САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 3

Выполнил:

Рыбалко Олег

Группа K33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

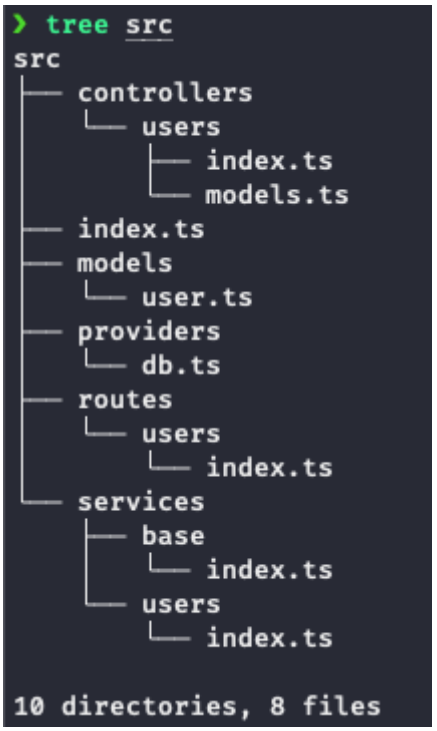
Необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций вашего приложения.

Ход работы

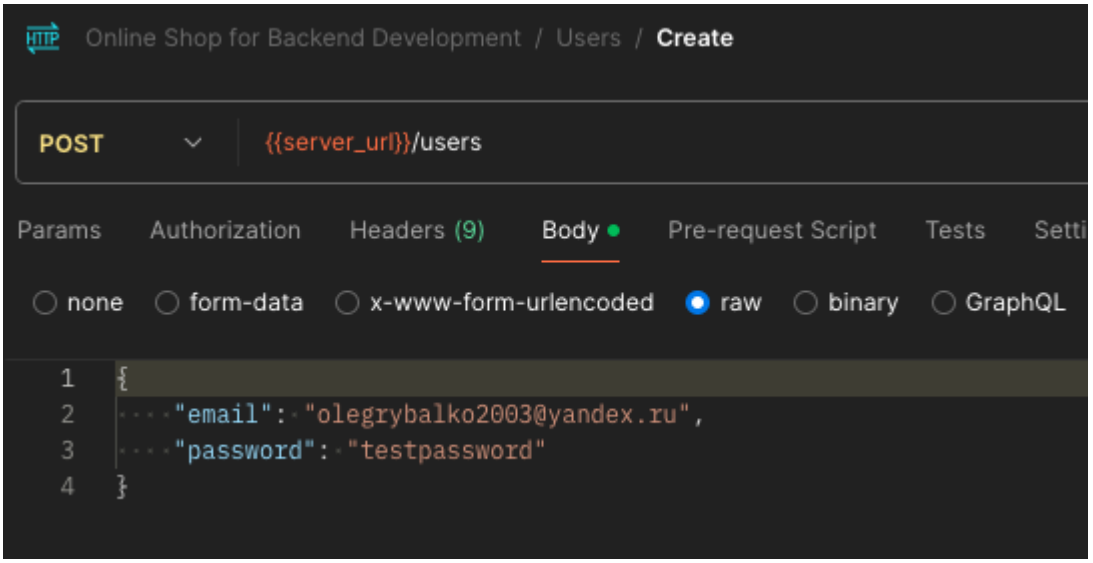
Отдельный микросервис будет работать с пользователями, а именно проводить их авторизацию и выполнять CRUD операции над ними. При помощи такого подхода мы сможем перенести хранение пользователей в отдельную базу данных, что облегчит дальнейшую работу с пользователями, так как при изменении другой логики мы точно не затронем пользователей.

