

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа 1

Выполнил:

Якунин Андрей

Группа  
К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

## Задача

ЛР1: Реализация boilerplate

Нужно написать свой boilerplate на express + TypeORM + typescript.

Должно быть явное разделение на:

- модели
- контроллеры
- роуты

## Ход работы

### 1) Описание архитектуры

**Контроллеры (controllers):** реализуют бизнес-логику и обрабатывают HTTP-запросы.

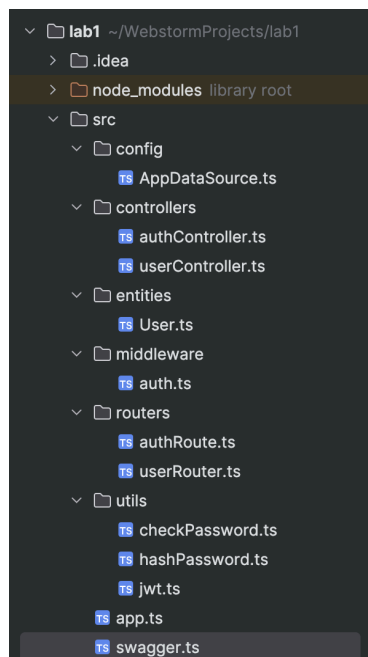
**Роутеры (routers):** маршрутизируют запросы к соответствующим контроллерам.

**Модели (entities):** описывают структуру таблиц базы данных через TypeORM.

**Утилиты (utils):** содержат вспомогательные функции — хеширование пароля, генерация JWT и т.п.

**Миддлвары (middleware):** реализуют общую логику (например, авторизация).

**Конфигурация (config):** подключение к базе данных.



## 2) Авторизация/ Регистрация и остальные команды

### 1. Регистрация

**POST** ▼ http://localhost:3000/api/auth/register

Params Authorization Headers (9) **Body** ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {
2   "username": "Andrucrut",
3   "email": "Andrucrut@mail.ru",
4   "password": "qwerty"
5 }
6
```

### 2. Авторизация

**POST** ▼ http://localhost:3000/api/auth/login

Params Authorization Headers (9) **Body** ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {
2   "email": "Andrucrut@mail.ru",
3   "password": "qwerty"
4 }
5
```

### 3. Получение всех юзеров

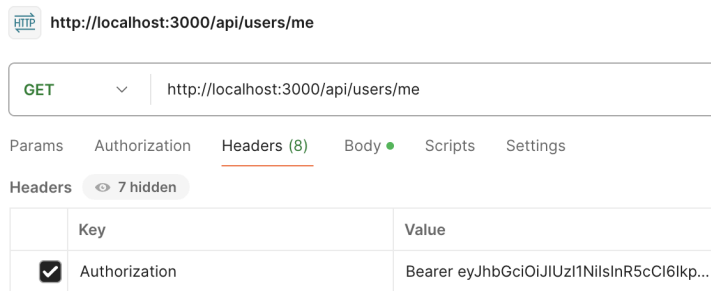
**GET** ▼ http://localhost:3000/api/users/

Params Authorization Headers (8) **Body** ● Scripts Settings

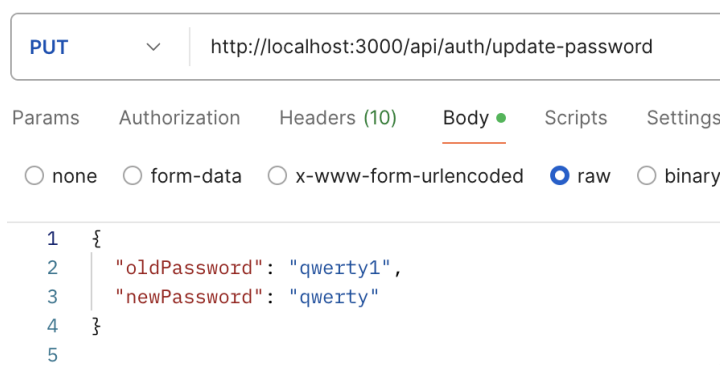
Headers 👁 7 hidden

	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp1eW...

### 4. Получение юзера



## 5. Обновление пароля



## Вывод:

Проект реализует надёжную и расширяемую архитектуру REST API с авторизацией через JWT, хешированием паролей и автогенерируемой Swagger-документацией. Благодаря модульной структуре кода (контроллеры, миддлвары, утилиты, роутеры), его легко поддерживать, масштабировать и внедрять новые функции. Использование TypeORM упрощает работу с базой данных, а Express обеспечивает быструю маршрутизацию и обработку запросов.