

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа № 2
Знакомство с ORM Sequelize

Выполнила:

Штрейх Анна

К33401

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

Задачи

- 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

Создан проект в VSCode, установлены необходимые пакеты, прописаны команды старта и преста́рта.

```
"name": "hw_2",
"version": "1.0.0",
"description": "",
"main": "index.js",
  > Debug
"scripts": {
  "prestart": "echo \"App starting..\"",
  "start": "node index.js"
},
"author": "",
"license": "ISC",
"dependencies": {
  "body-parser": "^1.19.2",
  "express": "^4.17.3",
  "sequelize": "^6.17.0",
  "sqlite3": "^4.2.0"
},
"devDependencies": {
  "nodemon": "^2.0.15",
  "sequelize-cli": "^6.4.1"
```

Рисунок 1 – Файл package.json

Создана модель пользователя. Полями являются имя, фамилия, почта, пароль, возраст и страна.

```
User.init({
  firstName: DataTypes.STRING,
  lastName: DataTypes.STRING,
  email: DataTypes.STRING,
  password: DataTypes.STRING,
  age: DataTypes.INTEGER,
  country: DataTypes.STRING
}, {
  sequelize,
  modelName: 'User',
});
return User;
```

Рисунок 2 – Модель пользоателя

Реализованы CRUD методы.

```
const express = require('express')
const bodyParser = require("body-parser");
const db = require('./models')

const app = express()
const port = 3002

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.post('/users', async (req, res) => {
  const user = await db.User.create(req.body)
  res.send(user.toJSON())
})
```

Рисунок 3 – Файл index.js(1)

```

app.get('/users', async (req, res) => {
  const users = await db.User.findAll()
  res.send(users)
})

app.get('/users/id/:id', async (req, res) => {
  const user = await db.User.findByPk(req.params.id)

  console.log('user is', user)

  if (user) {
    respBody = user.toJSON()
  }
  else {
    respBody = { "msg": "user not found" }
  }
  res.send(respBody)
})

```

Рисунок 4 – Файл index.js(2)

```

app.get('/users/email/:email', async (req, res) => {
  const user = await db.User.findOne({ where: { email: req.params.email } })
  if (user) {
    respBody = user.toJSON()
  }
  else {
    respBody = { "msg": "user not found" }
  }
  res.send(respBody)
})

app.put('/users/:id', async (req, res) => {
  const num = await db.User.update(req.body, { where: { id: req.params.id } })
  if (num == 1) {
    respBody = { "status": "success" }
  }
  else {
    respBody = { "msg": "user not found" }
  }
  res.send(respBody)
})

app.delete('/users/:id', async (req, res) => {
  const user = await db.User.destroy({ where: { id: req.params.id } })
  if (user) {
    respBody = { "status": "success" }
  }
  else {
    {
      respBody = { "msg": "user not found" }
    }
  }
  res.send(respBody)
})

```

Рисунок 5 – Файл index.js(3)

Выводы:

Изучены основы работы с Express и Sequelize, написаны CRUD-методы для работы с пользователем, в том числе получение его по определенной почте.