

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**ОТЧЕТ**

**ЛАБОРАТОРНАЯ РАБОТА №1**

Работу выполнил:  
Костылев Иван,  
студент группы К33401

Работу проверил:  
Добряков Давид Ильич

Санкт-Петербург  
2022 г.

## Задание

- Разработать собственный boilerplate на express + sequelize + Typescript.

## Ход работы

1. Инициализация проекта
2. Продумать и реализовать свою собственную модель пользователя.  
Данная модель будет содержать 5 полей:

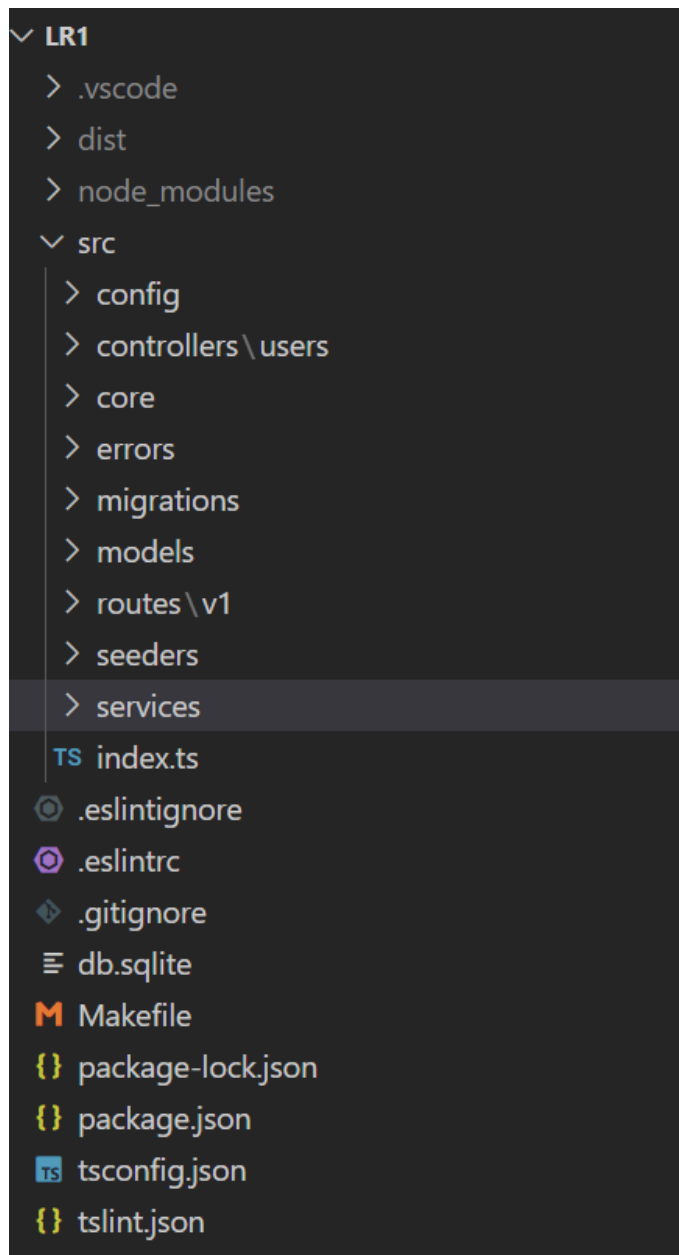
- a. Имя пользователя
- b. Пароль
- c. Электронная почта
- d. Домашний адрес
- e. Возраст

```
const User = sequelize.define<UserInstance>('Users',  
  {  
    username: {  
      type: DataTypes.STRING,  
    },  
    password: {  
      type: DataTypes.STRING,  
    },  
    email: {  
      type: DataTypes.STRING,  
    },  
    address: {  
      type: DataTypes.STRING,  
    },  
    age: {  
      type: DataTypes.INTEGER,  
    },  
  },  
);
```

3. Описание эндпоинтов и реализация контроллера и сервиса.

```
import express from "express";  
import UserController from "../controllers/users/index";  
  
const router: express.Router = express.Router()  
  
const controller = new UserController()  
  
router.route('/').get(controller.getHello)  
  
router.route('/users/new/').  
  .post(controller.createUser)  
  
router.route('/users').  
  .get(controller.getAll)  
  
router.route('/users/:id').  
  .get(controller.getById)  
  
router.route('/users/age/:age').  
  .get(controller.getByAge)  
  
export default router
```

4. Разработка проекта по паттерну “репозиторий”. Структуру проекта можно видеть на картинке ниже



## Результат

Ниже представлена работа основных эндпоинтов

## 1. GET /v1/users/ - получение списка всех пользователей

GET  [Send](#)

Params Authorization Headers (9) Body **●** Pre-request Script Tests Settings [Cookies](#)

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 36 ms 612 B [Save Response](#)

Pretty Raw Preview Visualize JSON

```
8   {
9     "age": 44,
10    "createdAt": "2022-05-26T11:38:48.115Z",
11    "updatedAt": "2022-05-26T11:38:48.115Z"
12  },
13  {
14    "id": 2,
15    "username": "Test3",
16    "password": "12345678",
17    "email": "test3@gmail.com",
```

## 2. POST /v1/users/new - создание нового пользователя

POST  [Send](#)

Params Authorization Headers (9) **Body ●** Pre-request Script Tests Settings [Cookies](#)

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#) [Beautify](#)

```
1  {
2    "username": "Test4",
3    "password": "12345678",
4    "email": "test4@gmail.com",
5    "address": "Metallistov",
6    "age": 20
7  }
```

Body Cookies Headers (7) Test Results 201 Created 26 ms 427 B [Save Response](#)

Pretty Raw Preview Visualize JSON

```
1  {
2    "id": 3,
3    "username": "Test4",
4    "password": "12345678",
```

### 3. GET /v1/users/:id - получение пользователя по id

GET ⌵ http://localhost:8000/v1/users/2 Send ⌵

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results 🌐 200 OK 11 ms 422 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ 🔍

```
1 {
2   "id": 2,
3   "username": "Test3",
4   "password": "12345678",
5   "email": "test3@gmail.com",
6   "address": "Metallistov",
7   "age": 44,
8   "createdAt": "2022-05-26T11:39:02.359Z",
9   "updatedAt": "2022-05-26T11:39:02.359Z"
10 }
```

### 4. GET /v1/users/age/:age - получение пользователей по возрасту

GET ⌵ http://localhost:8000/v1/users/age/20 Send ⌵

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results 🌐 200 OK 12 ms 424 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ 🔍

```
1 {
2   "id": 3,
3   "username": "Test4",
4   "password": "12345678",
5   "email": "test4@gmail.com",
6   "address": "Metallistov",
7   "age": 20,
8   "createdAt": "2022-05-26T12:22:07.915Z",
9   "updatedAt": "2022-05-26T12:22:07.915Z"
10 }
```

## **Вывод**

В результате выполнения данной лабораторной работы нами был разработан небольшой pet-проект с использованием Express, Sequelize и Typescript. Кажется, что работа с Typescript на маленьком проекте много усложняет работу. Этому есть ряд причин, одна из которых - нет готовых удобных решений для работы в базой данных.