

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа №2

Выполнил:

Фоменко Иван

Группа

К33401

Проверил:

Добряков Д. И.

Санкт-Петербург

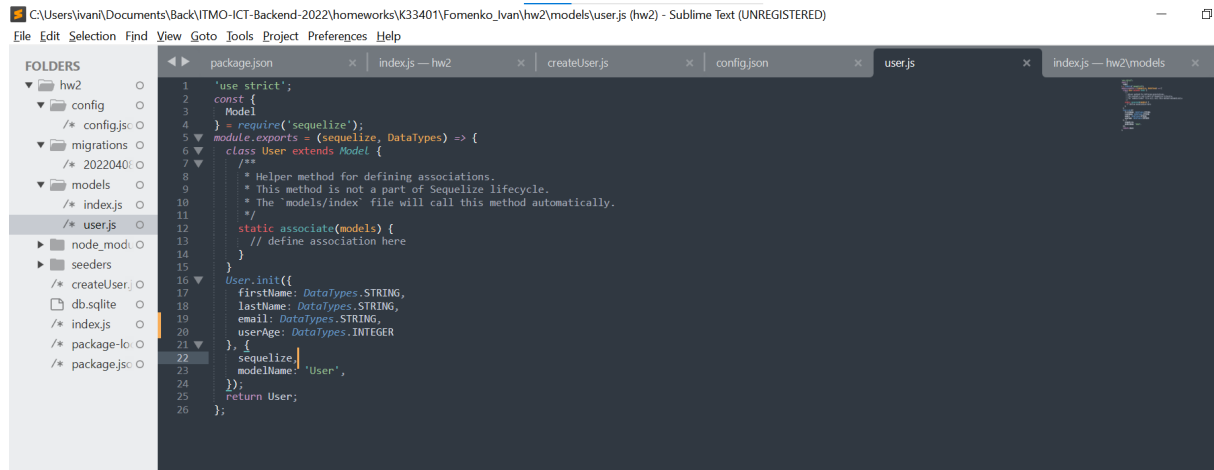
2022 г.

Задача

- Продумать свою собственную модель пользователя

Ход работы

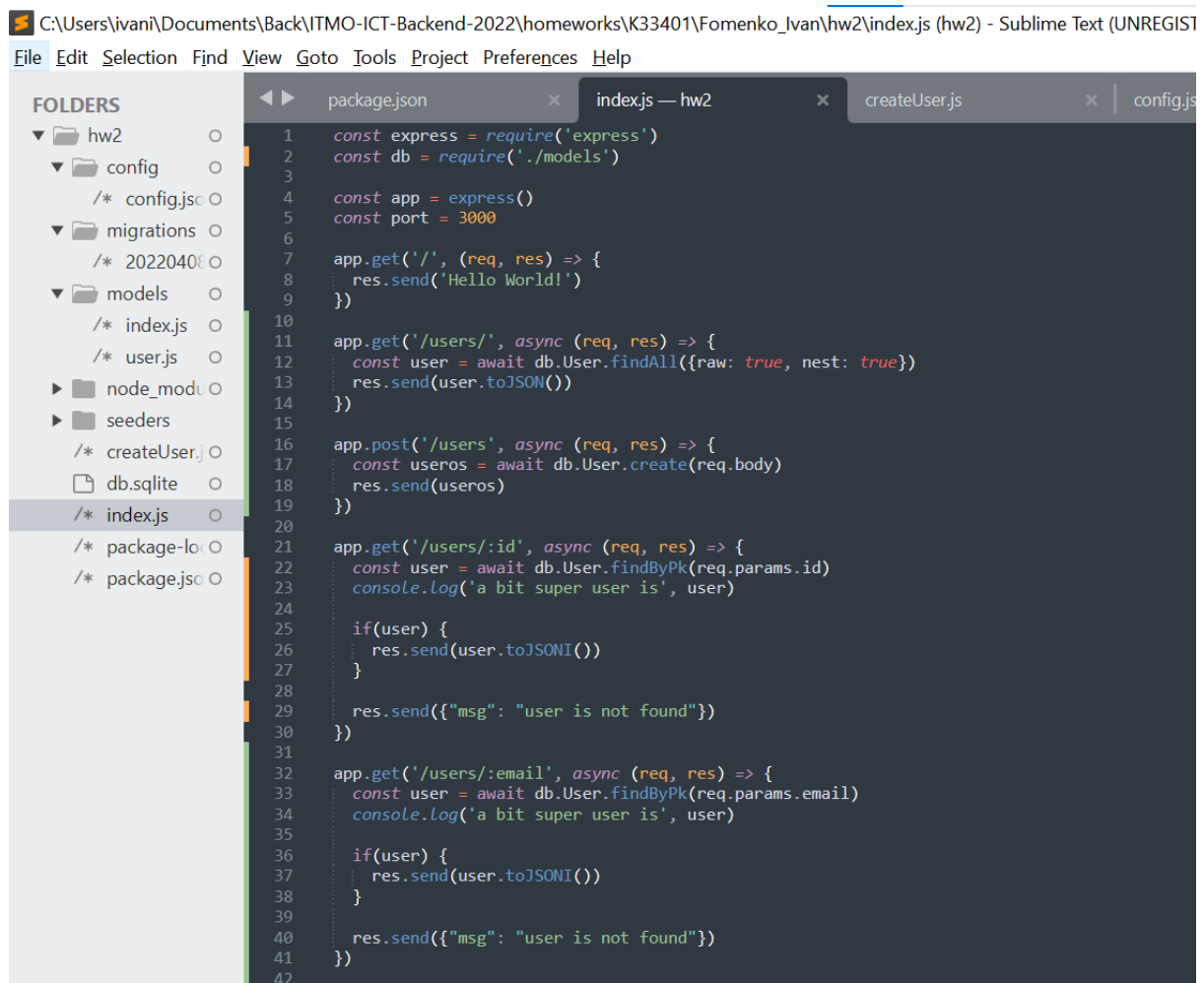
Модель юзера, добавил возраст, чтобы не продавать кальяны молодым и старым (если реализовывать подобный сервис)



