

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет по
домашней работе “Знакомство с микрофреймворком
Express и ORM Sequelize”

Выполнил:
Пронина Мария

Группа:
К33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задание:

Продумать модель пользователя, реализовать набор из CRUD-методов для работы с пользователями средствами Express + Sequelize, написать запрос для получения пользователя по id/email.

Ход работы:

1. Инициализация проекта:

```
D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (hw2)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2\package.json:
{
  "name": "hw2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes
npm notice
npm notice New minor version of npm available! 10.4.0 -> 10.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.0
npm notice Run npm install -g npm@10.5.0 to update!
npm notice
D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2>_
```

2. Установка express, sequelize, sequelize-cli

```
D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2>npm install express
added 64 packages, and audited 65 packages in 11s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2>npm install sequelize
added 23 packages, and audited 88 packages in 13s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2>npm install sequelize-cli
added 94 packages, and audited 182 packages in 16s

29 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

3. Создание проекта:

```
PS D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2> npx sequelize init

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

Created "config\config.json"
Successfully created models folder at "D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2\models".
Successfully created migrations folder at "D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2\migrations".
Successfully created seeders folder at "D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2\seeders".
PS D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\K33392\Пронина Мария\hw2>
```

4. Изменение СУБД с mysql на sqlite:

```
{ } config.json ×
config > { } config.json > { } development
1 {
2   "development": {
3     "username": "root",
4     "password": null,
5     "database": "database_development",
6     "host": "127.0.0.1",
7     "dialect": "sqlite",
8     "storage": "db.sqlite"
9   },
```

5. Создание модели пользователя с параметрами имя, фамилия, номер телефона, электронная почта и статус уведомлений (включены/выключены) и проведение миграции:

```
PS D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\У33392\Пронина Мария\hw2> npx sequelize-cli model:generate --name User --attributes firstName:string,secondName:string,phone:string,email:string,notifications:boolean
Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

New model was created at D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\У33392\Пронина Мария\hw2\models\User.js .
New migration was created at D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\У33392\Пронина Мария\hw2\migrations\20240314200958-create-user.js .
PS D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\У33392\Пронина Мария\hw2>
PS D:\itmo\backend\ITMO-ICT-Backend-2024\homeworks\У33392\Пронина Мария\hw2> npx sequelize db:migrate
Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

Loaded configuration file "config/config.json".
Using environment "development".
== 20240314200958-create-user: migrating =====
== 20240314200958-create-user: migrated (0.319s)
```

6. Выделение порта, на котором будет работать сервер:

```
const port = 3000

app.listen(port, () => {
  console.log("listening...")
})
```

7. Реализация эндпоинта для просмотра всех пользователей:

```
app.get('/users', async (req, res) => {
  try {
    data = await db.User.findAll()
    return res.send(data)
  }
  catch(error) {
    return res.status(500).json({ message: error.message });
  }
})
```

Просмотр пользователей:

GET

http://localhost:3000/users

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
	Key	Value

Body

Cookies

Headers (7)

Test Results

200 OK 9 ms 433 B

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "id": 9,
3    "firstName": "Ivan",
4    "secondName": "Ivanov",
5    "phone": "7981111111",
6    "email": "ivan@gmail.com",
7    "notifications": true,
8    "createdAt": "2024-03-14T21:24:22.025Z",
9    "updatedAt": "2024-03-14T21:24:22.025Z"
10 }
11
12
```

8. Реализация эндпоинта для создания нового пользователя:

```
app.post('/users/new', async (req, res) => {
  try {
    const data = await db.User.create(req.body)
    return res.send(data)
  }
  catch(error) {
    return res.status(400).json({ message: error.message });
  }
})
```

Добавление пользователя:

