

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«Национальный исследовательский университет ИТМО»**  
**(Университет ИТМО)**

**Факультет      Инфокоммуникационных технологий**

**Образовательная программа 09.03.03 Мобильные и сетевые технологии**

**О Т Ч Е Т**

по лабораторной работе № 2  
по дисциплине «Бэк-энд разработка»

**Выполнил:**

Мосин З. И.

**Проверил:**

Добряков Д. И.



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург, 2024

## Цели работы

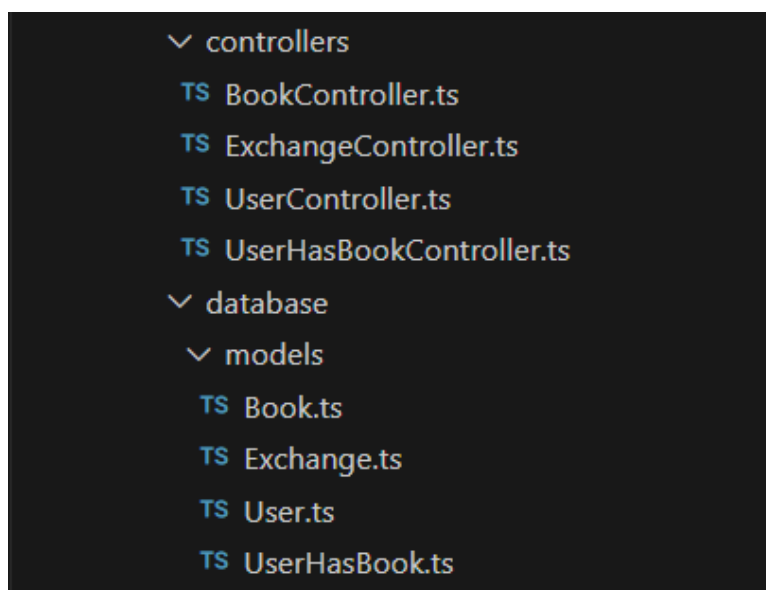
В рамках данной лабораторной работы Вам предложено выбрать один из нескольких вариантов. Выбранный вариант останется единым на весь курс и будет использоваться в последующих лабораторных работах.

По выбранному варианту необходимо будет реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate).

## Ход работы

Для данной работы мною был выбран вариант с сервисом для буккроссинга.

Далее приведены созданные мною модели и Controller-ы для работы с данными:



Пользуясь уже реализованной частью создания моделей данных в boilerplate, создал оставшиеся модели данных, всё также используя библиотеку sequelize-typescript и декораторы.

```

labs > K33402 > Мосин_Захар > lab1 > app > src > database > TS index.ts > ...
1  import dotenv from 'dotenv'
2
3  import { Sequelize } from 'sequelize-typescript'
4
5  import Book from './models/Book'
6  import Exchange from './models/Exchange'
7  import User from './models/User'
8  import UserHasBook from './models/UserHasBook'
9
10 dotenv.config()
11 console.log(process.env.DB_NAME)
12 console.log(process.env.DB_USERNAME)
13 console.log(process.env.DB_PASSWORD)
14 console.log(process.env.DB_HOST)
15 const sequelize = new Sequelize({
16   database: process.env.DB_NAME,
17   username: process.env.DB_USERNAME,
18   password: process.env.DB_PASSWORD,
19   dialect: 'postgres',
20   storage: 'db.postgres',
21   host: process.env.DB_HOST,
22   port: 5432,
23   repositoryMode: true,
24   logging: console.log,
25 })
26
27 sequelize.addModels([User, Book, UserHasBook, Exchange])
28
29 sequelize.sync().then(() => {
30   console.log('sync sequelize')
31 })

```

Все созданные способы работы с данными подключаются к нашему серверу по соответствующим эндпоинтам, подключенным через общий Router.

