

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашнее задание №2

Выполнил:

Дорофеева Арина

Группа к33401

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

## Задача

1. Продумать свою собственную модель пользователя
2. Реализовать набор из CRUD-методов для работы с пользователями средствами Express + Sequelize
3. Написать запрос для получения пользователя по id / email

## Ход работы

1. Модель пользователя

```
User.init({ attributes: {  
  firstName: DataTypes.STRING,  
  lastName: DataTypes.STRING,  
  phone: DataTypes.STRING,  
  nickName: DataTypes.STRING  
}, options: {  
  sequelize,  
  modelName: 'User',  
}},  
return User;  
};
```

2. CRUD-методы и получение пользователя по id / phone / nickname

```
app.post('/users', handlers.async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {  
  const user = await db.User.create(req.body)  
  res.send(user.toJSON())  
})  
  
app.get('/users/', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {  
  const userList = await db.User.findAll({options: {raw: true, nest: true}})  
  res.send(userList)  
})  
  
app.get('/users/:id', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {  
  const user = await db.User.findByPk(req.params.id)  
  console.log('user is', user)  
  
  if (user) {  
    res.send(user.toJSON())  
  }  
  
  res.send({body: {"msg": "user is not found"}})  
})
```

```

app.get('/users/:phone', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  const user = await db.User.findByPk(req.params.phone)
  console.log('user is', user)

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

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

app.get('/users/:nickname', async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  const user = await db.User.findByPk(req.params.nickname)
  console.log('user is', user)

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

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

```

```

app.put( path: '/users/:id', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  const user = await db.User.findByPk(req.params.id)
  if (user) {
    await db.User.update(req.body, {
      where: {
        id: req.params.id
      }
    })
  } else {
    await db.User.create(req.body)
  }
  res.send(req.body)
})

app.delete( path: '/users/:id', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  const user = await db.User.findByPk(req.params.id)
  if (user) {
    user.destroy();
    res.sendStatus( code: 200 )
  }
  else res.send( body: { "msg": "user is not found" })
})

```

## Вывод

Я продумала модель пользователя для своего сервиса с аксолотлями и реализовала набор методов для работы с пользователями средствами Express + Sequelize.