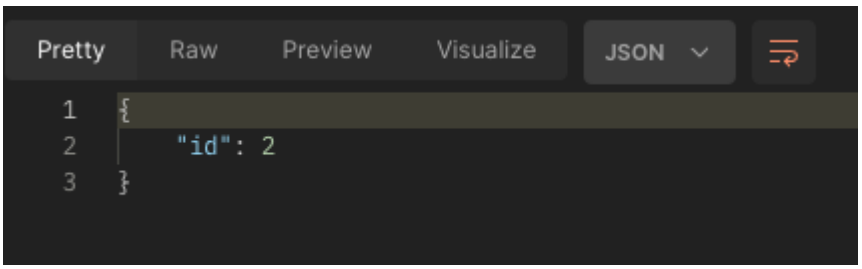
1. Создадим директорию для нашего сервиса и перенесем файлы конфигурации из предыдущей лабораторной работы

2. Перенесем все компоненты, необходимые для работы с пользователем в директорию src

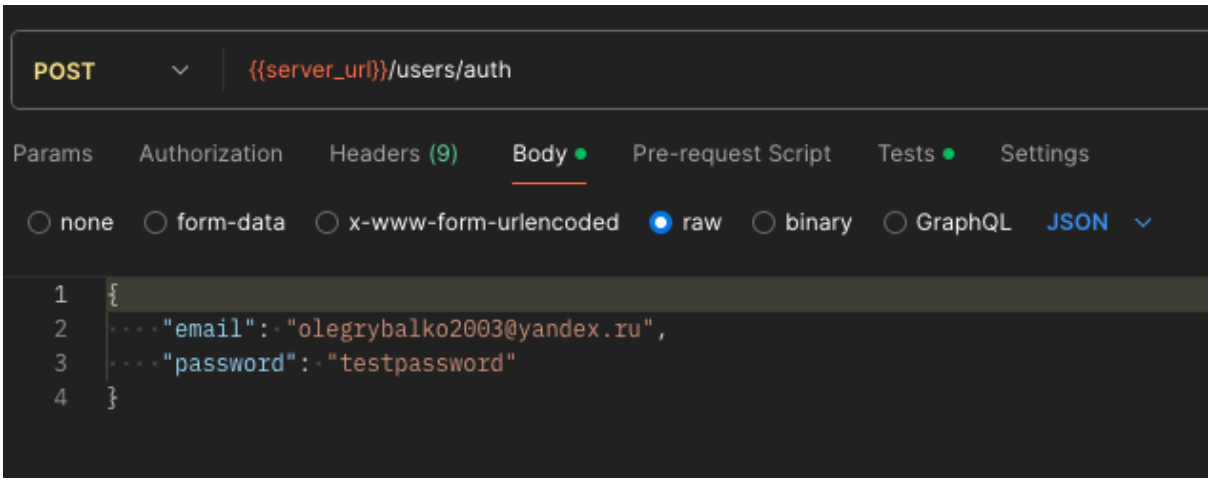


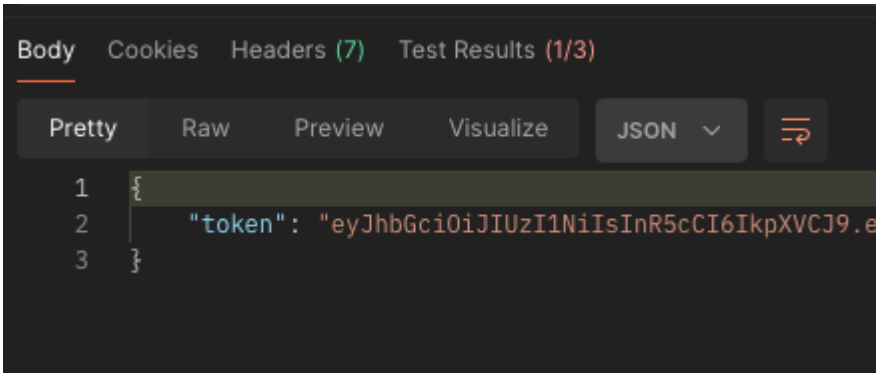
3. Запустим и проверим работу сервиса при помощи Postman
   a. Создадим пользователя



