client side.

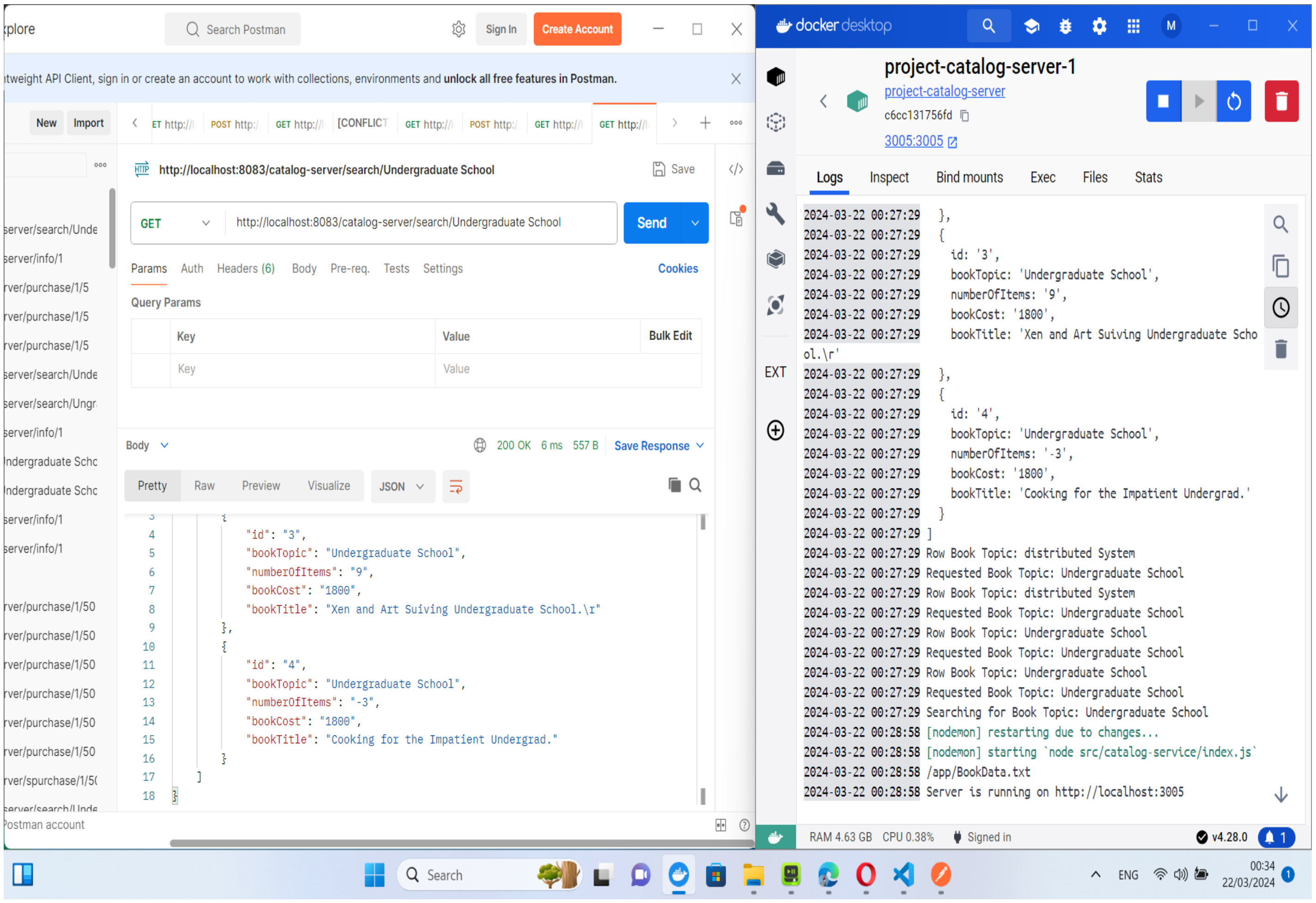


```
app.post("/purchase/:id/:number", async (req, res) => {
  const orderId = req.params.id; // Extract id from URL parameters
  console.log("Received order ID:", orderId);

  const number = req.params.number; // Extract number from URL parameters
  console.log("Received order cost:", number);

  const order = {
    id: orderId,
    number: number
  };

  console.log("Order object:", order); // Log the constructed order object

  try {
    const response = await axios.post(`http://catalog-server:3005/order`, order);
    console.log("Response from catalog:", response.data); // Log the response from th
    console.log("Request sent to catalog");
  } catch (error) {
    console.error("Error:", error); // Log any errors that occur during the request
    return res.status(500).send({ error: "Internal Server Error" });
  }

  // Send a response back to the client if needed
  res.send('Order received successfully!');
});
```