The screenshot shows the Sublime Text editor with the file `user.js` open. The file is part of a project named `hw2`. The code defines a Sequelize model for a user with fields `firstName`, `lastName`, `email`, and `userAge`. It also includes a static `associate` method and a `user.init` function.

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

Методы

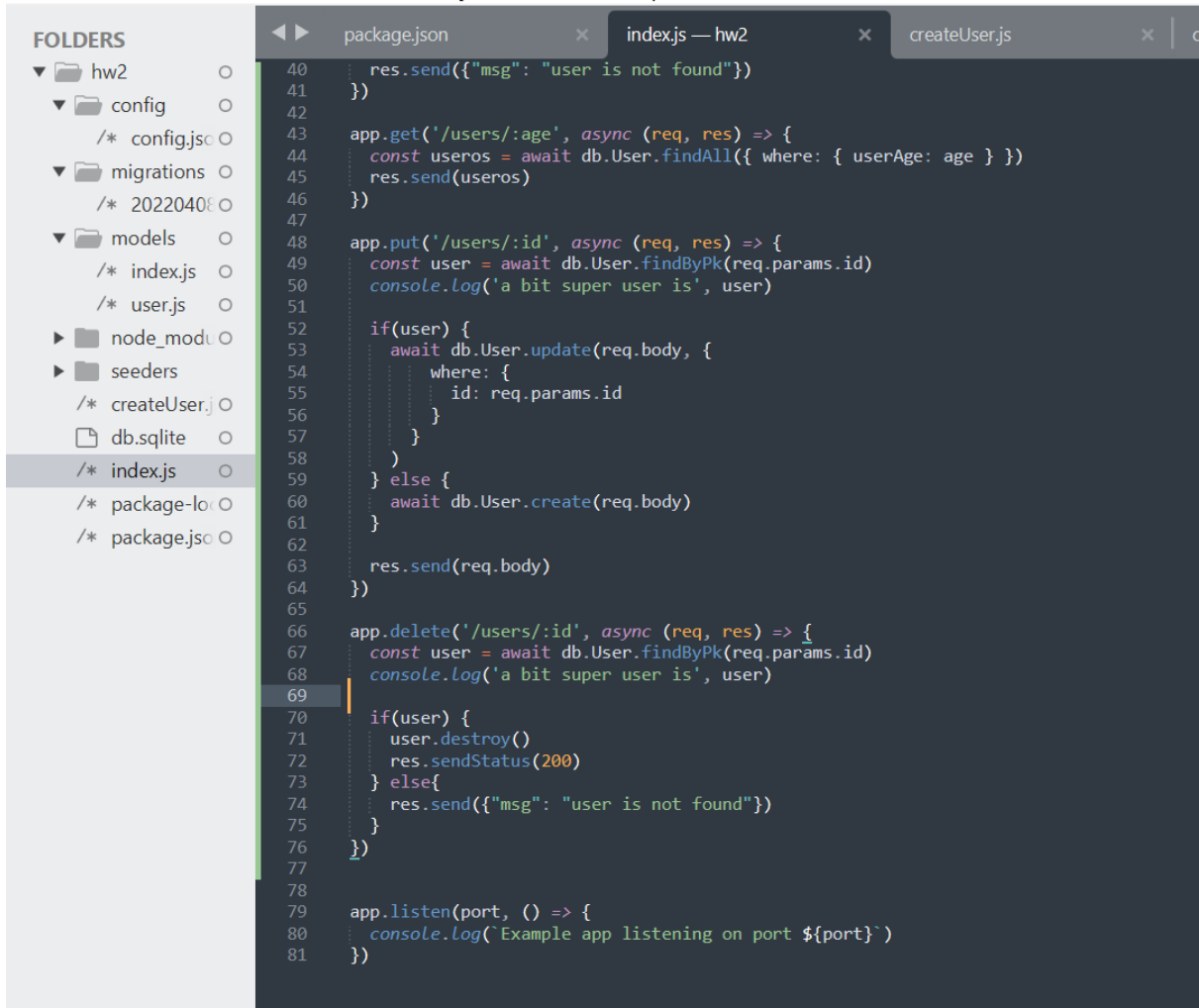


The screenshot shows the Sublime Text editor with the file `index.js` open. The file is part of a project named `hw2`. The code defines an Express application with routes for `/`, `/users/`, `/users/:id`, and `/users/:email`. It uses the `sequelize` database to interact with the `User` model.

```
1 const express = require('express')
2 const db = require('./models')
3
4 const app = express()
5 const port = 3000
6
7 app.get('/', (req, res) => {
8   res.send('Hello World!')
9 })
10
11 app.get('/users/', async (req, res) => {
12   const user = await db.User.findAll({raw: true, nest: true})
13   res.send(user.toJSON())
14 })
15
16 app.post('/users', async (req, res) => {
17   const users = await db.User.create(req.body)
18   res.send(users)
19 })
20
21 app.get('/users/:id', async (req, res) => {
22   const user = await db.User.findByPk(req.params.id)
23   console.log('a bit super user is', user)
24
25   if(user) {
26     res.send(user.toJSON())
27   }
28
29   res.send({msg: "user is not found"})
30 })
31
32 app.get('/users/:email', async (req, res) => {
