

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа 2

Выполнила:
Самчук Анита
К34402

Проверил:
Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Знакомство с ORM Sequelize

Ход работы

1. Созданная модель пользователя:

```
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    2 usages
7    class User extends Model {
8      /**
9       * Helper method for defining associations.
10      * This method is not a part of Sequelize lifecycle.
11      * The `models/index` file will call this method automatically.
12      */
13      2 usages
14      static associate(models) {
15        // define association here
16      }
17    }
18    User.init({ attributes: {
19      firstName: DataTypes.STRING,
20      lastName: DataTypes.STRING,
21      email: DataTypes.STRING,
22      phoneNumber: DataTypes.STRING,
23      password: DataTypes.STRING,
24      username: DataTypes.STRING
25    }, options: {
26      sequelize,
27      modelName: 'User',
28    }
29  });
30  return User;
31  };
```

2. Реализованные CRUD методы:

```
11 app.get('/users', async (req, res) : Promise<any> => {
12     const users : Model[] = await db.user.findAll()
13
14     console.log(users)
15
16     if (users) {
17         return res.status(200).send(users.toJSON())
18     }
19
20     return res.send({"msg": "no users"})
21 })
22
23 app.post('/users/', async (req, res) : Promise<any> => {
24     const user = await db.user.create(req.body)
25
26     console.log('New user', user)
27
28     if (user) {
29         return res.status(201).send(user.toJSON())
30     }
31
32     return res.send({"msg": "failed"})
33 })
```

```
35 app.delete('/users/:id', async (req, res) : Promise<any> => {
36     const deleted = await db.user.destroy({ where: { id: req.params.id } })
37
38     console.log('user is deleted')
39
40     if (deleted) {
41         return res.status(200).send({"msg": "user is deleted"})
42     }
43
44     return res.status(404).send({"msg": "user is not found"})
45 })
46
47 app.put('/users/:id', async (req, res) : Promise<any> => {
48     const user = await db.user.update(req.body, { where: { id: req.params.id } })
49
50     console.log('user is updated')
51
52     if (user) {
53         return res.status(200).send({"msg": "user is updated"})
54     }
55
56     return res.status(404).send({"msg": "user is not found"})
57 })
```

3. Юзер по id и email

```
59 app.get('/users/:id', async (req, res) : Promise<any> => {
60   const user : Promise<...> | Promise<...> = await db.user.findByPk(req.params.id)
61
62   console.log('user is', user)
63
64   if (user) {
65     return res.status(200).send(user.toJSON())
66   }
67
68   return res.status(404).send({ "msg": "user is not found" })
69 })
70
71 app.get('/users/:email', async (req, res) : Promise<any> => {
72   const user : Promise<...> | Promise<...> = await db.user.findByPk(req.params.email)
73
74   console.log('user is', user)
75
76   if (user) {
77     return res.status(200).send(user.toJSON())
78   }
79
80   return res.status(404).send({ "msg": "user is not found" })
81 })
```

Вывод

Освоена ORM Sequelize и создана модель пользователя и необходимые CRUD-методы для работы с моделью