

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3

Выполнила:

Шурубова Прасковья
К3343

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 <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

```
1 [
2   {
3     "comment_id": 1,
4     "text": "This is a great recipe! Loved it.",
5     "created_at": "2025-05-22T08:47:48.184Z",
6     "user": {
7       "user_id": 2,
8       "username": "Annet",
9       "email": "Anna@example.com",
10      "password": "$2b$08$v1f.xqUZpbMDmmJbZrUNK.1LLf0xgWBKcvotSFQ36/Dmvj8.Vn8iC",
11      "bio": "cooking"
12    },
13    "recipe": {
14      "recipe_id": 3,
15      "title": "Яблочный пирог",
16      "description": "Классический ароматный пирог с сочными яблоками и корицей.",
17      "ingredients": "Мука, яйца, сахар, сливочное масло, яблоки, корица, разрыхлитель",
18      "instructions": "1. Нарезать яблоки\n2. Приготовить тесто\n3. Выложить яблоки в форму\n4. Залить тестом и выпекать при 180°C 40 минут",
19      "difficulty_level": "легкий",
20      "preparation_time": 60,
21      "created_at": "2025-05-22T08:44:20.037Z",
22      "updated_at": "2025-05-22T08:44:20.037Z"
```

HTTP <http://localhost:3000/api/recipe/categories> Save Share

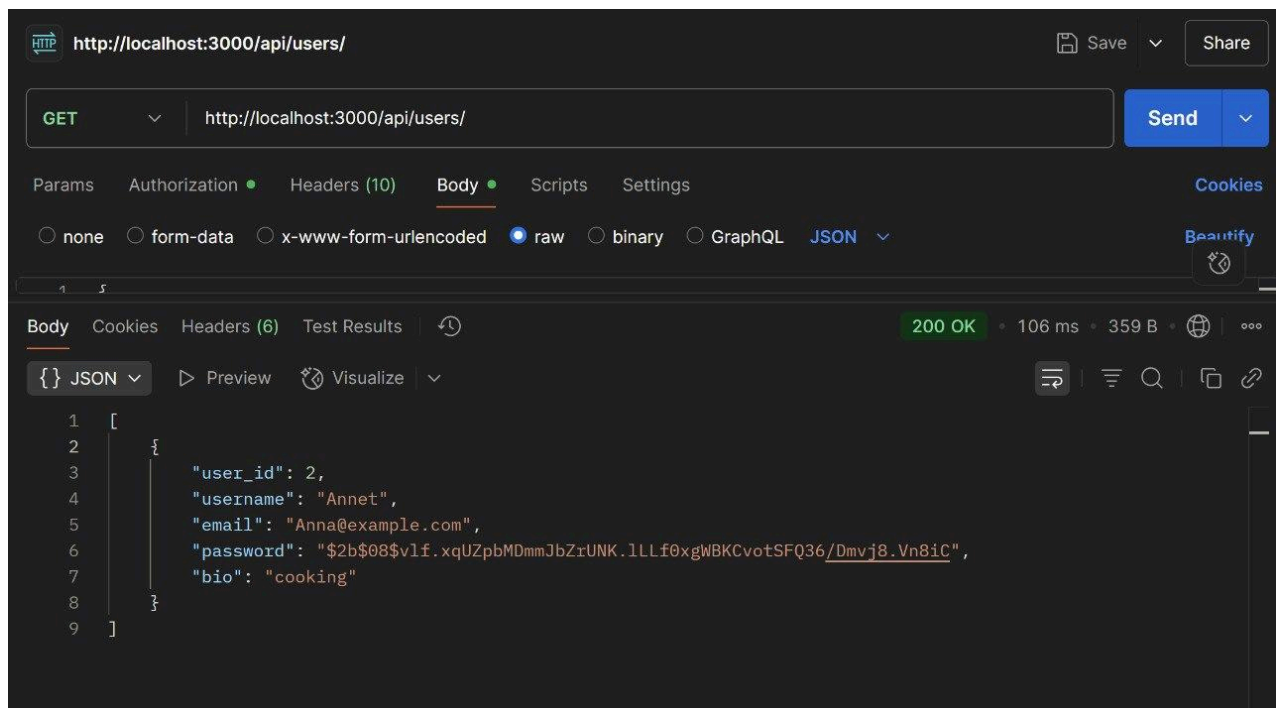
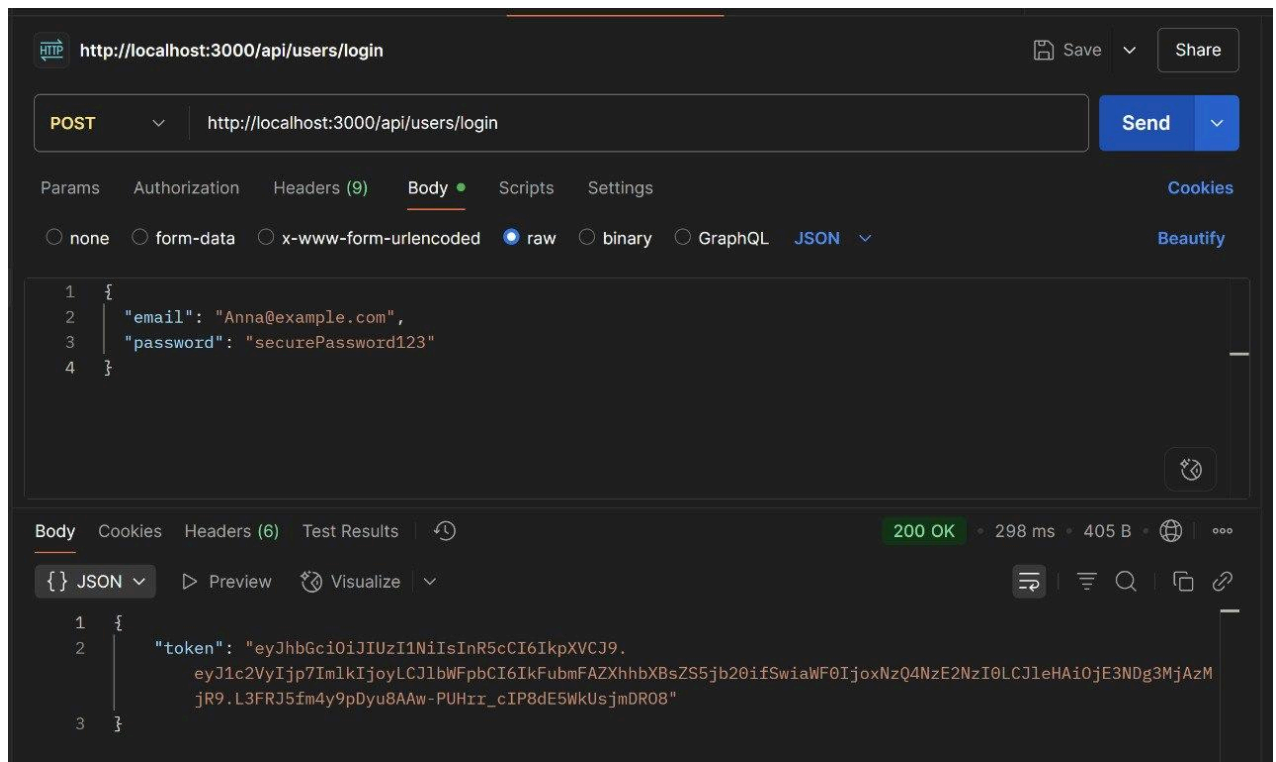
GET <http://localhost:3000/api/recipe/categories> Send