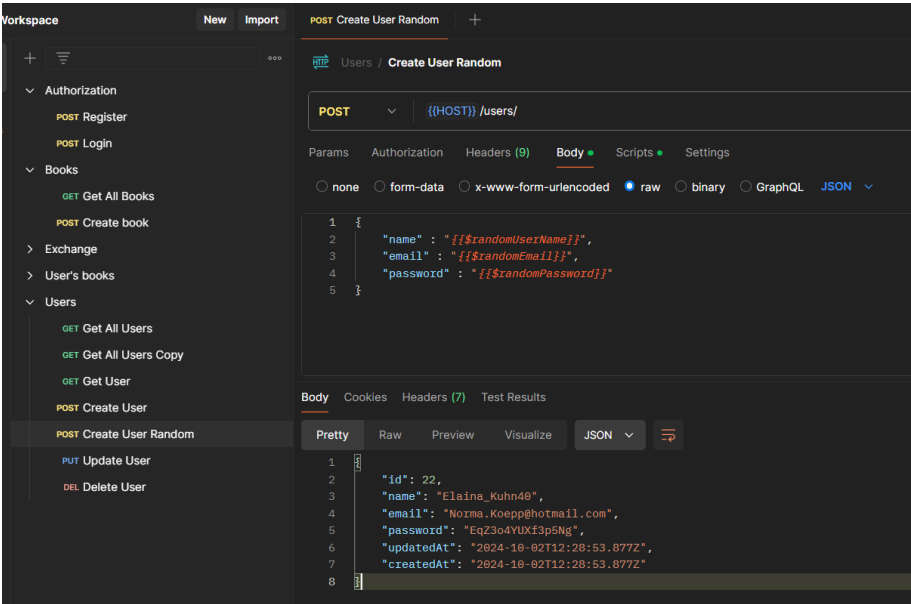
```

TS index.ts  X
labs > K33402 > Мосин_Захар > lab1 > app > src > routes > TS index.ts > ...
1  import express from 'express'
2
3  import bookRoutes from './BookRoutes'
4  import exchangeRoutes from './ExchangeRoutes'
5  import userBooksRoutes from './UserBooksRoutes'
6  import userRoutes from './UserRoutes'
7
8  const router = express.Router()
9
10 router.use('/users', userRoutes)
11 router.use('/books', bookRoutes)
12 router.use('/exchange', exchangeRoutes)
13 router.use('/userBooks', userBooksRoutes)
14
15 export default router
16

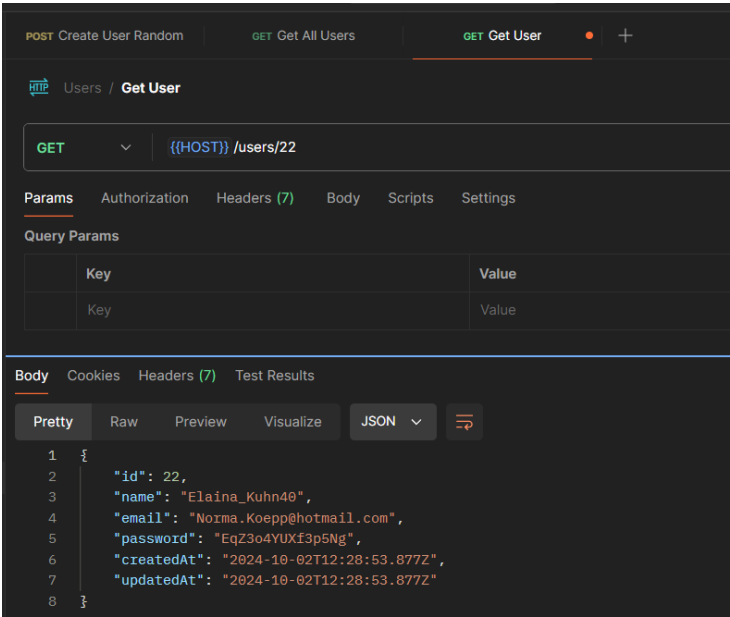
```

В итоге при отправке запросов через Postman, наш сервер взаимодействует с выбранной базой данных PostgreSQL, создавая и изменяя выбранные нами записи.

Создаём случайного пользователя:



Получаем созданного пользователя:



Также наблюдаем результат в pgadmin:

/	19	cooluser133 /	spam@mail.ru	\$2b\$10\$V1sbqVexz9riS3/y1l
8	20	Justus.Nitzsche21	Roma.Windler11@yahoo.com	tMewqhj0Q19vLiS
9	22	Elaina_Kuhn40	Norma.Koepp@hotmail.com	EqZ3o4YUXf3p5Ng
Total rows: 9 of 9    Query complete: 00:00:00.200    Rows selected: 1    In 1 Col 1				

## **ВЫВОД**

В процессе работы создали RESTful API, используя express и typescript. Подключили несколько моделей данных, поработали с данными через отправку тестовых запросов.