

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
Факультет инфокоммуникационных технологий

**ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 4**  
по теме: Развертывание  
по дисциплине: Бэк-энд разработка

Специальность:

09.03.03 Мобильные и сетевые технологии

Выполнила:

Малютина Е.А., К33402

Проверил:

Добряков Д.И.

Санкт-Петербург,

2024

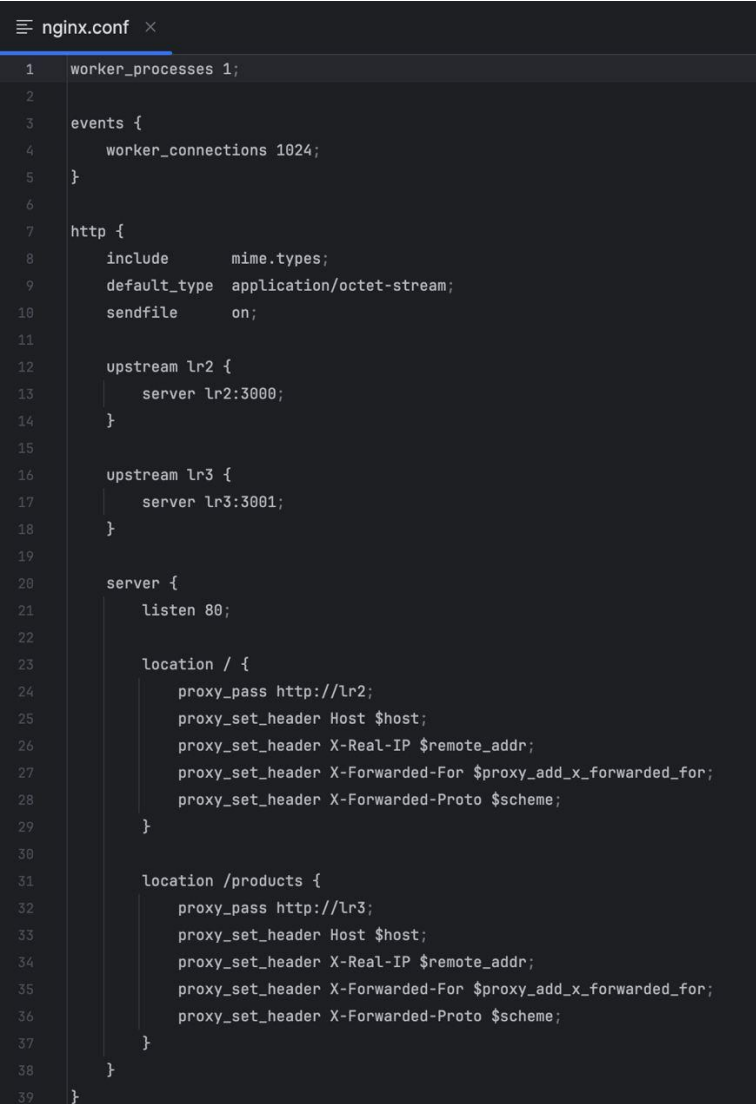
## Задание.

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения, а также настроить общение микросервисов между собой посредством RabbitMQ. Делать это можно как с помощью docker-compose так и с помощью docker swarm.

## Ход работы.

Была произведена замена RabbitMQ на Nginx.