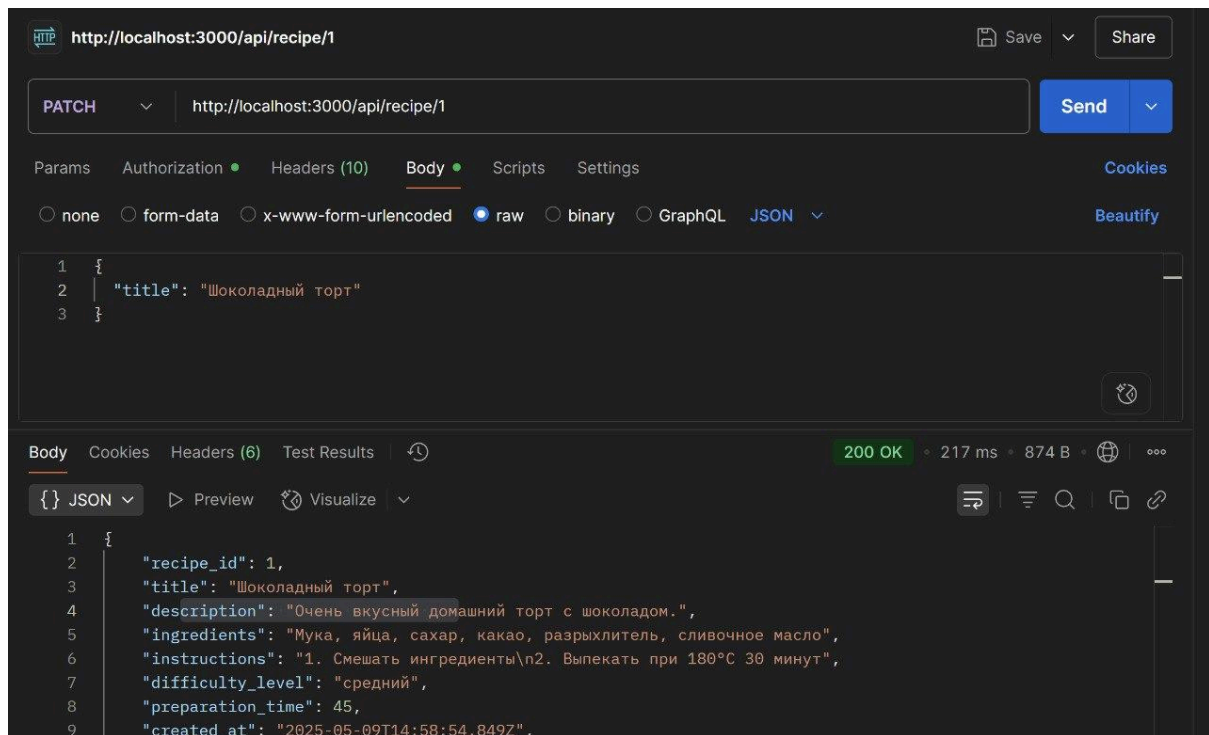
Params Authorization Headers (7) Body Scripts Settings Cookies

