

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3:
Микросервисы

Выполнила:

Лорс Хава

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

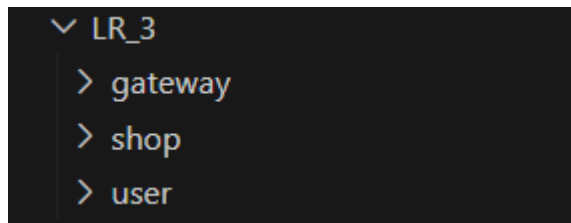
Задача

Необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций вашего приложения.

Ход работы

Реализованная ранее работа была переделана с монолитной архитектуры в микросервисную. Под микросервисы я выделила следующее: работа с пользователями *user*, с магазином *shop*. Также используется *gateway* для перенаправления по микросервисам.

Структура приложения:



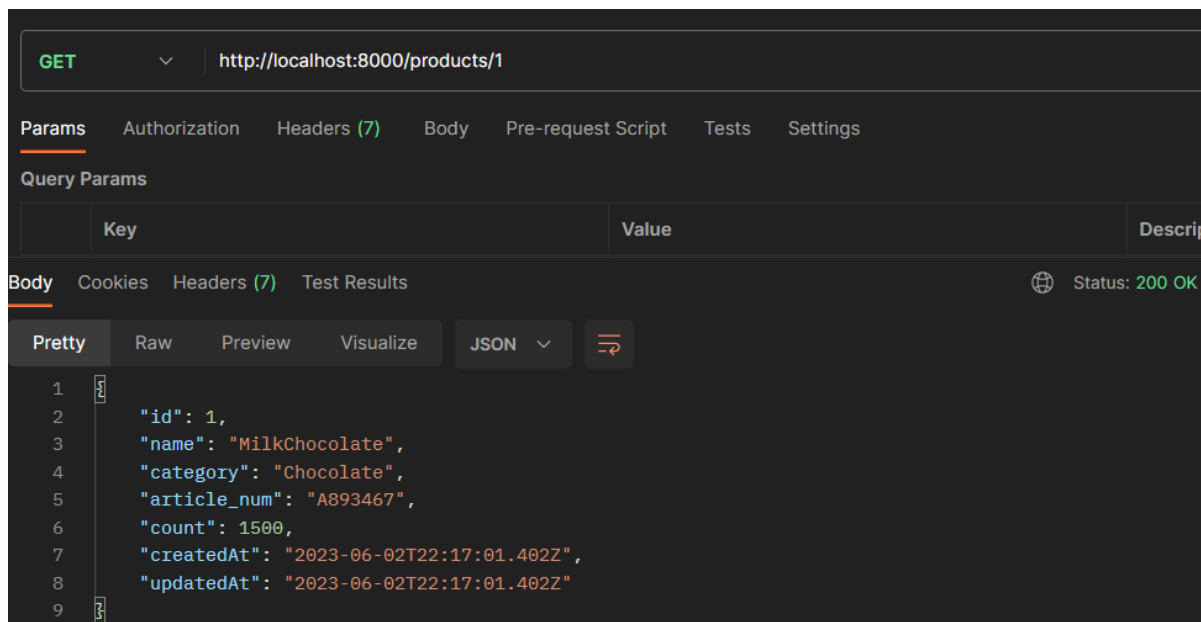
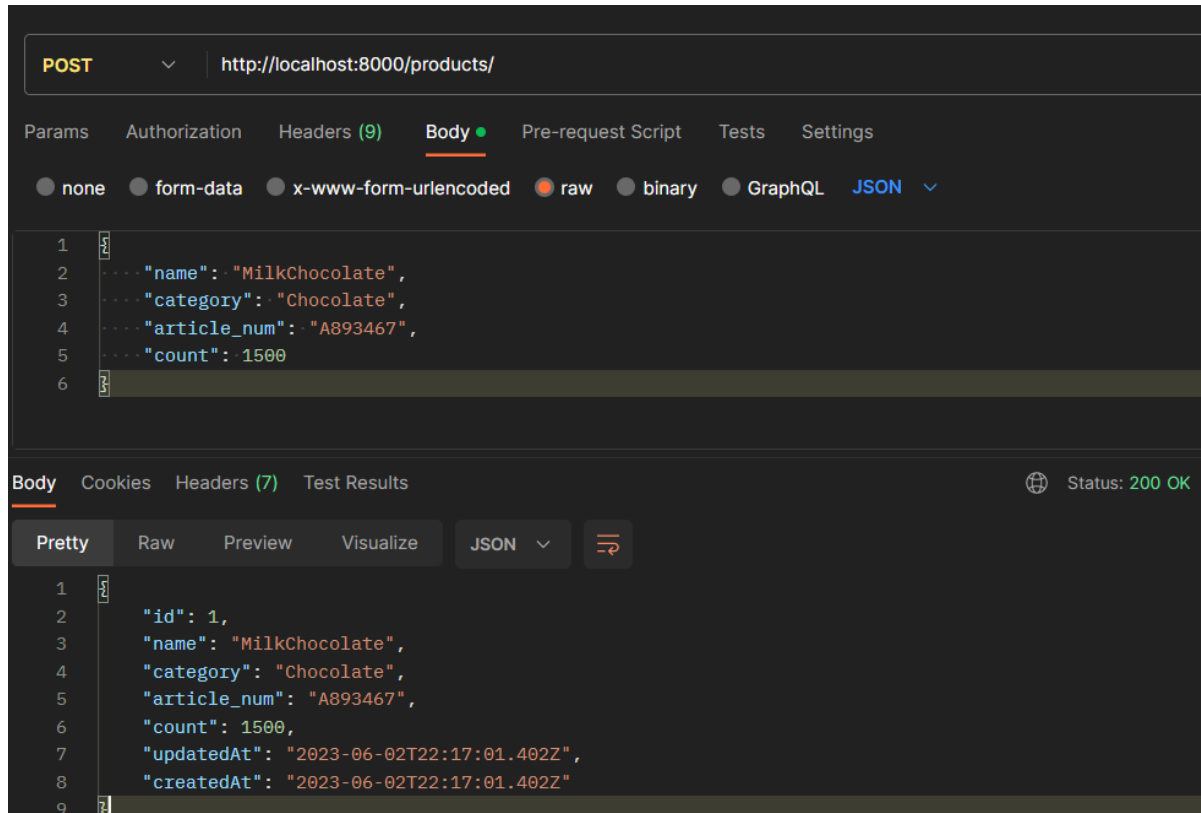
Для успешной работы и передачи данных, необходимо было правильно реализовать *gateway*, с указанием портов для переадресации.

