САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3

Выполнила:

Исмагилова Карина
К3344

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Провести разделение API на микросервисы и настроить сетевое взаимодействие между микросервисами.

Ход работы

Для начала создали gateway который принимает все внешние HTTP-запросы и перенаправляет их к нужным микросервисам.
Для адресов микросервисов config.ts:

```
export const services = {
 user: 'http://localhost:3001',
 recipe: 'http://localhost:3002',
 file: 'http://localhost:3003',
 article: 'http://localhost:3004',
 feedback: 'http://localhost:3005',
 preference: 'http://localhost:3006',
};
```

С помощью proxyRoutes перенаправляем запросы по маршрутам /api/... к соответствующим микросервисам:

```
const router = express.Router();

router.use('/api/users', createProxyMiddleware({ target:
services.user, changeOrigin: true }));
router.use('/api/recipe', createProxyMiddleware({ target:
services.recipe, changeOrigin: true }));
router.use('/api/files', createProxyMiddleware({ target:
services.file, changeOrigin: true }));
router.use('/api/articles', createProxyMiddleware({
target: services.article, changeOrigin: true }));
router.use('/api/preference', createProxyMiddleware({
target: services.preference, changeOrigin: true }));
router.use('/api/feedback', createProxyMiddleware({
target: services.feedback, changeOrigin: true }));

export default router;
```

server.ts запускает Express-сервер на порту 3000 и подключает маршруты прокси.

```
import express from 'express';
import proxyRoutes from "./routes/proxyRoutes";
const app = express();
app.use(proxyRoutes);
app.listen(3000, () => {
 console.log('Gateway running on http://localhost:3000');
});
```

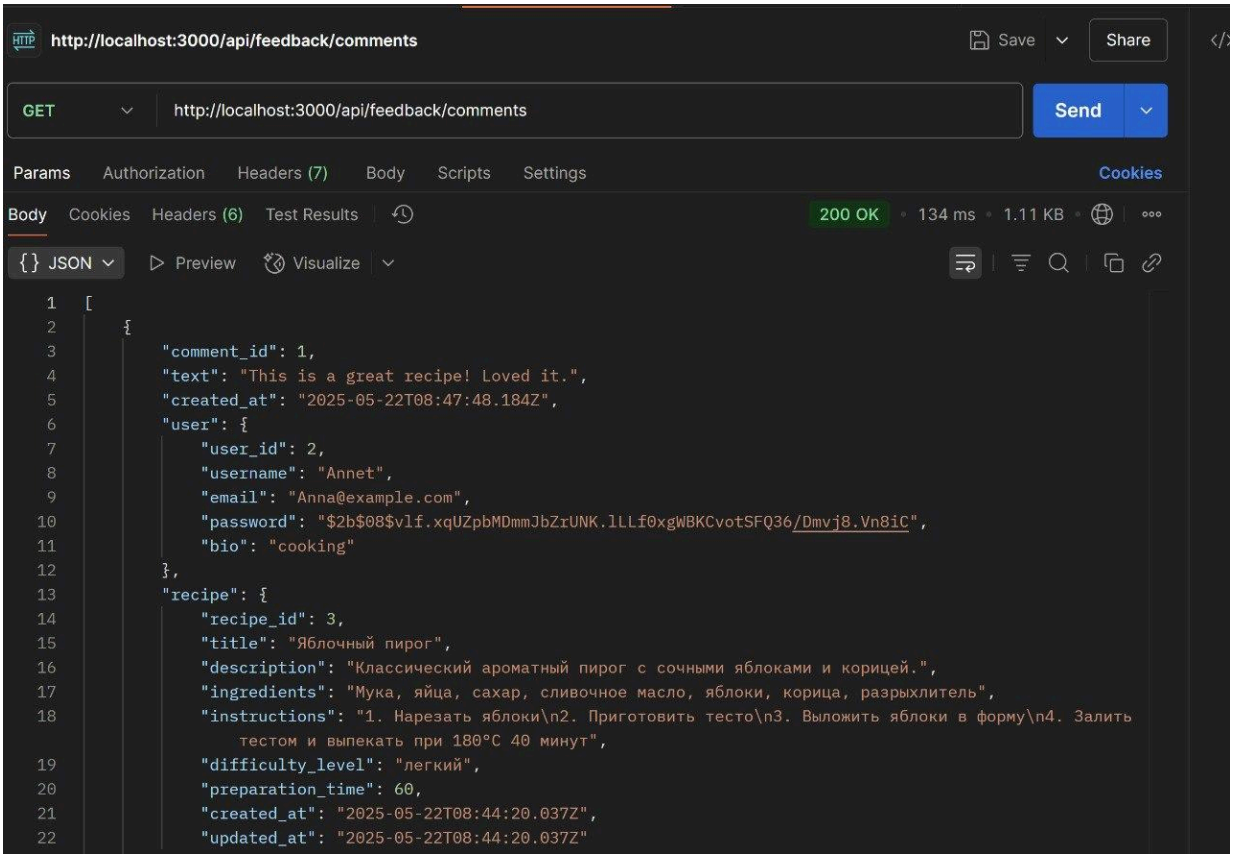
Затем разделили всё на микросервисы:

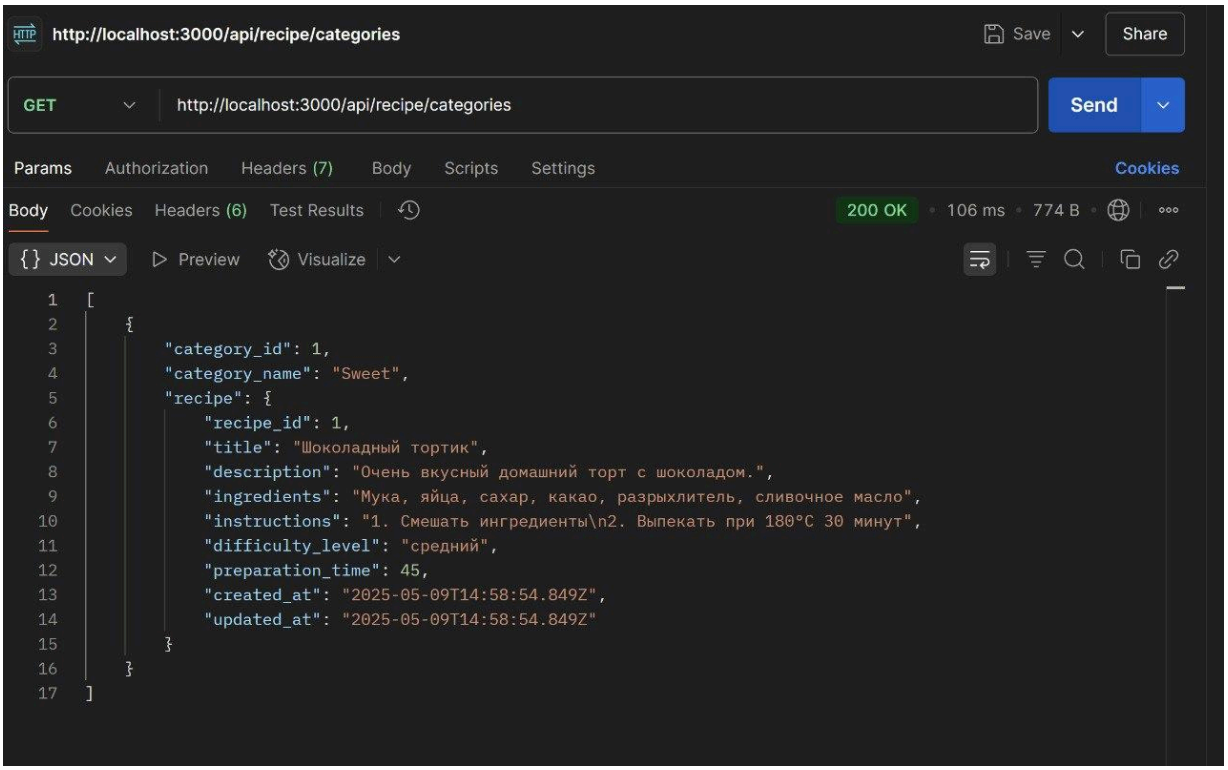
1) users-service (аутентификация/регистрация пользователей)
2) feedback-service (комментарии и оценки пользователя)
3) recipes-service (рецепты и их категории)
4) files-service (сервис для файлов к статьям и рецептам)
5) articles-service (статьи)
6) preference-service (избранное и коллекции пользователя)

