

"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"

Факультет «Инфокоммуникационных технологий»

Направление подготовки «09.03.03 Прикладная информатика»

Бакалаврская программа «Мобильные и сетевые технологии»

ОТЧЕТ
по практической работе 2
по дисциплине «Backend разработка»

Тема

Выполнил(и)

___Тихонов Л.А.____ / _____
(подпись) (Фамилия И.О.)

Проверил

_____ / Добряков Д.И.

Дата _____

Санкт-Петербург
2022

Задачи:

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:

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!')
})

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

app.get('/users/email/:email', async (req, res) => {
  const user = await db.User.findOne({ where: { email: req.params.email } })
  console.log('user is', user)
  res.send(user.toJSON())
})

app.get('/users/:id', async (req, res) => {
  const user = await db.User.findByPk(req.params.id)
  console.log('user is', user)
  res.send(user.toJSON())
})

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

```

app.put('/users/update/:id', async (req, res) => {
  db.User.findOne({ where: {id: req.params.id}})
    .then(record => {
      if (!record) {
        throw new Error('No record found')
      }
      let values = {
        firstName: req.body.firstName,
        lastName: req.body.lastName,
        email: req.body.email,
      }

      record.update(values).then( updatedRecord => {
        // login into your DB and confirm update
      })

    })
    .catch((error) => {
      //error
      throw new Error(error)
    })
    console.log('user was update')
    res.send('user was update')
  })

app.post ('/create/', async (req,res) => {
  await db.User.create(req.body)
  console.log('user was create')
  res.send('user was create')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})

```

createUser.js

```

const db = require('./models')

async function main() {
  await db.User.create({
    firstName: 'Test1',
    lastName: 'Test1',
    email: 'crond@example.com'
  })
}

main()

```

Вывод:

В ходе работы я научился создавать простых пользователей с помощью Nodejs, используя инструменты Express и Sequelize для создания CRUD-методов для работы с пользователями.