

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа 2

Знакомство с ORM Sequelize

Выполнил:

Хайрнасов Андрей Константинович

K33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## Задача

Задание:

- 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

Модель user состоит из firstName, lastName, email, password, username.

```
File: user.js
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    class User extends Model {static associate(models) {}}
7    User.init({
8      firstName: DataTypes.STRING,
9      lastName: DataTypes.STRING,
10     email: DataTypes.STRING,
11     password: DataTypes.STRING,
12     username: DataTypes.STRING
13   }, {
14     sequelize,
15     modelName: 'User',
16   });
17   return User;
18 };
19
```

API

```

10 app.get('/users/:id', async (req, res) => {
11     const user = await db.User.findById(req.params.id)
12
13     console.log('User is', user)
14     if (user) {
15         return res.send(user.toJSON())
16     }
17
18     res.status(404)
19     return res.send({"FATAL": "User not found exception"})
20 })
21
22 app.get('/users', async (req, res) => {
23     const users = await db.User.findAll()
24     return res.send(users)
25 })
26
27 app.post('/users', async (req, res) => {
28     try {
29         const user = await db.User.create(req.body)
30         return res.send(user.toJSON())
31     } catch (e) {
32         res.status(400)
33         return res.send({"FATAL": "Failed to create user"})
34     }
35 })
36
37 app.put('/users/:id', async (req, res) => {
38     var user = await db.User.findById(req.params.id)
39
40     if (user) {
41         try {
42             user.update(req.body, {where: {id: req.params.id}})
43             user = await db.User.findById(req.params.id)
44             return res.send(user.toJSON())
45         } catch (e) {
46             res.status(400)
47             return res.send({"FATAL": "Failed to update user"})
48         }
49     }
50
51     res.status(404)
52     return res.send({"FATAL": "User not found exception"})
53 })
54

```

```

55 app.delete('/users/:id', async (req, res) => {
56     const user = await db.User.findByPk(req.params.id)
57
58     if (user) {
59         try {
60             user.destroy({where: {id: req.params.id}})
61             return res.send({"msg": "User successfully disintegrated"})
62         } catch (e) {
63             res.status(400)
64             return res.send({"FATAL": "Failed to disintegrate user"})
65         }
66     }
67
68     res.status(404)
69     return res.send({"FATAL": "User not found exception"})
70 })
71
72 app.get('/', async (req, res) => {
73     res.send('https://youtu.be/dQw4w9WgXcQ')
74 })
75

```

## Запуск приложения

```

[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Demo app is listening on port 8080. Ara-ara~

```

## Пример использования API

```

dazai at DazaiOsamu in /home/dazai
$ curl 127.0.0.1:8080/users/ -X POST --data '{"email": "asd", "firstName": "asd", "lastName": "dsfsdf", "password": "asdw", "username": "asfasf"}' -H 'content-type: application/json'
{"id":15,"email":"asd","firstName":"asd","lastName":"dsfsdf","password":"asdw","username":"asfasf","updatedAt":"2023-06-04T11:23:15.288Z","createdAt":"2023-06-04T11:23:15.288Z"}
dazai at DazaiOsamu in /home/dazai
$ curl 127.0.0.1:8080/users/0 (0.032s)
{"id":15,"firstName":"asd","lastName":"dsfsdf","email":"asd","password":"asdw","username":"asfasf","createdAt":"2023-06-04T11:23:15.288Z","updatedAt":"2023-06-04T11:23:15.288Z"}
dazai at DazaiOsamu in /home/dazai
$ curl 127.0.0.1:8080/users/15 0 (0.015s)
{"id":15,"firstName":"asd","lastName":"dsfsdf","email":"asd","password":"asdw","username":"asfasf","createdAt":"2023-06-04T11:23:15.288Z","updatedAt":"2023-06-04T11:23:15.288Z"}
dazai at DazaiOsamu in /home/dazai
$ curl 127.0.0.1:8080/users/15 -X PUT --data '{"email": "a@b.c"}' -H 'content-type: application/json' 0 (0.016s)
{"id":15,"firstName":"asd","lastName":"dsfsdf","email":"a@b.c","password":"asdw","username":"asfasf","createdAt":"2023-06-04T11:23:15.288Z","updatedAt":"2023-06-04T11:23:39.921Z"}
dazai at DazaiOsamu in /home/dazai
$ curl 127.0.0.1:8080/users/15 -X DELETE 0 (0.038s)
{"msg":"User successfully disintegrated"}
dazai at DazaiOsamu in /home/dazai
$
0 (0.015s)

```

## Вывод

В ходе выполнения данной домашней работы были успешно реализованы ключевые задачи, связанные с разработкой модели пользователя и

созданием CRUD-методов для работы с пользователями с использованием Express и Sequelize.