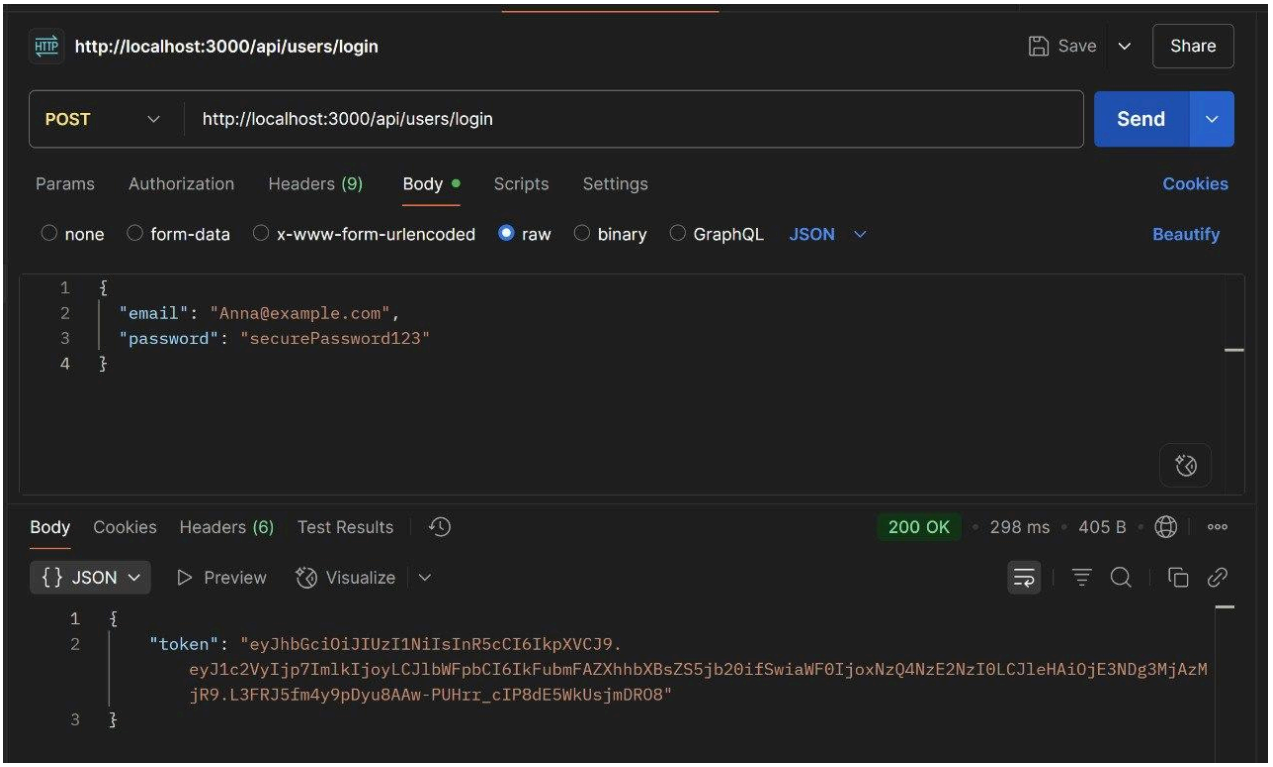
Для каждого сервиса прописали server, пример для articles-server:

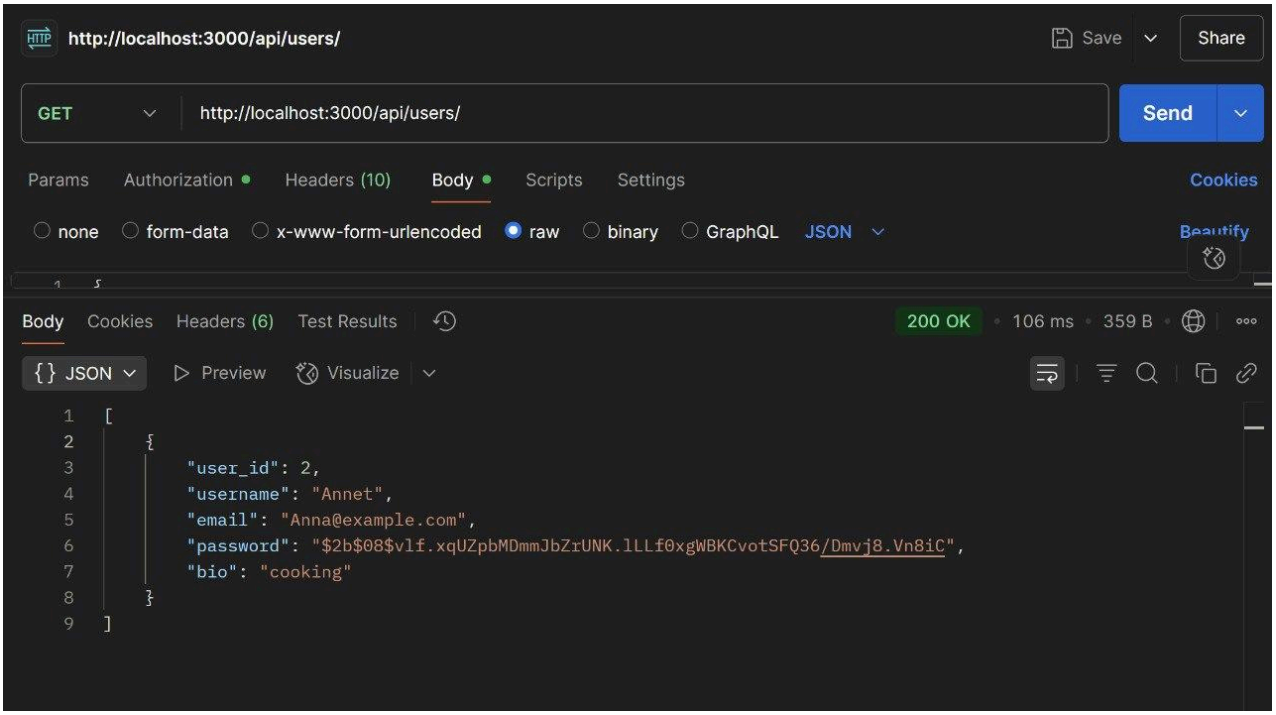
```
dotenv.config();
const app = express();
app.use(express.json());
app.use('/', articleRoutes);

const PORT = process.env.PORT || 3004;

AppDataSource.initialize().then(() => {
 console.log('Articles Service DB connected');
 app.listen(PORT, () => {
            console.log(`Articles    service    running    on
http://localhost:${PORT}`);
 });
}).catch((err) => {
 console.error('DB init error', err);
});
```