The screenshot shows a REST client interface. At the top, a POST request is configured to `http://localhost:3000/users/new`. The 'Body' tab is selected, and the request body is a JSON object: `{ "firstName": "Ivan", "secondName": "Ivanov", "phone": "7981111111", "email": "ivan@gmail.com", "notifications": true }`. Below the request, the 'Response' tab is selected, showing a 200 OK status with a response body: `{ "id": 9, "firstName": "Ivan", "secondName": "Ivanov", "phone": "7981111111", "email": "ivan@gmail.com", "notifications": true, "updatedAt": "2024-03-14T21:24:22.025Z", "createdAt": "2024-03-14T21:24:22.025Z" }`. The response is formatted as JSON.

```
POST http://localhost:3000/users/new

{
  "firstName": "Ivan",
  "secondName": "Ivanov",
  "phone": "7981111111",
  "email": "ivan@gmail.com",
  "notifications": true
}
```

Body Cookies Headers (7) Test Results 200 OK 188 ms 431 B

```
{
  "id": 9,
  "firstName": "Ivan",
  "secondName": "Ivanov",
  "phone": "7981111111",
  "email": "ivan@gmail.com",
  "notifications": true,
  "updatedAt": "2024-03-14T21:24:22.025Z",
  "createdAt": "2024-03-14T21:24:22.025Z"
}
```

9. Реализация эндпоинта для удаления пользователя:

```
app.delete('/users/delete/:id', async (req, res) => {
  try {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
      await user.destroy()
      return res.status(200).send()
    }
    return res.status(404).json({ error: 'User not found' })
  }
  catch (error) {
    res.status(500).json({ error: error.message })
  }
})
```

Удаление пользователя:

DELETE ▼ http://localhost:3000/users/delete/3

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params


	Key	Value
	Key	Value

Body

Cookies

Headers (5)

Test Results

 200 OK 481 ms 145 B

10. Реализация эндпоинта для просмотра информации о пользователе по id:

```
app.get('/users/:id', async(req, res) => {
  try {
    const data = await db.User.findByPk(req.params.id)
    if (data) {
      return res.send(data)
    }
    return res.status(404).json({ error: 'User not found' })
  }
  catch(error) {
    return res.status(500).json({ message: error.message });
  }
})
```

Просмотр информации:

GET ⌵ http://localhost:3000/users/9

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

	Key	Value
	Key	Value

Body Cookies Headers (7) Test Results 🌐 200 OK 34 ms 431 B S

Pretty Raw Preview Visualize JSON ⌵ 🔗

```
1  {
2    "id": 9,
3    "firstName": "Ivan",
4    "secondName": "Ivanov",
5    "phone": "7981111111",
6    "email": "ivan@gmail.com",
7    "notifications": true,
8    "createdAt": "2024-03-14T21:24:22.025Z",
9    "updatedAt": "2024-03-14T21:24:22.025Z"
10 }
```

11. Реализация эндпоинта для обновления информации о пользователе:

```
app.patch('/users/update/:id', async(req, res) => {
  try {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
      const data = await user.update(req.body)
      return res.send(data)
    }
    return res.status(404).json({ error: 'User not found' })
  }
  catch(error){
    return res.status(500).json({ message: error.message });
  }
})
```


Обновление информации о пользователе:

The screenshot displays a REST client interface. At the top, a PATCH request is configured for the URL `http://localhost:3000/users/update/9`. The 'Body' tab is selected, showing a JSON payload: `{ "email": "ivanov@gmail.com" }`. Below the request, the response is shown in the 'Body' tab, displaying a JSON object: `{ "id": 9, "firstName": "Ivan", "secondName": "Ivanov", "phone": "7981111111", "email": "ivanov@gmail.com", "notifications": true, "createdAt": "2024-03-14T21:24:22.025Z", "updatedAt": "2024-03-14T22:19:28.542Z" }`. The status bar indicates a 200 OK response with a 205 ms duration and 433 B of data.

```
PATCH http://localhost:3000/users/update/9
```

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "email": "ivanov@gmail.com"
3 }
```

Body Cookies Headers (7) Test Results 200 OK 205 ms 433 B S

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 9,
3   "firstName": "Ivan",
4   "secondName": "Ivanov",
5   "phone": "7981111111",
6   "email": "ivanov@gmail.com",
7   "notifications": true,
8   "createdAt": "2024-03-14T21:24:22.025Z",
9   "updatedAt": "2024-03-14T22:19:28.542Z"
10 }
```

Вывод

В результате выполнения работы были освоены основы express и sequelize и реализованы CRUD-методы для работы с пользователями.