Then forward the request to catalog-service using axios request, if:

- Not existing item will return “Item not found for ID”.
- Out of stock: “Not enough items in stock for ID”.
- Exist: make the order & update the count of book in file

Search request



http://localhost:8083/catalog-server/search/Undergraduate School

GET http://localhost:8083/catalog-server/search/Undergraduate School

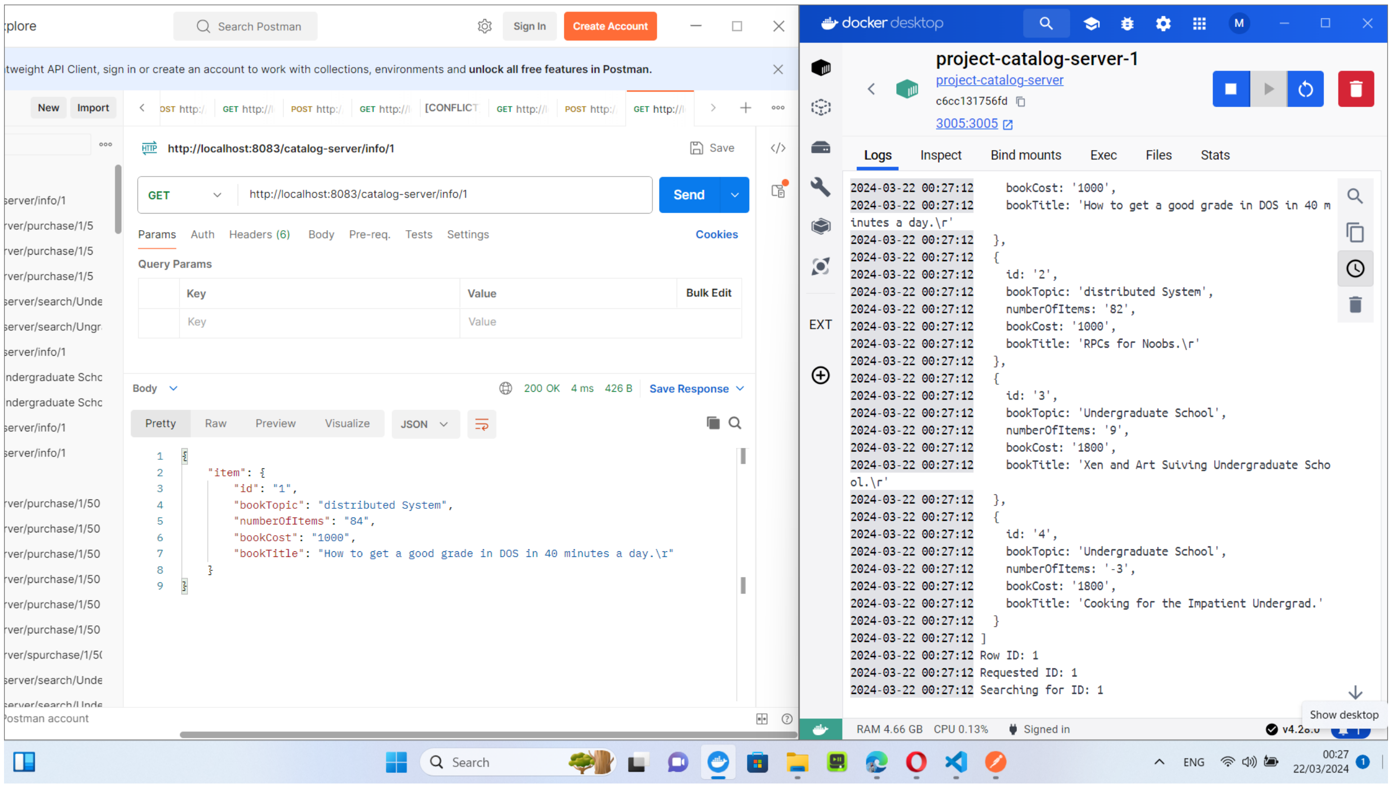
200 OK 6 ms 557 B

```
        {
            "id": "3",
            "bookTopic": "Undergraduate School",
            "numberOfItems": "9",
            "bookCost": "1800",
            "bookTitle": "Xen and Art Suiving Undergraduate School.\r"
        },
        {
            "id": "4",
            "bookTopic": "Undergraduate School",
            "numberOfItems": "-3",
            "bookCost": "1800",
            "bookTitle": "Cooking for the Impatient Undergrad."
        }
    ]
}
```

