

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа 2

«Знакомство с ORM Sequelize»

Выполнил:

Галиновский Роман

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

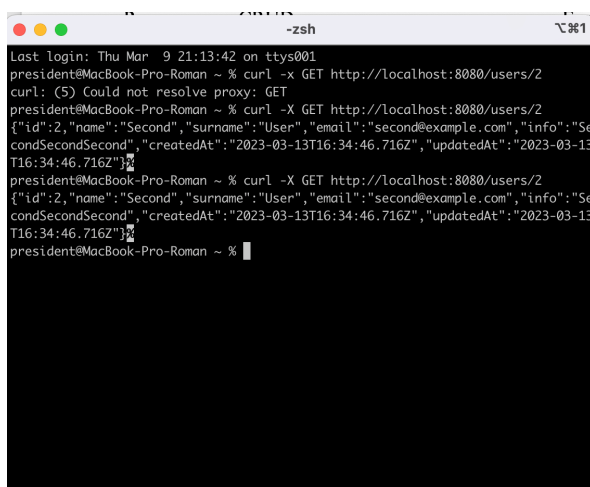
1) Создана модель пользователя, что состоит из таких полей, как:

1. Имя
2. Фамилия
3. Почта
4. Информация

```
User.init({
  name: DataTypes.STRING,
  surname: DataTypes.STRING,
  email: DataTypes.STRING,
  info: DataTypes.STRING
}, {
  sequelize,
  modelName: 'User',
});
```

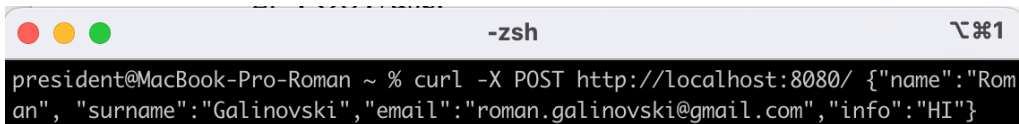
Реализованы CRUD-методы и запросы с помощью Express.js и Sequelize

1. GET/user



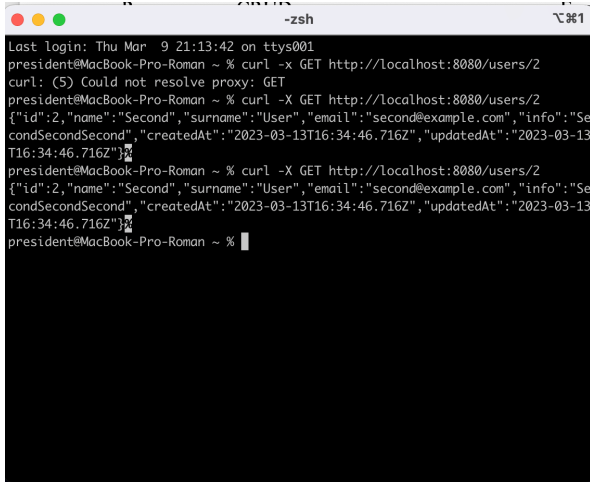
```
-zsh
Last login: Thu Mar  9 21:13:42 on ttys001
president@MacBook-Pro-Roman ~ % curl -x GET http://localhost:8080/users/2
curl: (5) Could not resolve proxy: GET
president@MacBook-Pro-Roman ~ % curl -X GET http://localhost:8080/users/2
{"id":2,"name":"Second","surname":"User","email":"second@example.com","info":"SecondSecondSecond","createdAt":"2023-03-13T16:34:46.716Z","updatedAt":"2023-03-13T16:34:46.716Z"}
president@MacBook-Pro-Roman ~ % curl -X GET http://localhost:8080/users/2
{"id":2,"name":"Second","surname":"User","email":"second@example.com","info":"SecondSecondSecond","createdAt":"2023-03-13T16:34:46.716Z","updatedAt":"2023-03-13T16:34:46.716Z"}
president@MacBook-Pro-Roman ~ %
```

## 2. POST/user



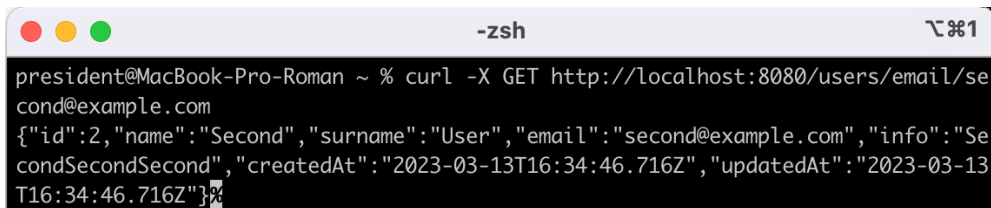
```
president@MacBook-Pro-Roman ~ % curl -X POST http://localhost:8080/ {"name":"Roman", "surname":"Galinovski", "email":"roman.galinovski@gmail.com", "info":"HI"}
```

## 3. GET/user/:id



```
Last login: Thu Mar 9 21:13:42 on ttys001
president@MacBook-Pro-Roman ~ % curl -x GET http://localhost:8080/users/2
curl: (5) Could not resolve proxy: GET
president@MacBook-Pro-Roman ~ % curl -X GET http://localhost:8080/users/2
{"id":2,"name":"Second","surname":"User","email":"second@example.com","info":"SecondSecondSecond","createdAt":"2023-03-13T16:34:46.716Z","updatedAt":"2023-03-13T16:34:46.716Z"}
president@MacBook-Pro-Roman ~ % curl -X GET http://localhost:8080/users/2
{"id":2,"name":"Second","surname":"User","email":"second@example.com","info":"SecondSecondSecond","createdAt":"2023-03-13T16:34:46.716Z","updatedAt":"2023-03-13T16:34:46.716Z"}
president@MacBook-Pro-Roman ~ %
```

## 4. GET/user/email/:email



```
president@MacBook-Pro-Roman ~ % curl -X GET http://localhost:8080/users/email/second@example.com
{"id":2,"name":"Second","surname":"User","email":"second@example.com","info":"SecondSecondSecond","createdAt":"2023-03-13T16:34:46.716Z","updatedAt":"2023-03-13T16:34:46.716Z"}
```

## 5. PUT/user/:id

## 6. DELETE/user/:id

## Вывод

Была реализована модель абстрактного пользователя с CRUD методами к ней, впечатления о Sequelize были произведены, как о не самом удобном инструменте.