Body Cookies Headers (6) Test Results 200 OK • 106 ms • 774 B

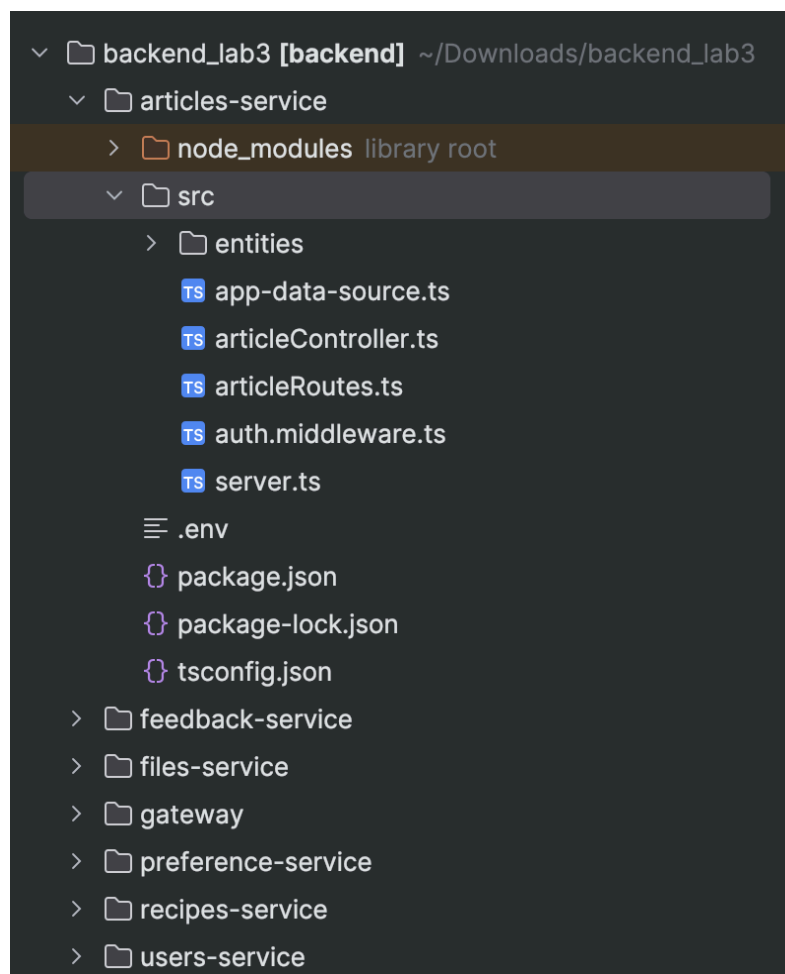
{ } JSON Preview Visualize

```
1 [
2   {
3     "category_id": 1,
4     "category_name": "Sweet",
5     "recipe": {
6       "recipe_id": 1,
7       "title": "Шоколадный торт",
8       "description": "Очень вкусный домашний торт с шоколадом.",
9       "ingredients": "Мука, яйца, сахар, какао, разрыхлитель, сливочное масло",
10      "instructions": "1. Смешать ингредиенты\n2. Выпекать при 180°C 30 минут",
11      "difficulty_level": "средний",
12      "preparation_time": 45,
13      "created_at": "2025-05-09T14:58:54.849Z",
14      "updated_at": "2025-05-09T14:58:54.849Z"
15    }
16  }
17 ]
```





Структура проекта:



Вывод: В ходе работы API было разделено на отдельные микросервисы. Для взаимодействия между ними был реализован gateway-сервер с использованием `http-proxy-middleware`, обеспечивающий централизованный доступ к сервисам.