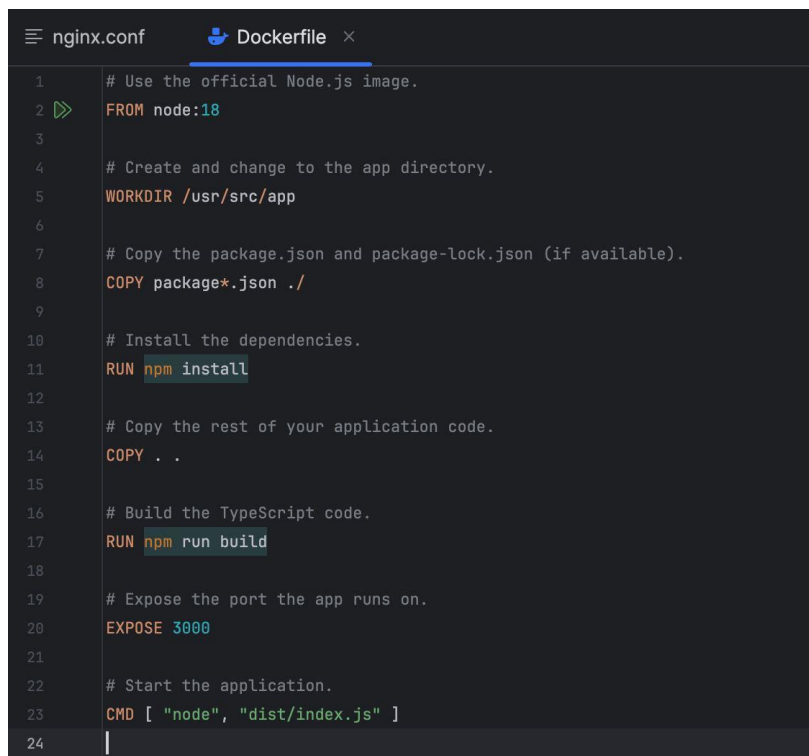
Для настройки nginx был сконфигурирован файл nginx.conf, который показан ниже (Рисунок 1). Данный файл отвечает за настройку и проксирование запросов (переотправку по указанному пути).



```
1 worker_processes 1;
2
3 events {
4     worker_connections 1024;
5 }
6
7 http {
8     include mime.types;
9     default_type application/octet-stream;
10    sendfile on;
11
12    upstream lr2 {
13        server lr2:3000;
14    }
15
16    upstream lr3 {
17        server lr3:3001;
18    }
19
20    server {
21        listen 80;
22
23        location / {
24            proxy_pass http://lr2;
25            proxy_set_header Host $host;
26            proxy_set_header X-Real-IP $remote_addr;
27            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
28            proxy_set_header X-Forwarded-Proto $scheme;
29        }
30
31        location /products {
32            proxy_pass http://lr3;
33            proxy_set_header Host $host;
34            proxy_set_header X-Real-IP $remote_addr;
35            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
36            proxy_set_header X-Forwarded-Proto $scheme;
37        }
38    }
39 }
```