project-catalog-server-1
project-catalog-server
c6cc131756fd
3005:3005

```
2024-03-22 00:27:29   },
2024-03-22 00:27:29   {
2024-03-22 00:27:29     id: '3',
2024-03-22 00:27:29     bookTopic: 'Undergraduate School',
2024-03-22 00:27:29     numberOfItems: '9',
2024-03-22 00:27:29     bookCost: '1800',
2024-03-22 00:27:29     bookTitle: 'Xen and Art Suiving Undergraduate Scho
ol.\r'
2024-03-22 00:27:29   },
2024-03-22 00:27:29   {
2024-03-22 00:27:29     id: '4',
2024-03-22 00:27:29     bookTopic: 'Undergraduate School',
2024-03-22 00:27:29     numberOfItems: '-3',
2024-03-22 00:27:29     bookCost: '1800',
2024-03-22 00:27:29     bookTitle: 'Cooking for the Impatient Undergrad.'
2024-03-22 00:27:29   }
2024-03-22 00:27:29 ]
2024-03-22 00:27:29 Row Book Topic: distributed System
2024-03-22 00:27:29 Requested Book Topic: Undergraduate School
2024-03-22 00:27:29 Row Book Topic: distributed System
2024-03-22 00:27:29 Requested Book Topic: Undergraduate School
2024-03-22 00:27:29 Row Book Topic: Undergraduate School
2024-03-22 00:27:29 Requested Book Topic: Undergraduate School
2024-03-22 00:27:29 Row Book Topic: Undergraduate School
2024-03-22 00:27:29 Requested Book Topic: Undergraduate School
2024-03-22 00:27:29 Searching for Book Topic: Undergraduate School
2024-03-22 00:28:58 [nodemon] restarting due to changes...
2024-03-22 00:28:58 [nodemon] starting `node src/catalog-service/index.js`
2024-03-22 00:28:58 /app/BookData.txt
2024-03-22 00:28:58 Server is running on http://localhost:3005
```

Info request



http://localhost:8083/catalog-server/info/1

GET http://localhost:8083/catalog-server/info/1

Body — 200 OK 4 ms 426 B

```
{
    "item": {
        "id": "1",
        "bookTopic": "distributed System",
        "numberOfItems": "84",
        "bookCost": "1000",
        "bookTitle": "How to get a good grade in DOS in 40 minutes a day.\r"
    }
}
```

