

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет

Домашнее задание 2

Выполнил:

Дао Куанг Ань

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.index.js

```
const express = require('express')

const db = require('./models')

const bp = require('body-parser')

const app = express()

const port = 3000

app.use(bp.json())

app.use(bp.urlencoded({ extended: true }))

app.get('/', (req, res) => {

  res.send('hello world')

})

app.get('/user/:id', async (req, res) => {

  const user = await db.User.findByPk(req.params.id)

  console.log('user is', user)

  if (user) {

    res.send(user.toJSON())

  }

})

app.get('/user_email/:email', async (req, res) => {
```

```
const user = await db.User.findOne({
  where: {
    email: req.params.email
  }
})

if (user) {
  res.send(user.toJSON())
}
})

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

app.delete('/delete/:id', async (req, res) => {
  const user = await db.User.destroy({
    where: {
      id: req.params.id
    }
  })

  if (user) {
    res.send({"msg": "user deleted"})
  }

  else {
    res.send({"msg": "user not found"})
  }
})
```

```

app.put('/update/:id', async (req,res) => {

  db.User.findByPk(req.params.id)

  .then(result => {

    if (result) {

      let val = {

        firstName: req.body.firstName,

        lastName: req.body.lastName,

        email: req.body.email

      }

      result.update(val)

    } else { res.send({"msg": "user not found"}) }

    res.send(req.body)

  })

})

app.post('/create', async (req,res) => {

  await db.User.create(req.body)

  res.send({"msg":"user created"})

})

app.listen(port, () => {

  console.log(`Example app listening on port ${port}, and host - localhost`)

})

```

2. user.js

```

'use strict';

const {

  Model

} = require('sequelize');

```

```
module.exports = (sequelize, DataTypes) => {  
  
  class User extends Model {  
  
    /**  
     * Helper method for defining associations.  
     * This method is not a part of Sequelize lifecycle.  
     * The `models/index` file will call this method automatically.  
     */  
  
    static associate(models) {  
      // define association here  
    }  
  }  
  
  User.init({  
    firstName: DataTypes.STRING,  
    lastName: DataTypes.STRING,  
    email: DataTypes.STRING  
  }, {  
    sequelize,  
    modelName: 'User',  
  });  
  
  return User;  
};
```

Вывод

- Посторил свою собственную модель пользователя, реализовал набор из CRUD-методов для работы с пользователями средствами Express + Sequelize, написал запрос для получения пользователя по id/email