LR_3 > gateway > JS index.js > app.all('/products/*') callback

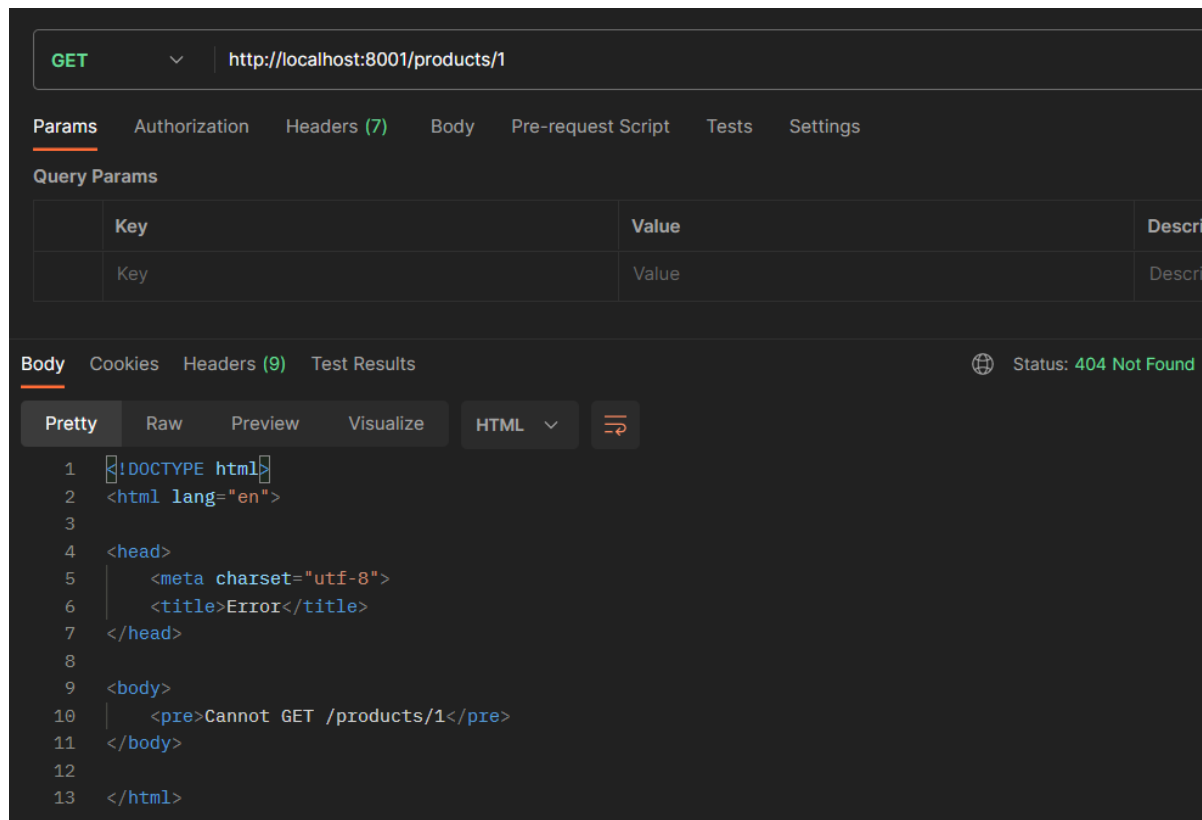
```
1  const express = require('express');
2  const axios = require('axios');
3
4  const app = express();
5  const port = 8000;
6
7  app.use(express.json());
8
9  app.all('/users/*', async (req, res) => {
10     const url = `http://localhost:8001${req.url}`;
11
12     try {
13         const response = await axios({
14             method: req.method,
15             url: url,
16             data: req.body,
17         });
18         res.status(response.status).send(response.data);
19     } catch (e) {
20         if (e.response) {
21             res.status(e.response.status).send(e.response.data);
22         } else {
23             res.status(500).send('Internal Server Error');
24         }
25     }
26 });
```

```
29 app.all('/products/*', async (req, res) => {
30     const url = `http://localhost:8002${req.originalUrl}`;
31
32     try {
33         const response = await axios({
34             method: req.method,
35             url: url,
36             data: req.body,
37         });
38         res.status(response.status).send(response.data);
39     } catch (error) {
40         if (error.response) {
41             res.status(error.response.status).send(error.response.data);
42         } else {
43             res.status(500).send('Internal Server Error');
44         }
45     }
46 });
47
48 app.listen(port, () => {
49     console.log(`Gateway listening at http://localhost:${port}`);
50 });
```

Проверим работу *gateway*, который работает на 8000 порту, для этого, например, создадим товар (*products*) и получим о нем информацию:



Проверим, что микросервисы разделены, для этого отправим, например, get-запрос на порт предназначенный для пользователей. Получаем ошибку:



Вывод

В ходе выполнения лабораторной работы были получены практические навыки работы с микросервисами.