project-catalog-server-1

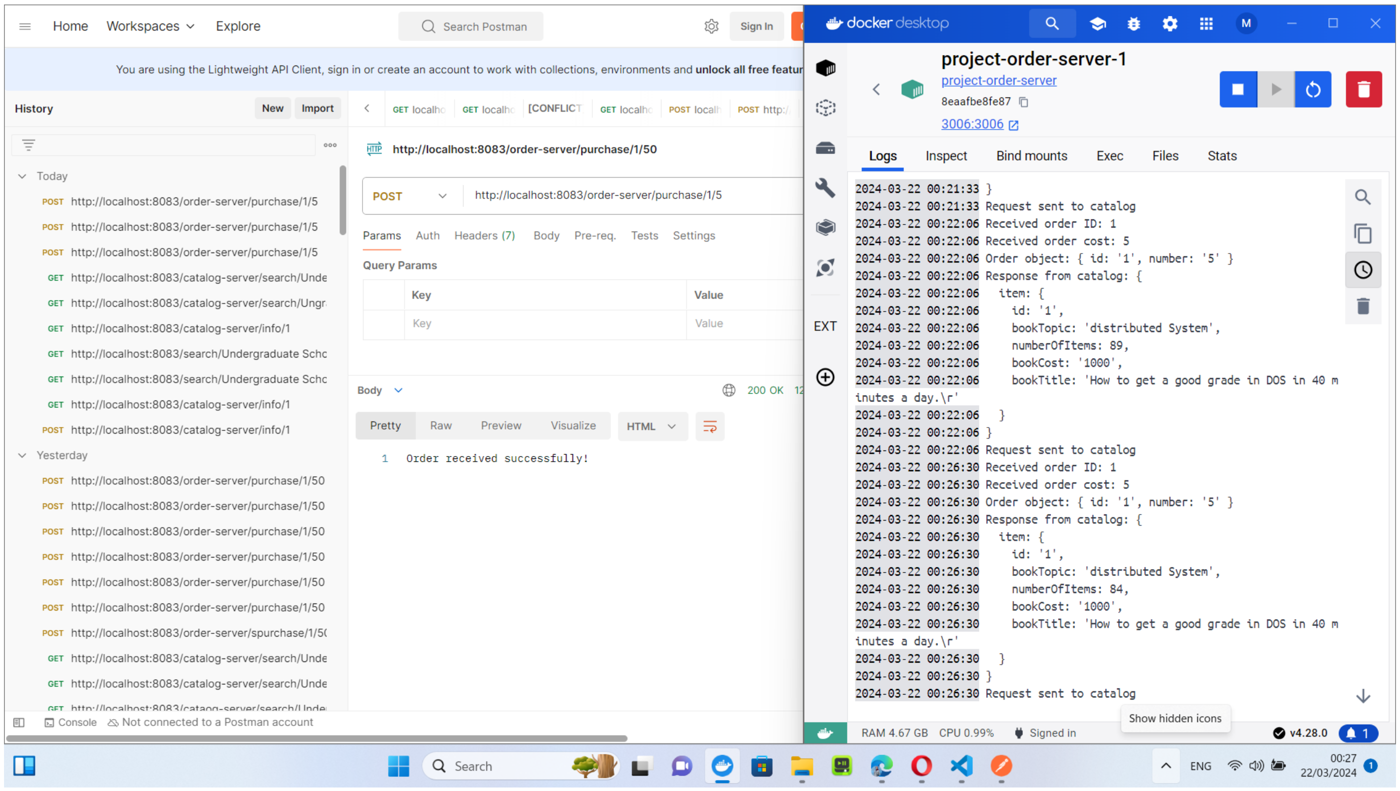
project-catalog-server

c6cc131756fd

3005:3005

```
2024-03-22 00:27:12     bookCost: '1000',
2024-03-22 00:27:12     bookTitle: 'How to get a good grade in DOS in 40 m
inutes a day.\r'
2024-03-22 00:27:12   },
2024-03-22 00:27:12   {
2024-03-22 00:27:12     id: '2',
2024-03-22 00:27:12     bookTopic: 'distributed System',
2024-03-22 00:27:12     numberOfItems: '82',
2024-03-22 00:27:12     bookCost: '1000',
2024-03-22 00:27:12     bookTitle: 'RPCs for Noobs.\r'
2024-03-22 00:27:12   },
2024-03-22 00:27:12   {
2024-03-22 00:27:12     id: '3',
2024-03-22 00:27:12     bookTopic: 'Undergraduate School',
2024-03-22 00:27:12     numberOfItems: '9',
2024-03-22 00:27:12     bookCost: '1800',
2024-03-22 00:27:12     bookTitle: 'Xen and Art Suiving Undergraduate Scho
ol.\r'
2024-03-22 00:27:12   },
2024-03-22 00:27:12   {
2024-03-22 00:27:12     id: '4',
2024-03-22 00:27:12     bookTopic: 'Undergraduate School',
2024-03-22 00:27:12     numberOfItems: '-3',
2024-03-22 00:27:12     bookCost: '1800',
2024-03-22 00:27:12     bookTitle: 'Cooking for the Impatient Undergrad.'
2024-03-22 00:27:12   }
2024-03-22 00:27:12 ]
2024-03-22 00:27:12 Row ID: 1
2024-03-22 00:27:12 Requested ID: 1
2024-03-22 00:27:12 Searching for ID: 1
```

Purchase request



History

Today
- POST http://localhost:8083/order-server/purchase/1/5
- POST http://localhost:8083/order-server/purchase/1/5
- POST http://localhost:8083/order-server/purchase/1/5
- GET http://localhost:8083/catalog-server/search/Unde
- GET http://localhost:8083/catalog-server/search/Ungr
- GET http://localhost:8083/catalog-server/info/1
- GET http://localhost:8083/search/Undergraduate Schc
- GET http://localhost:8083/search/Undergraduate Schc
- GET http://localhost:8083/catalog-server/info/1
- POST http://localhost:8083/catalog-server/info/1

Yesterday
- POST http://localhost:8083/order-server/purchase/1/50
