

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 5

Выполнил:

Якунин Андрей

Группа

К3441

Проверил:

Добряков Д. И.

Санкт-Петербург

2026 г.

Задача

- выделить самостоятельные модули в вашем приложении;
- провести разделение своего API на микросервисы (минимум, их должно быть 3);
- настроить сетевое взаимодействие между микросервисами.

Ход работы

Создал 5 микросервисов

1. authService

Назначение: Аутентификация и регистрация пользователей.

Что делает: Регистрирует новых пользователей. Хранит пароли (в хешированном виде) и роли. Выдаёт и проверяет JWT-токены.

Отвечает за логин, регистрацию, верификацию токена. После регистрации синхронизирует пользователя с user-service

2. user-service

Назначение: Управление бизнес-данными пользователей.

Что делает: Хранит профили пользователей (id, username, email, роль и т.д.)

Позволяет получать, обновлять, удалять пользователей. Не хранит пароли.

Использует токен для авторизации. Получает пользователей из authService через синхронизацию

3. property-service

Назначение: Управление объектами недвижимости и избранным.

Что делает: Позволяет создавать, получать, обновлять, удалять объекты недвижимости. Позволяет добавлять объекты в избранное. Проверяет пользователя через user-service и authService. Требуется авторизация для создания/изменения объектов

4. rental-service

Назначение: Управление арендой недвижимости.

Что делает: Позволяет создавать, получать, обновлять, удалять аренды.

Проверяет, что объект недвижимости существует (через property-service).

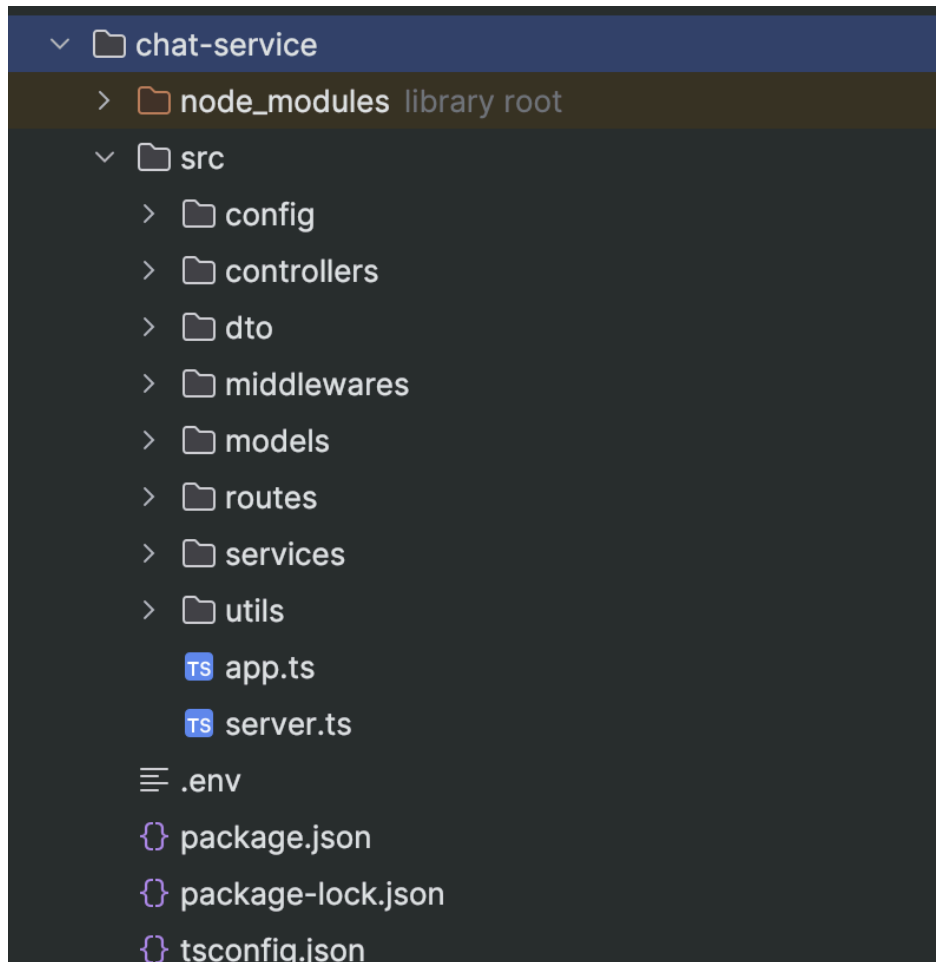
Проверяет пользователя через user-service и authService. Требуется авторизация для всех операций

5. chat-service

Назначение: Сообщения между пользователями.

Что делает: Позволяет отправлять, получать, редактировать, удалять сообщения. Привязывает сообщения к пользователям и объектам недвижимости. Проверяет пользователя через user-service и authService. Требуется авторизация для всех операций

Архитектура сервиса сообщений



/src/controllers/ — контроллеры (логика для сообщений)

/src/routes/ — роуты (маршруты для сообщений)

/src/services/ — бизнес-логика (работа с сообщениями, взаимодействие с user-service и property-service)

/src/models/ — модели (Message)

/src/middlewares/ — мидлвары (авторизация, обработка ошибок)

/src/dto/ — DTO для валидации сообщений

/src/config/ — конфиги (подключение к БД)

/src/utils/ — утилиты

/src/app.ts, server.ts — точка входа приложения

Вывод

В ходе этой работы я развернул и настроил микросервисную архитектуру для системы аренды недвижимости. Я реализовал и запустил пять сервисов:

- `authService` — отвечает за регистрацию, аутентификацию пользователей, выдачу токенов и хранение ролей.
- `user-service` — управляет бизнес-данными пользователей, их поиском, обновлением и удалением.
- `property-service` — занимается объектами недвижимости и избранным.
- `rental-service` — отвечает за аренду объектов.
- `chat-service` — реализует обмен сообщениями между пользователями.

Я настроил взаимодействие сервисов через HTTP и JWT-авторизацию, реализовал автоматическую синхронизацию пользователей между `authService` и `user-service`, устранил типовые ошибки интеграции, корректно прописал переменные окружения и очистил базы для чистого тестирования. В результате все сервисы успешно запускаются, корректно взаимодействуют между собой, а пользователь, зарегистрированный в `authService`, автоматически появляется в `user-service`. Я убедился, что все основные функции работают через API, и проект готов к дальнейшему развитию и интеграции с фронтендом.