

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 2

Выполнил:

Пономарев Константин Витальевич

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Продумать свою собственную модель пользователя

Реализовать набор из CRUD-методов для работы с пользователями средствами Express + Sequelize

Написать запрос для получения пользователя по id/email

Ход работы

```
npm ERR! A complete log of this run can be found in: /Users/ko.ponomarev/.npm/_logs/2024-03-17/17_15_43_9342-00009-0.log
ko.ponomarev@macbook-J02YKWH46F hw_2 % npx sequelize-cli model:generate --name User --attributes "firstName:string,lastName:string,email:string,password:string"

Sequelize CLI [Node: 21.7.1, CLI: 6.6.2, ORM: 6.37.1]

New model was created at /Users/ko.ponomarev/Developer/ITMO-ICT-Backend-2024/homeworks/K33402/ponomarev_konstantin/hw_2/models/user.js .
New migration was created at /Users/ko.ponomarev/Developer/ITMO-ICT-Backend-2024/homeworks/K33402/ponomarev_konstantin/hw_2/migrations/20240317171559-create-user.js .
ko.ponomarev@macbook-J02YKWH46F hw_2 % npx sequelize db:migrate
```

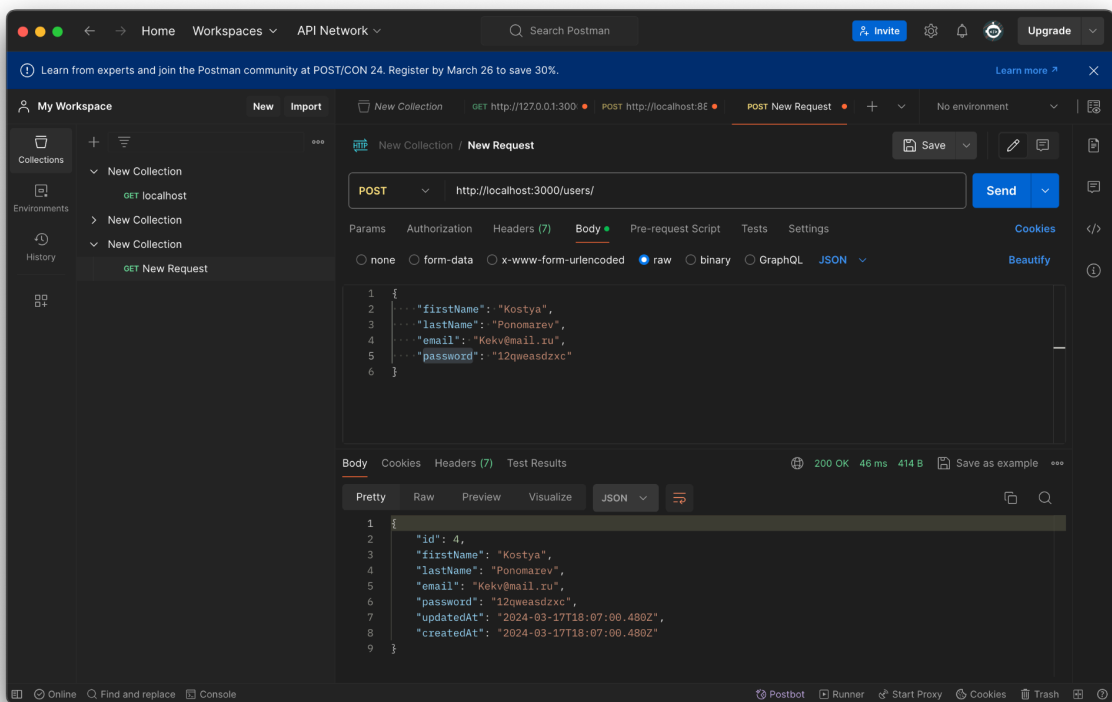
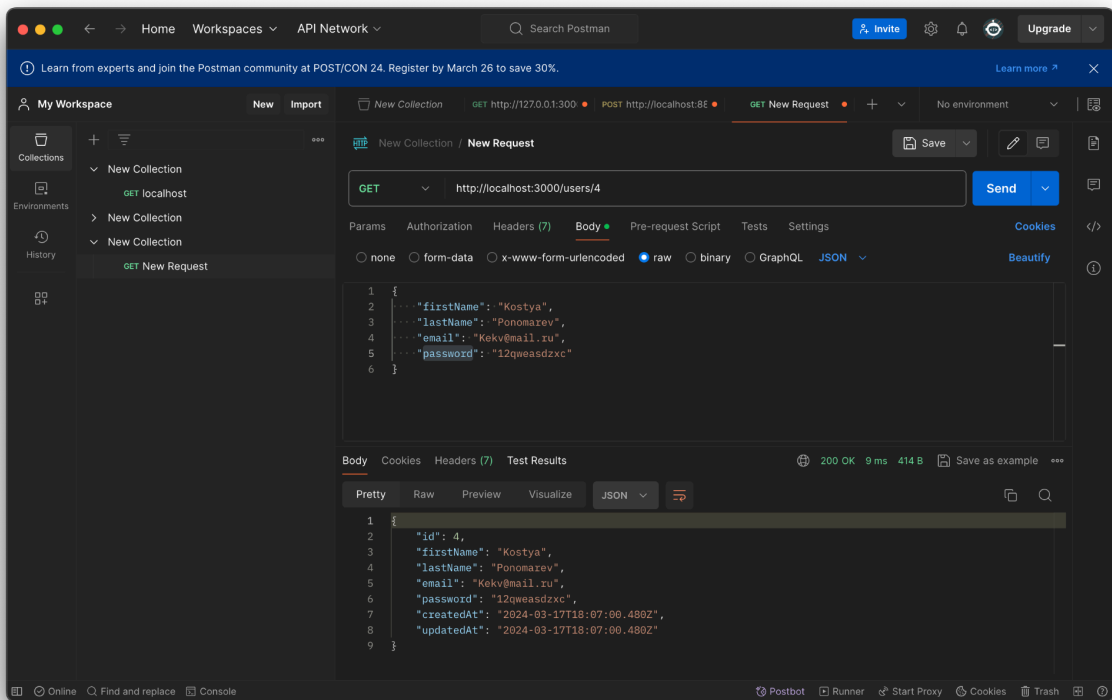
```
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    2 usages
7    class User extends Model {
8      /**
9       * Helper method for defining associations.
10      * This method is not a part of Sequelize lifecycle.
11      * The 'models/index' file will call this method automatically.
12      */
13      2 usages
14      static associate(models) {
15        // define association here
16      }
17    }
18    User.init({ attributes: {
19      firstName: DataTypes.STRING,
20      lastName: DataTypes.STRING,
21      email: DataTypes.STRING,
22      password: DataTypes.STRING
23    }, options: {
24      sequelize,
25      modelName: 'User',
26    });
27    return User;
28  };
29
```