- POST http://localhost:8083/order-server/purchase/1/50
- POST http://localhost:8083/order-server/purchase/1/50
- POST http://localhost:8083/order-server/purchase/1/50
- POST http://localhost:8083/order-server/purchase/1/50
- POST http://localhost:8083/order-server/purchase/1/50
- POST http://localhost:8083/order-server/spurchase/1/5(
- GET http://localhost:8083/catalog-server/search/Unde
- GET http://localhost:8083/catalog-server/search/Unde

http://localhost:8083/order-server/purchase/1/50

POST http://localhost:8083/order-server/purchase/1/5

Body — 200 OK

```
Order received successfully!
```

project-order-server-1

project-order-server

8eaafbe8fe87

3006:3006

```
2024-03-22 00:21:33 }
2024-03-22 00:21:33 Request sent to catalog
2024-03-22 00:22:06 Received order ID: 1
2024-03-22 00:22:06 Received order cost: 5
2024-03-22 00:22:06 Order object: { id: '1', number: '5' }
2024-03-22 00:22:06 Response from catalog: {
2024-03-22 00:22:06   item: {
2024-03-22 00:22:06     id: '1',
2024-03-22 00:22:06     bookTopic: 'distributed System',
2024-03-22 00:22:06     numberOfItems: 89,
2024-03-22 00:22:06     bookCost: '1000',
2024-03-22 00:22:06     bookTitle: 'How to get a good grade in DOS in 40 m
inutes a day.\r'
2024-03-22 00:22:06   }
2024-03-22 00:22:06 }
2024-03-22 00:22:06 Request sent to catalog
2024-03-22 00:26:30 Received order ID: 1
2024-03-22 00:26:30 Received order cost: 5
2024-03-22 00:26:30 Order object: { id: '1', number: '5' }
2024-03-22 00:26:30 Response from catalog: {
2024-03-22 00:26:30   item: {
2024-03-22 00:26:30     id: '1',
2024-03-22 00:26:30     bookTopic: 'distributed System',
2024-03-22 00:26:30     numberOfItems: 84,
2024-03-22 00:26:30     bookCost: '1000',
2024-03-22 00:26:30     bookTitle: 'How to get a good grade in DOS in 40 m
inutes a day.\r'
2024-03-22 00:26:30   }
2024-03-22 00:26:30 }
2024-03-22 00:26:30 Request sent to catalog
```