

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №2

Выполнил:
Шутов Даниил

Группа:
К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2022 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

1. Создана модель пользователя

```
1      const db = require('./models')
2
3      async function main() {
4          await db.User.create({
5              firstName: 'Danya',
6              lastName: 'Shutov',
7              email: 'danya@example.com',
8              age: 21,
9              country: 'Russia',
10             city: 'Spb'
11         })
12     }
13
14     main()
```

2. CRUD-методы для работы с пользователем:

```
1  const express = require('express')
2  const db = require('./models')
3
4  const app = express()
5  const port = 3000
6  const bodyParser = require('body-parser');
7  app.use(bodyParser.json());
8
9
10 app.post( path: '/users', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
11     const user = await db.User.create(req.body)
12     res.send(user)
13 })
14
15 app.get('/', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
16     const users = await db.User.findAll()
17     res.send(users)
18 })
19
20 app.get('/users/email', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
21     const user = await db.User.findOne({ where: { email: req.query.email } })
22     if (user) {
23         res.send(user)
24     } else {
25         res.send( body: {"status": "error"})
26     }
27 })
28
29 app.get('/users/id', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
30     const user = await db.User.findById(req.query.id)
31     if (user) {
32         res.send(user)
33     } else {
34         res.send( body: {"status": "error"})
35     }
36 })
37
38 app.delete( path: '/users/:id', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
39     const user = await db.User.destroy({ where: { id: req.params.id } })
40     if (user) {
41         res.send( body: {"status": "user deleted"})
42     } else {
43         res.send( body: {"status": "error"})
44     }
45 })
46
47 app.put( path: '/users/:id', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
48     const num = await db.User.update(req.body, { where: { id: req.params.id } })
49     if (num == 1) {
50         res.send( body: {"status": "user updated"})
51     } else {
52         res.send( body: {"status": "error"})
53     }
54 })
```

3. Пример работы

```
← → 🔄 🌐 localhost:3000
[{"id":1,"firstName":"Test","lastName":"Test","email":"test@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:16:32.609Z","updatedAt":"2022-03-21T14:16:32.609Z"},
{"id":2,"firstName":"Danya","lastName":"Shutov","email":"danya@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:39:19.957Z","updatedAt":"2022-03-21T14:39:19.957Z"},
{"id":4,"firstName":"Max","lastName":"Sidorov","email":"van@mail.ru","country":"USA","city":"NY City","age":30,"createdAt":"2022-03-21T15:02:39.832Z","updatedAt":"2022-03-21T15:06:38.941Z"}]
```

4. Добавление нового пользователя

POST

http://localhost:3000/users

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Beautiful

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {"firstName": "Den", "lastName": "Petrov", "email": "den@mail.ru", "age": 20, "country": "Rus", "city": "Moscow"}
```

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 85 ms

Size: 400 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 5,
3   "firstName": "Den",
4   "lastName": "Petrov",
5   "email": "den@mail.ru",
6   "age": 20,
7   "country": "Rus",
8   "city": "Moscow",
9   "updatedAt": "2022-03-21T15:23:02.803Z",
10  "createdAt": "2022-03-21T15:23:02.803Z"
}
```

Результат

```
[{"id":1,"firstName":"Test","lastName":"Test","email":"test@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:16:32.609Z","updatedAt":"2022-03-21T14:16:32.609Z"}, {"id":2,"firstName":"Danya","lastName":"Shutov","email":"danya@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:39:19.957Z","updatedAt":"2022-03-21T14:39:19.957Z"}, {"id":4,"firstName":"Max","lastName":"Sidorov","email":"van@mail.ru","country":"USA","city":"NY City","age":30,"createdAt":"2022-03-21T15:02:39.832Z","updatedAt":"2022-03-21T15:06:38.941Z"}, {"id":5,"firstName":"Den","lastName":"Petrov","email":"den@mail.ru","country":"Rus","city":"Moscow","age":20,"createdAt":"2022-03-21T15:23:02.803Z","updatedAt":"2022-03-21T15:23:02.803Z"}]
```

5. Удаление нового пользователя

DELETE

http://localhost:3000/users/4

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Beautiful

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {"firstName": "Den", "lastName": "Petrov", "email": "den@mail.ru", "age": 20, "country": "Rus", "city": "Moscow"}
```

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 35 ms

Size: 237 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "status": "user deleted"
}
```

Результат

```
[{"id":1,"firstName":"Test","lastName":"Test","email":"test@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:16:32.609Z","updatedAt":"2022-03-21T14:16:32.609Z"}, {"id":2,"firstName":"Danya","lastName":"Shutov","email":"danya@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:39:19.957Z","updatedAt":"2022-03-21T14:39:19.957Z"}, {"id":5,"firstName":"Den","lastName":"Petrov","email":"den@mail.ru","country":"Rus","city":"Moscow","age":20,"createdAt":"2022-03-21T15:23:02.803Z","updatedAt":"2022-03-21T15:23:02.803Z"}]
```

6. Обновление информации о пользователе

The screenshot shows the Postman interface for a PUT request. The URL is `http://localhost:3000/users/5`. The request body is a JSON object: `{ "firstName": "Den", "lastName": "KS", "email": "den@mail.ru", "age": 20, "country": "Rus", "city": "Moscow" }`. The response status is 200 OK, with a time of 22 ms and a size of 237 B. The response body is `{ "status": "user updated" }`.

```
1 { "firstName": "Den", "lastName": "KS", "email": "den@mail.ru", "age": 20, "country": "Rus", "city": "Moscow" }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 22 ms Size: 237 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "user updated"
3 }
```

Результат:

```
[{"id":1,"firstName":"Test","lastName":"Test","email":"test@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:16:32.609Z","updatedAt":"2022-03-21T14:16:32.609Z"}, {"id":2,"firstName":"Danya","lastName":"Shutov","email":"danya@example.com","country":"Russia","city":"Spb","age":21,"createdAt":"2022-03-21T14:39:19.957Z","updatedAt":"2022-03-21T14:39:19.957Z"}, {"id":5,"firstName":"Den","lastName":"KS","email":"den@mail.ru","country":"Rus","city":"Moscow","age":20,"createdAt":"2022-03-21T15:23:02.803Z","updatedAt":"2022-03-21T15:26:08.445Z"}]
```

Вывод

В данном домашнем задании удалось освоить CRUD-методы для работы с пользователями средствами Express + Sequelize. Были проведены запросы по добавлению, удалению и обновлению информации о пользователе с помощью Postman.