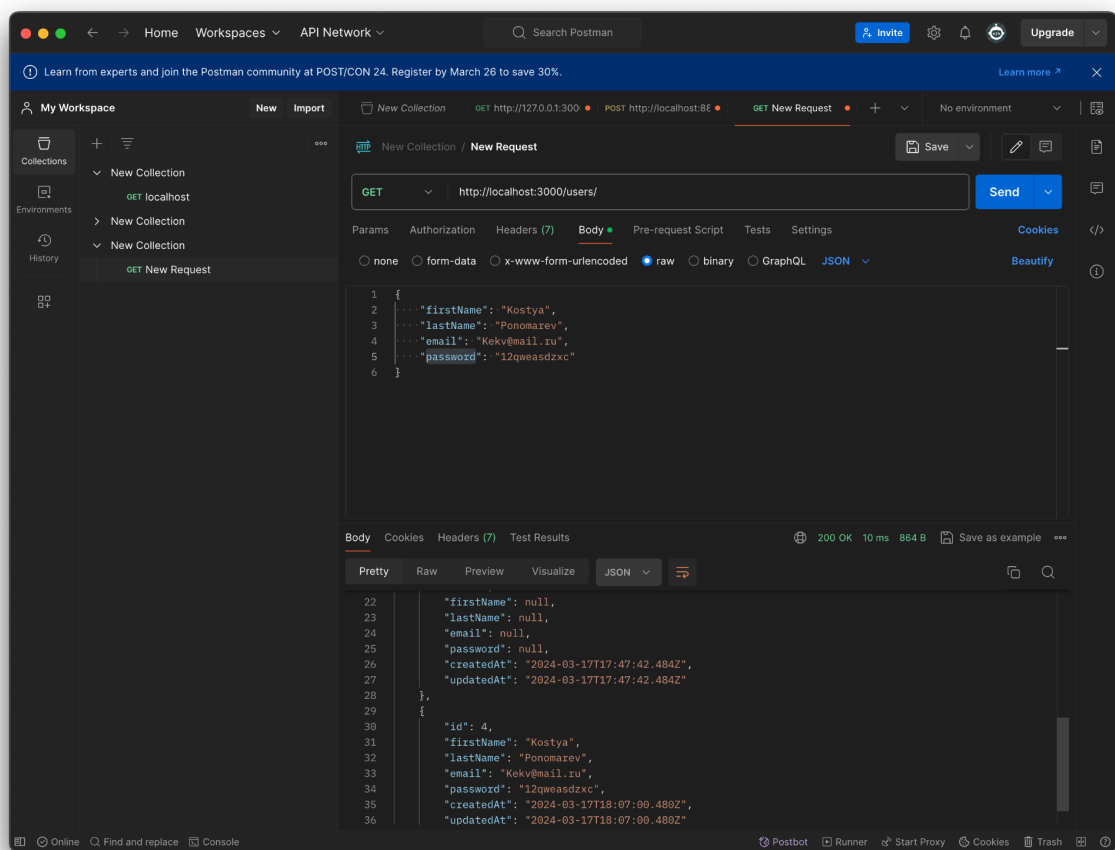
```
Loaded configuration file "config/config.json".
Using environment "development".
== 20240317171559-create-user: migrating =====
== 20240317171559-create-user: migrated (0.007s)
```

```
ke.ponomarev@macbook- ID2YKWH4GE hw 2 %
```

```
8 app.get('/users', async (req : Request<...> , res : Response<ResBody, LocalsObj> ) : Promise<any> => {
9   const user : Model[] = await db.User.findAll()
10  return res.send(user)
11
12 })
13
14 new *
15 app.get('/users/:id', async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<any> => {
16   const user : Promise<...> | Promise<...> = await db.User.findById(req.params.id)
17
18   if (user) {
19     return res.send(user.toJSON())
20   }
21   return res.status(404).send({ body: {errorMessage: "user is not found"}})
22 })
23
24 new *
25 app.post( {path: "/users", handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<any> => {
26   const user = await db.User.create(req.body)
27   return res.send(user.toJSON())
28 })
29
30 new *
31 app.put( {path: "/users/:id", handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<void> => {
32   const user : Promise<...> | Promise<...> = await db.User.findById(req.params.id)
33   if (user) {
34     await user.update(req.body)
35     res.send(user.toJSON())
36   } else {
37     res.status(404).send({ body: {errorMessage: "user not found"}})
38   }
39 })
40
41 new *
42 app.delete( {path: "/users/:id", handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<void> => {
43   const user : Promise<...> | Promise<...> = await db.User.findById(req.params.id)
44   if (user) {
45     await user.destroy()
46     res.send()
47   } else {
48
49   }
50 })
51
52 db
```

Однако json неправильно парсился и приходили в sqlite null, чтобы это исправить, я завязался на библиотеку body-parser. И теперь можно спокойно пользоваться сервисом. Все нормально парсится, данные не теряются.





Вывод

Научился работать с sequelize для создания моделей и express для создания простого сервиса CRUD