33   const user = await db.User.findByPk(req.params.email)
34   console.log('a bit super user is', user)
35
36   if(user) {
37     res.send(user.toJSON())
38   }
39
40   res.send({msg: "user is not found"})
41 })
42
```

C:\Users\ivani\Documents\Back\ITMO-ICT-Backend-2022\homeworks\K33401\Fomenko_Ivan\hw2\index.js (hw2) - Sublime Text (UN

File Edit Selection Find View Goto Tools Project Preferences Help



```
40 res.send({ "msg": "user is not found" })
41 })
42
43 app.get('/users/:age', async (req, res) => {
44   const users = await db.User.findAll({ where: { userAge: age } })
45   res.send(users)
46 })
47
48 app.put('/users/:id', async (req, res) => {
49   const user = await db.User.findById(req.params.id)
50   console.log('a bit super user is', user)
51
52   if(user) {
53     await db.User.update(req.body, {
54       where: {
55         id: req.params.id
56       }
57     })
58   }
59   } else {
60     await db.User.create(req.body)
61   }
62
63   res.send(req.body)
64 })
65
66 app.delete('/users/:id', async (req, res) => {
67   const user = await db.User.findById(req.params.id)
68   console.log('a bit super user is', user)
69
70   if(user) {
71     user.destroy()
72     res.sendStatus(200)
73   } else {
74     res.send({ "msg": "user is not found" })
75   }
76 })
77
78
79 app.listen(port, () => {
80   console.log(`Example app listening on port ${port}`)
81 })
```

Вывод

Минимальный пользователь готов