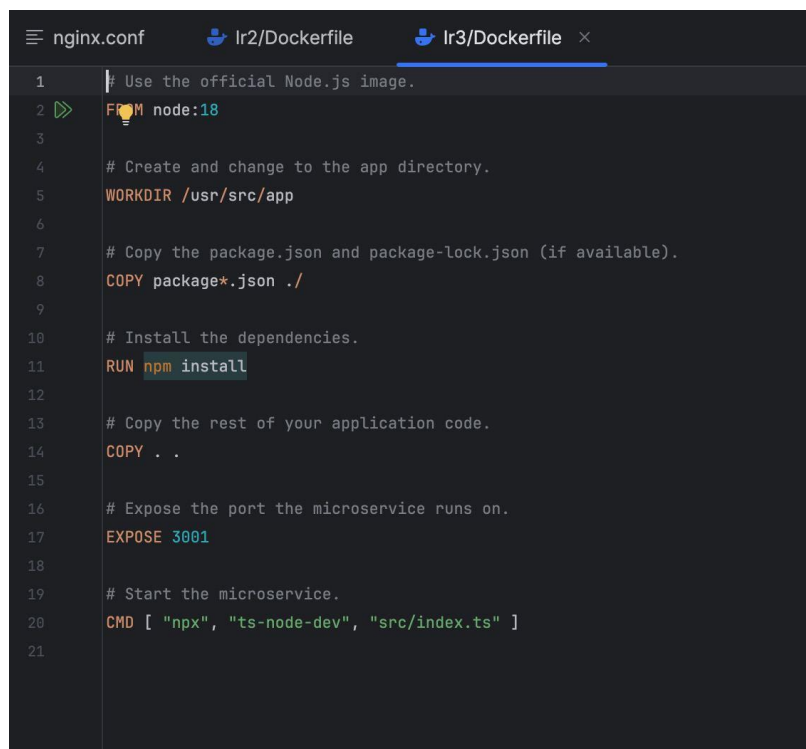
Рисунок 1 – nginx.conf

Также в ранее реализованных лабораторных 2 и 3 были написаны Dockerfile-ы к каждому сервису, как и к nginx с базой данных PostgreSQL, и затем объединены в единую сеть и описаны в docker-compose.yml (Рисунки 2-5).



```
nginx.conf Dockerfile x
1 # Use the official Node.js image.
2 FROM node:18
3
4 # Create and change to the app directory.
5 WORKDIR /usr/src/app
6
7 # Copy the package.json and package-lock.json (if available).
8 COPY package*.json ./
9
10 # Install the dependencies.
11 RUN npm install
12
13 # Copy the rest of your application code.
14 COPY . .
15
16 # Build the TypeScript code.
17 RUN npm run build
18
19 # Expose the port the app runs on.
20 EXPOSE 3000
21
22 # Start the application.
23 CMD [ "node", "dist/index.js" ]
24
```

Рисунок 2 – lr2 Dockerfile



```
nginx.conf lr2/Dockerfile lr3/Dockerfile x
1 # Use the official Node.js image.
2 FROM node:18
3
4 # Create and change to the app directory.
5 WORKDIR /usr/src/app
6
7 # Copy the package.json and package-lock.json (if available).
8 COPY package*.json ./
9
10 # Install the dependencies.
11 RUN npm install
12
13 # Copy the rest of your application code.
14 COPY . .
15
16 # Expose the port the microservice runs on.
17 EXPOSE 3001
18
19 # Start the microservice.
20 CMD [ "npx", "ts-node-dev", "src/index.ts" ]
21
```

Рисунок 3 – lr3 Dockerfile

```

nginx.conf  Ir2/Dockerfile  Ir3/Dockerfile  Dockerfile.nginx

i Plugins supporting *.nginx files found.

1 FROM nginx:alpine
2 COPY nginx.conf /etc/nginx/nginx.conf
3

```

Рисунок 4 – nginx Dockerfile

```

docker-compose.yml  docker-compose.yml

1 version: '3.9'
2
3 services:
4   lr2:
5     container_name: lr2
6     build:
7       context: ./lr2
8       dockerfile: Dockerfile
9     ports:
10      - "3000"
11     depends_on:
12      - postgres
13     environment:
14      - DATABASE_URL=postgres://postgres:postgres@postgres:5432/db
15     volumes:
16      - ./lr2:/usr/src/app:cached
17      - /usr/src/app/node_modules
18     networks:
19      - my-network
20
21   lr3:
22     container_name: lr3
23     build:
24       context: ./lr3
25       dockerfile: Dockerfile
26     ports:
27      - "3001"
28     depends_on:
29      - postgres
30     environment:
31      - DATABASE_URL=postgres://postgres:postgres@postgres:5432/db
32     volumes:
33      - ./lr3:/usr/src/app:cached
34      - /usr/src/app/node_modules
35     networks:
36      - my-network
37
38   nginx:
39     container_name: nginx
40     image: nginx:alpine
41     ports:
42      - "80:80"
43     depends_on:
44      - lr2
45      - lr3
46     volumes:
47      - ./nginx.conf:/etc/nginx/nginx.conf
48     networks:
49      - my-network
50
51   postgres:
52     container_name: postgres
53     image: postgres
54     environment:
55      - POSTGRES_USER: postgres
56      - POSTGRES_PASSWORD: postgres
57      - POSTGRES_DB: db
58     volumes:
59      - pgdata:/var/lib/postgresql/data
60     networks:
61      - my-network
62
63   networks:
64     my-network:
65       driver: bridge
66
67   volumes:
68     pgdata:
69
Document 1/1  version:  Document 1/1  version:

```

Рисунок 5 – docker-compose.yml

Далее были запущены контейнеры и протестирована работоспособность (Рисунки 6-11).
