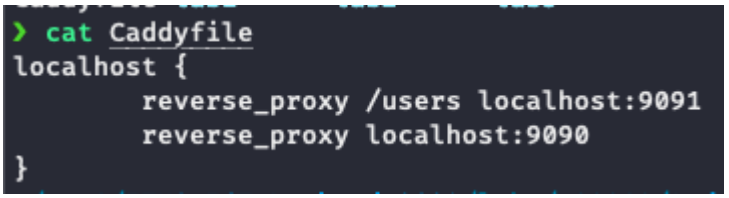


б. Авторизуемся

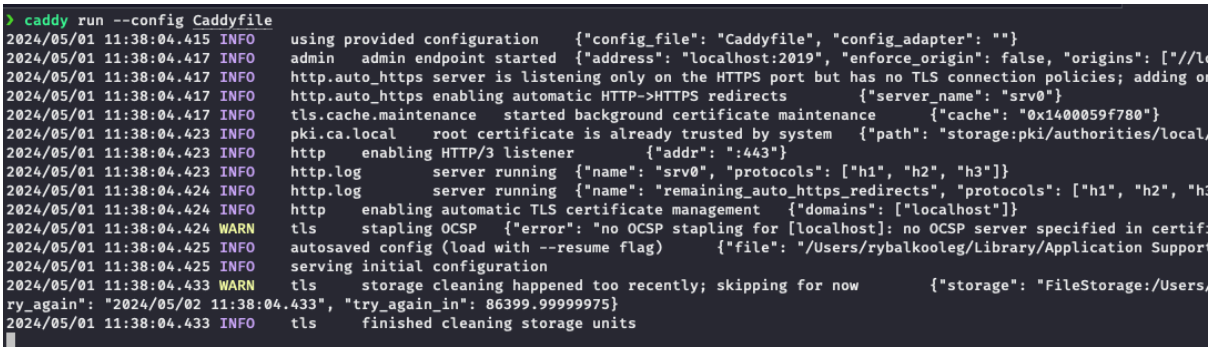


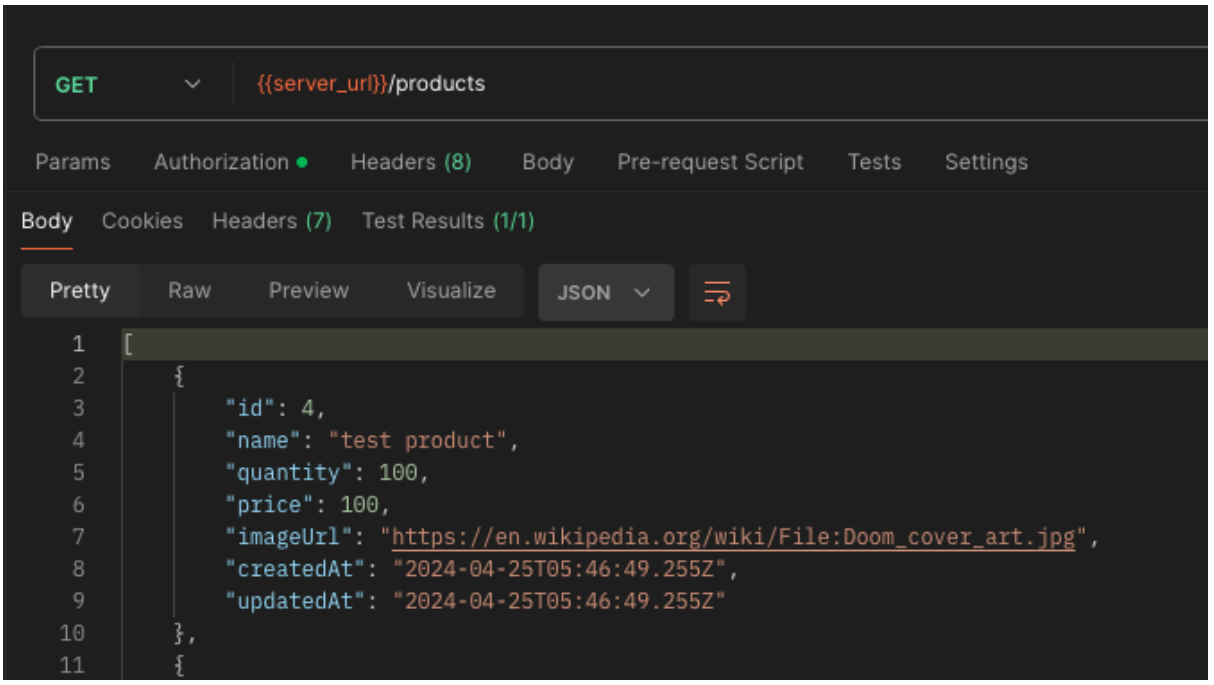


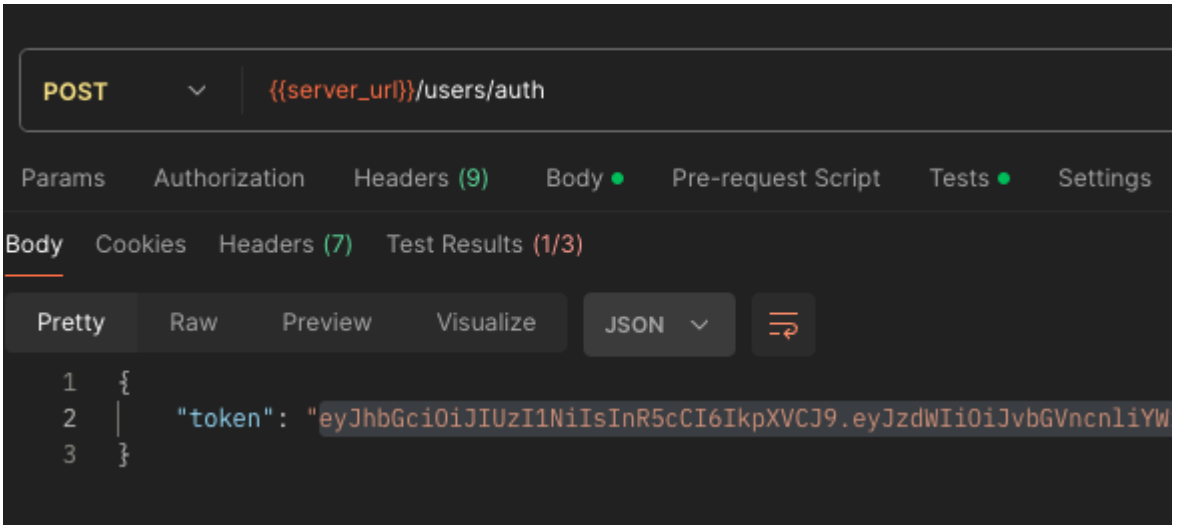
4. Наш по работе с пользователями работает, но теперь необходимо указывать два разных порта для отправки запросов, что не совсем удобно. Для решения этой проблемы на данном этапе воспользуемся caddy https://caddyserver.com/. Создадим Caddyfile, в котором укажем, что запросы с путем /users должны отправляться на порт 9091, а все остальные на порт 9090.



Запустим прокси сервер с нашей конфигурацией и убедимся, что все работает. Также caddy автоматически настраивает SSL сертификат, поэтому мы можем обращаться к серверу при помощи https протокола.






POST
{{server_url}}/users/auth
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Body
Cookies
Headers (7)
Test Results (1/3)
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJvbGVncnliYW
3 }

Вывод

В данной лабораторной работе удалось создать микросервис для работы с пользователями. Более того получилось настроить reverse proxy, для удобства обращения к серверу. В итоге клиенту необходимо обращаться к одному адресу сервера, а затем caddy распределяет запросы по необходимым сервисам.