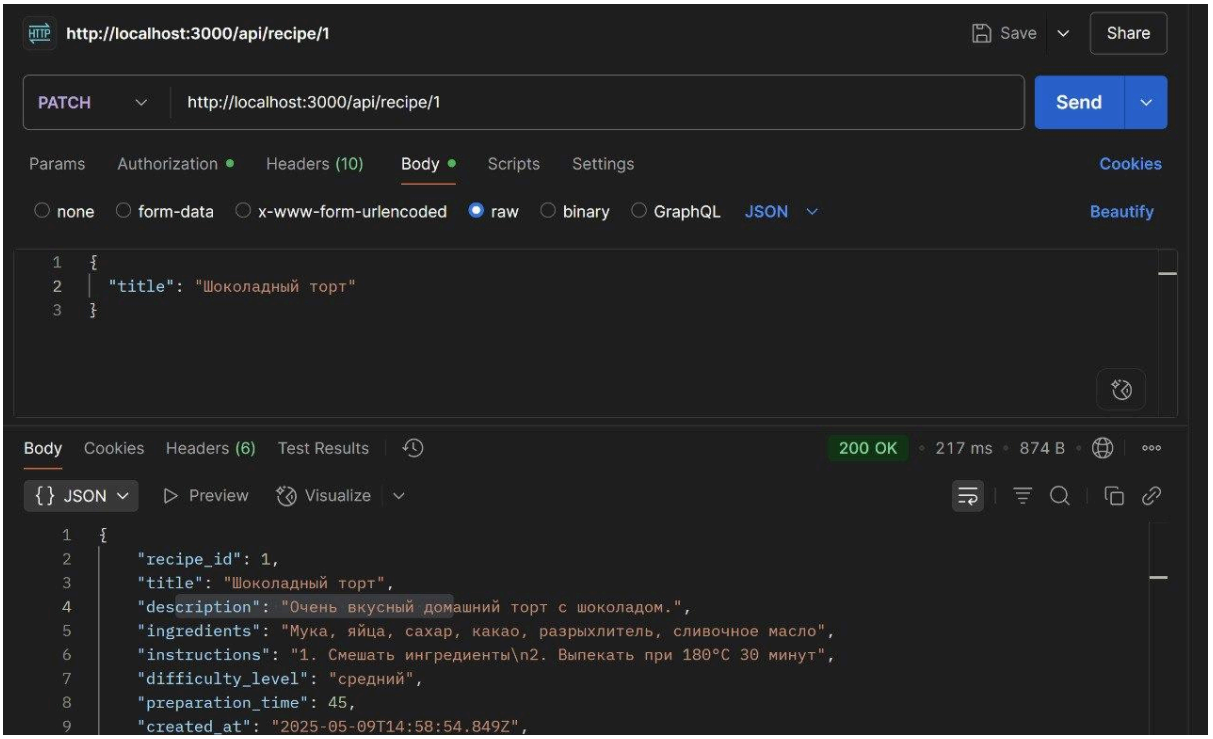
Затем проверили работоспособность:


http://localhost:3000/api/feedback/comments
Save
Share
GET
http://localhost:3000/api/feedback/comments
Send
Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies
Body
Cookies
Headers (6)
Test Results
200 OK
134 ms
1.11 KB
{} JSON
Preview
Visualize
[
{
"comment_id": 1,
"text": "This is a great recipe! Loved it.",
"created_at": "2025-05-22T08:47:48.184Z",
"user": {
"user_id": 2,
"username": "Annet",
"email": "Anna@example.com",
"password": "$2b$08$vlf.xqUZpbMDmmJbZrUNK.lLLf0xgWBKCvotSFQ36/Dmvj8.Vn8iC",
"bio": "cooking"
},
"recipe": {
"recipe_id": 3,
"title": "Яблочный пирог",
"description": "Классический ароматный пирог с сочными яблоками и корицей.",
"ingredients": "Мука, яйца, сахар, сливочное масло, яблоки, корица, разрыхлитель",
"instructions": "1. Нарезать яблоки\n2. Приготовить тесто\n3. Выложить яблоки в форму\n4. Залить тестом и выпекать при 180°C 40 минут",
"difficulty_level": "легкий",
"preparation_time": 60,
"created_at": "2025-05-22T08:44:20.037Z",
"updated_at": "2025-05-22T08:44:20.037Z"