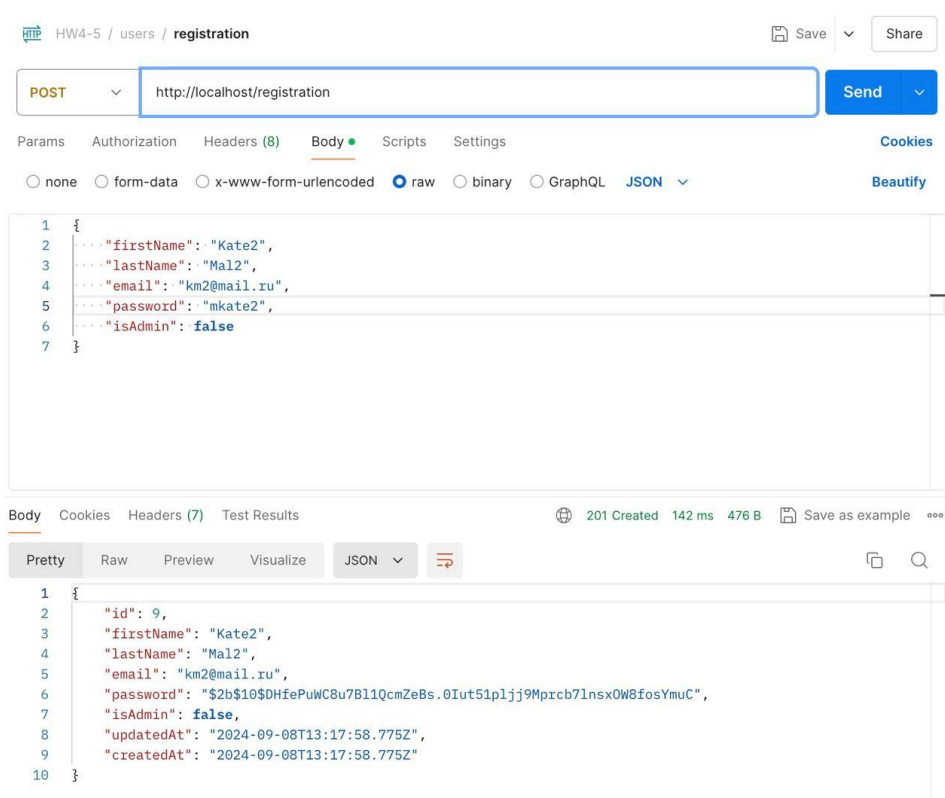
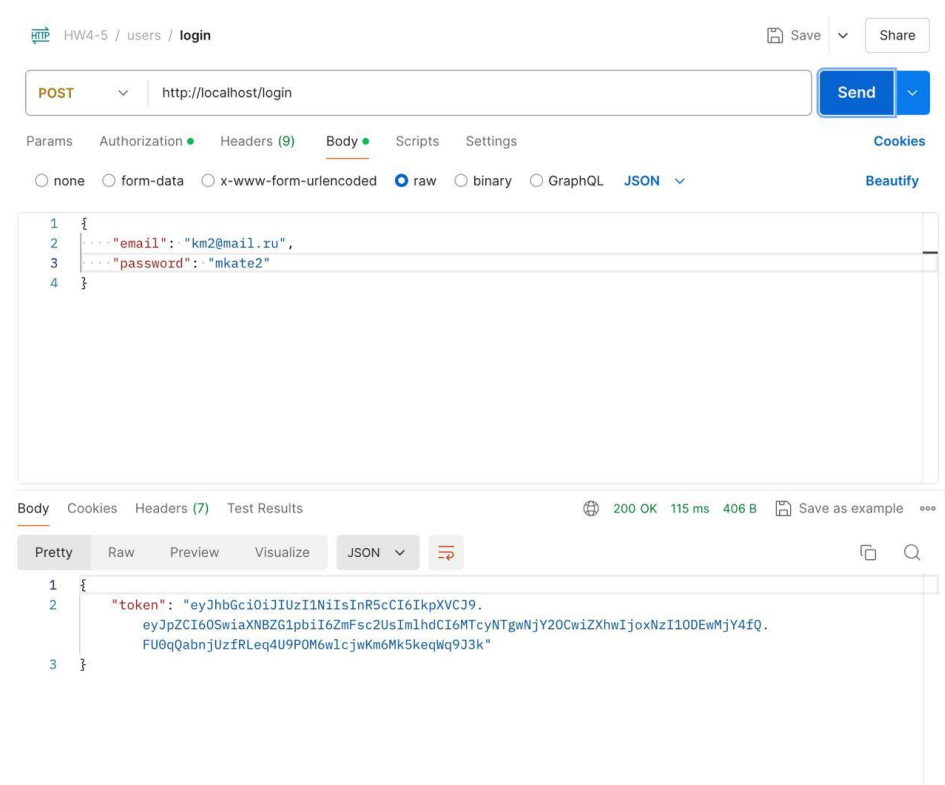
<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	 lr4		Running (4/4)		0%	9 seconds ago	<input type="checkbox"/>  
<input type="checkbox"/>	 postgres 776f734ba6d8	<a href="#">postgres</a>	Running		0%	10 seconds ago	<input type="checkbox"/>  
<input type="checkbox"/>	 lr2 06667a248c05	<a href="#">lr4-lr2</a>	Running	<a href="#">59964:3000</a>	0%	10 seconds ago	<input type="checkbox"/>  
<input type="checkbox"/>	 lr3 2089f9a2a186	<a href="#">lr4-lr3</a>	Running	<a href="#">59963:3001</a>	0%	10 seconds ago	<input type="checkbox"/>  
<input type="checkbox"/>	 nginx c7de6c37f645	<a href="#">nginx:alpine</a>	Running	<a href="#">80:80</a>	0%	9 seconds ago	<input type="checkbox"/>  

Рисунок 6 – Запущенные контейнеры



*Рисунок 7 – Проверка регистрации*



*Рисунок 8 – Проверка авторизации*

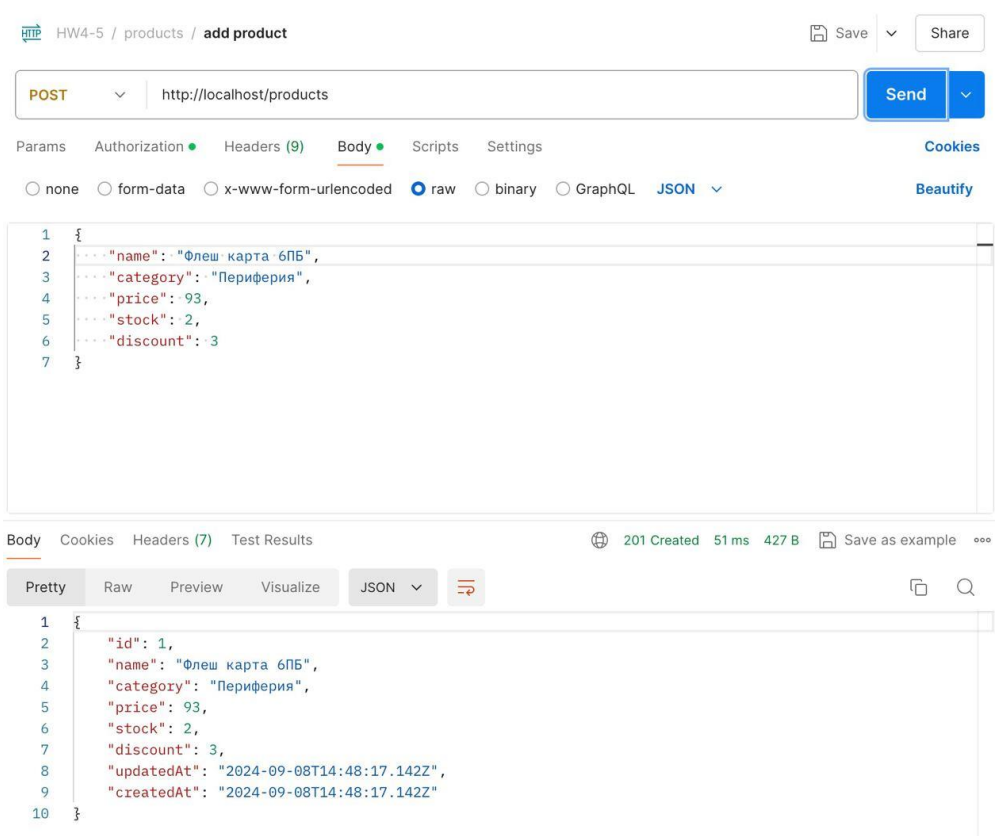


Рисунок 9 – Проверка добавления продукта

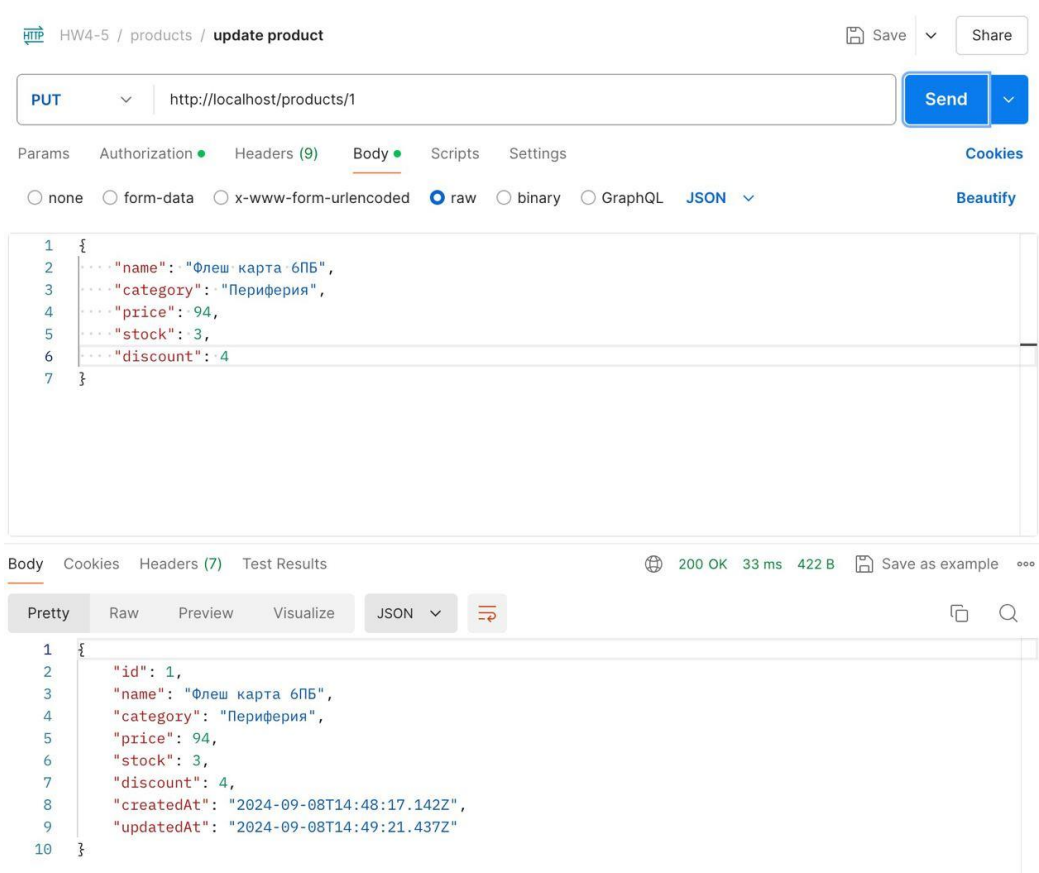