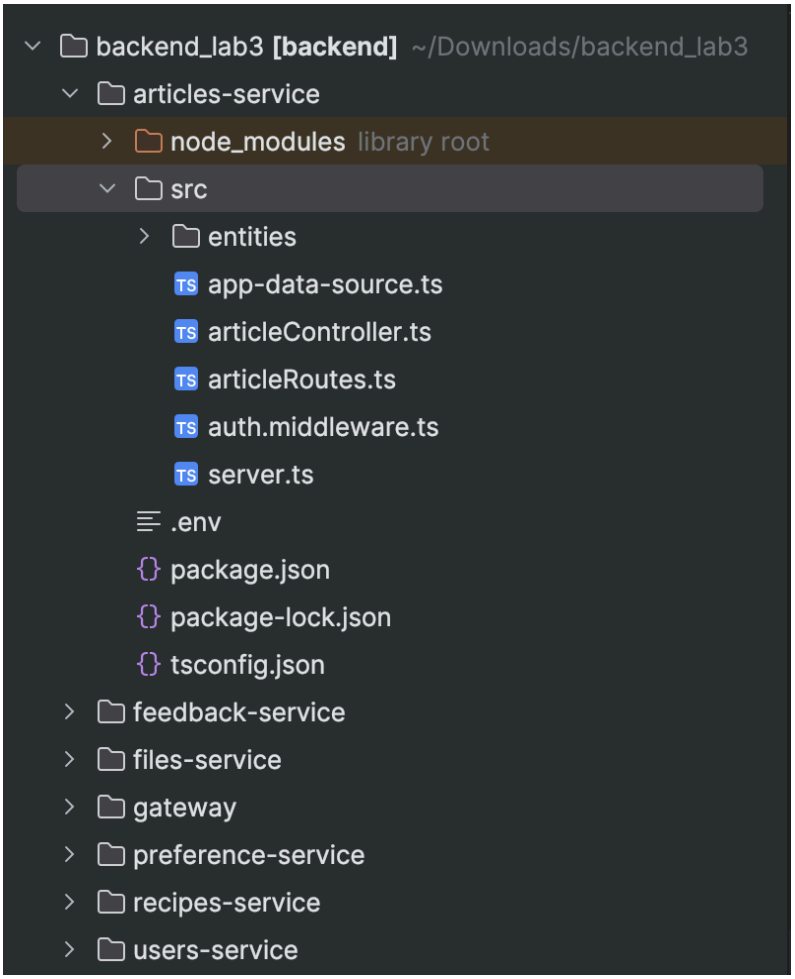

http://localhost:3000/api/recipe/categories
Save
Share
GET
http://localhost:3000/api/recipe/categories
Send
Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies
Body
Cookies
Headers (6)
Test Results
200 OK
106 ms
774 B
{} JSON
Preview
Visualize
[
{
"category_id": 1,
"category_name": "Sweet",
"recipe": {
"recipe_id": 1,
"title": "Шоколадный тортик",
"description": "Очень вкусный домашний торт с шоколадом.",
"ingredients": "Мука, яйца, сахар, какао, разрыхлитель, сливочное масло",
"instructions": "1. Смешать ингредиенты\n2. Выпекать при 180°C 30 минут",
"difficulty_level": "средний",
"preparation_time": 45,
"created_at": "2025-05-09T14:58:54.849Z",
"updated_at": "2025-05-09T14:58:54.849Z"
}
}
]


http://localhost:3000/api/users/login
Save
Share
POST
http://localhost:3000/api/users/login
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"email": "Anna@example.com",
"password": "securePassword123"
}
Body
Cookies
Headers (6)
Test Results
200 OK
298 ms
405 B
JSON
Preview
Visualize
{
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ1c2VyIjp7ImlkIjoyLCJlbWFpbCI6IkFubmFAZXhhbXBsZS5jb20ifSwiaWF0IjoxNzQ4NzE2NzI0LCJleHAiOjE3NDg3MjAzM
jR9.L3FRJ5fm4y9pDyu8AAw-PUHrr_cIP8dE5WkUsjmDRO8"
}


http://localhost:3000/api/users/
Save
Share
GET
http://localhost:3000/api/users/
Send
Params
Authorization
Headers (10)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
Body
Cookies
Headers (6)
Test Results
200 OK
106 ms
359 B
JSON
Preview
Visualize
[
{
"user_id": 2,
"username": "Annet",
"email": "Anna@example.com",
"password": "$2b$08$vlf.xqUZpbMDmmJbZrUNK.lLLf0xgWBKCvotSFQ36/Dmvj8.Vn8iC",
"bio": "cooking"
}
]



Структура проекта:



Вывод: В результате выполнения работы была успешно реализована миграция API в микросервисную архитектуру. Gateway-сервер обеспечил эффективное взаимодействие между сервисами, упростив маршрутизацию запросов.