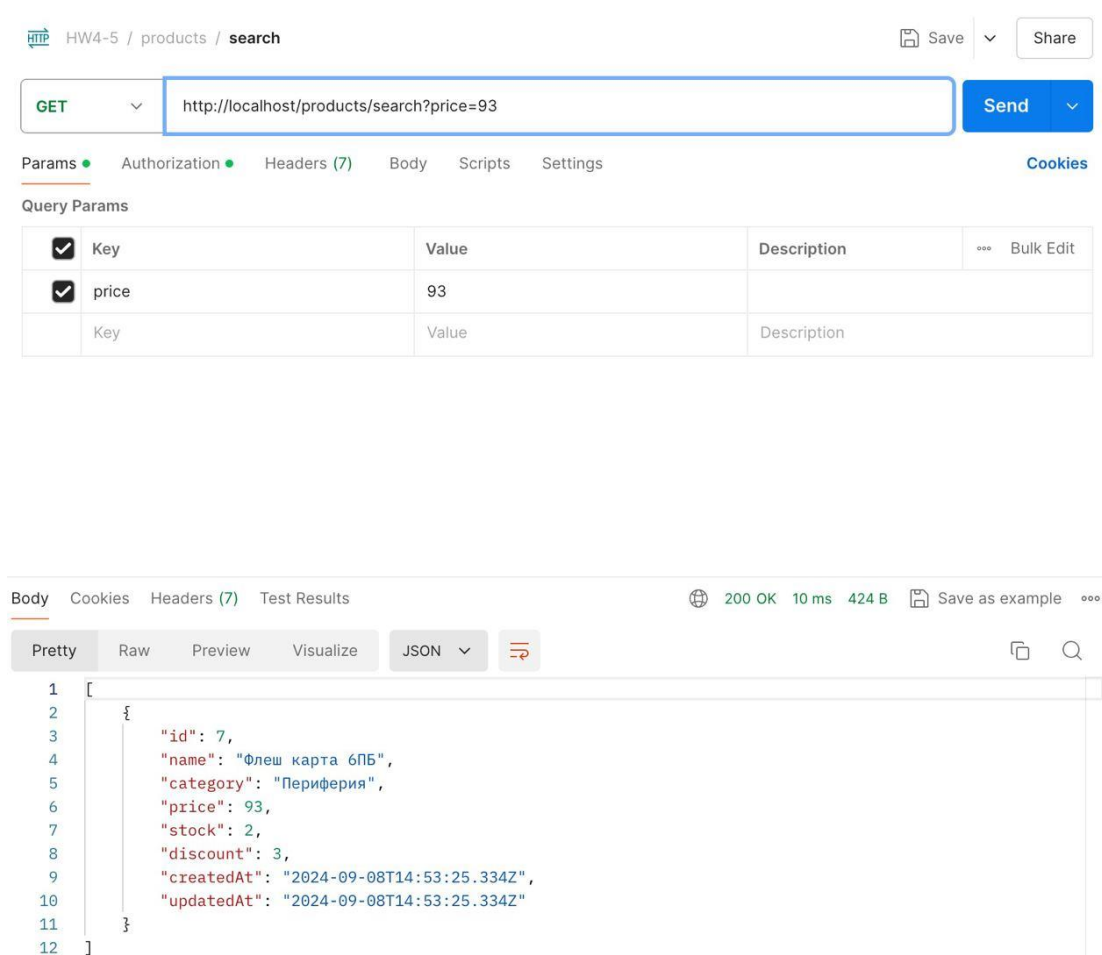


Рисунок 10 – Проверка обновления продукта



*Рисунок 11 – Проверка поиска продукта*

## Выводы.

В ходе выполнения лабораторной работы приложение было упаковано в Docker-контейнеры и настроено сетевое взаимодействие между микросервисами с помощью nginx.

Docker compose использовался для организации и устройства приложения. Данных хранятся в базе данных PostgreSQL, обрабатываются на сервере с помощью TypeScript.