

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО**

Дисциплина:

Бекенд разработка

Отчет

по домашней работе №2
Работа с Express и Sequelize

Выполнил:
Прохоров Николай

Группа: K33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2022 г.

Цель работы

Изучить работу с Express и Sequelize

Ход работы:

- 1) Первичная настройка проекта. Ради эксперимента принял решение использовать Typescript, и писать исключительно на нем. Для форматирования кода по принятым стандартам дополнительно установил Prettier.
- 2) Имплементация модели пользователя

```
class User extends Model<InferAttributes<User>, InferCreationAttributes<User>> {  
  declare id: CreationOptional<number>  
  
  declare firstName: string  
  declare lastName: string  
  declare middleName: string | null  
  declare email: string  
  
  declare createdAt: CreationOptional<Date>  
  declare updatedAt: CreationOptional<Date>  
}
```

- 3) Разработка CRUD-методов для пользователя. Ввиду наличия в Sequelize валидаторов «из коробки», решил имплементировать корректную обработку ошибок при POST запросе на создание пользователя. Код и результат представлен ниже:

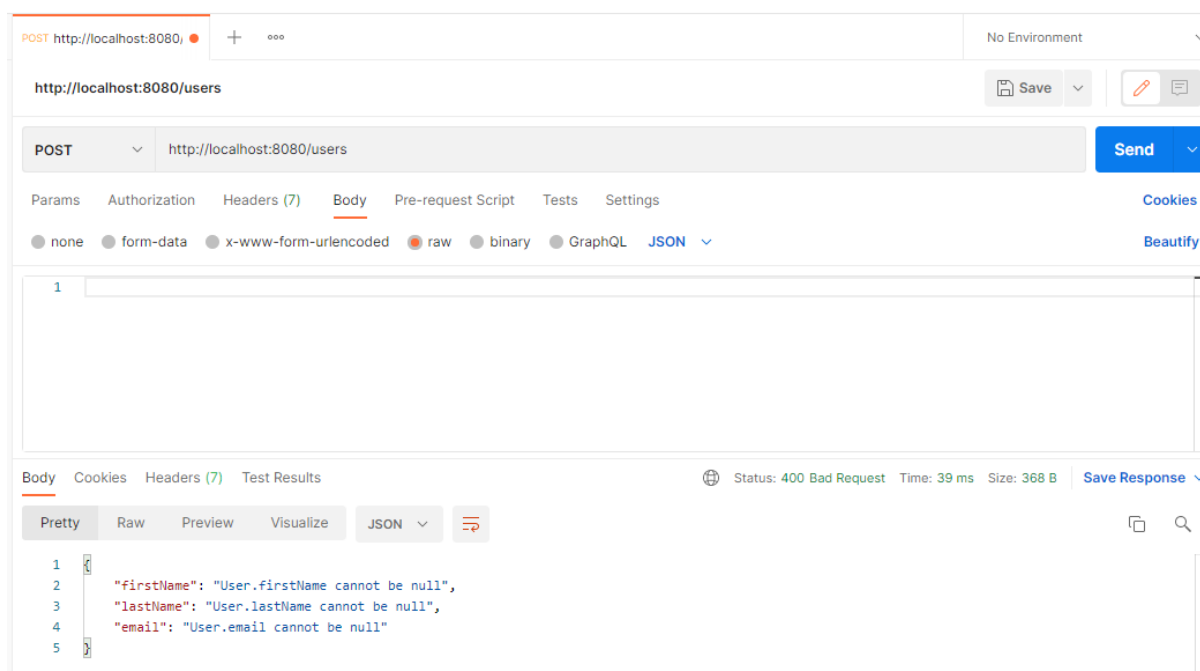
```
app.post( path: '/users', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {  
  try {  
    const user = await User.create(req.body)  
    res.send(user.toJSON())  
  } catch (e) {  
    console.error(e)  
    const error = useSequelizeErrors(e)  
    res.status( code: 400 ).send(error)  
  }  
})
```

```

export function useSequelizeErrors(err: any) {
  if (err.name === 'SequelizeValidationError') {
    const errObj: { [key: string]: any } = {};
    err.errors.map((er: any) => {
      errObj[er.path] = er.message;
    })
    return errObj
  } else {
    return { 'error': 'Неизвестная ошибка!' }
  }
}

```

При попытке отослать неверно заполненную форму, вернутся ошибки в человеко-читаемом формате, позволяющие понять, как именно нужно заполнить поля:



При корректном заполнении полей юзер успешно создается:

POST http://localhost:8080/users

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "firstName": "Nikolay",
3   ... "lastName": "Prokhorov",
4   ... "email": "mail@prokhn.ru"
5 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 44 ms Size: 392 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "firstName": "Nikolay",
4   "lastName": "Prokhorov",
5   "email": "mail@prokhn.ru",
6   "updatedAt": "2022-03-30T01:39:45.816Z",
7   "createdAt": "2022-03-30T01:39:45.816Z"
8 }
```

Другие методы так же успешно обрабатывают. Результат запросов через Postman ниже:

GET http://localhost:8080/users

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "firstName": "Nikolay",
3   ... "lastName": "Prokhorov",
4   ... "email": "mail@prokhn.ru"
5 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 11 ms Size: 412 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "firstName": "Nikolay",
5     "lastName": "Prokhorov",
6     "middleName": null,
7     "email": "mail@prokhn.ru",
8     "createdAt": "2022-03-30T01:39:45.816Z",
9     "updatedAt": "2022-03-30T01:39:45.816Z"
10  }
11 }
```

GET http://localhost:8080/users/1

Save

GET http://localhost:8080/users/1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (7) Test Results Status: 200 OK Time: 8 ms Size: 410 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "firstName": "Nikolay",
4   "lastName": "Prokhorov",
5   "middleName": null,
6   "email": "mail@prokhn.ru",
7   "createdAt": "2022-03-30T01:39:45.816Z",
8   "updatedAt": "2022-03-30T01:39:45.816Z"
9 }
```

Обновление данных с помощью метода PUT:

PUT http://localhost:8080/users/1

Save

PUT http://localhost:8080/users/1 S

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "middleName": "Igorovich"
3 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 27 ms Size: 257 B Save R

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "user updated"
3 }
```

GET http://localhost:8080/L + ... No Environment

http://localhost:8080/users/1 Save

GET http://localhost:8080/users/1 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "middleName": "Igorovich"
3 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 10 ms Size: 417 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "firstName": "Nikolay",
4   "lastName": "Prokhorov",
5   "middleName": "Igorovich",
6   "email": "mail@prokhn.ru",
7   "createdAt": "2022-03-30T01:39:45.816Z",
8   "updatedAt": "2022-03-30T01:41:43.851Z"
9 }
```

Поиск по email:

GET http://localhost:8080/L + ... No Environment

http://localhost:8080/users/email/mail@prokhn.ru Save

GET http://localhost:8080/users/email/mail@prokhn.ru Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

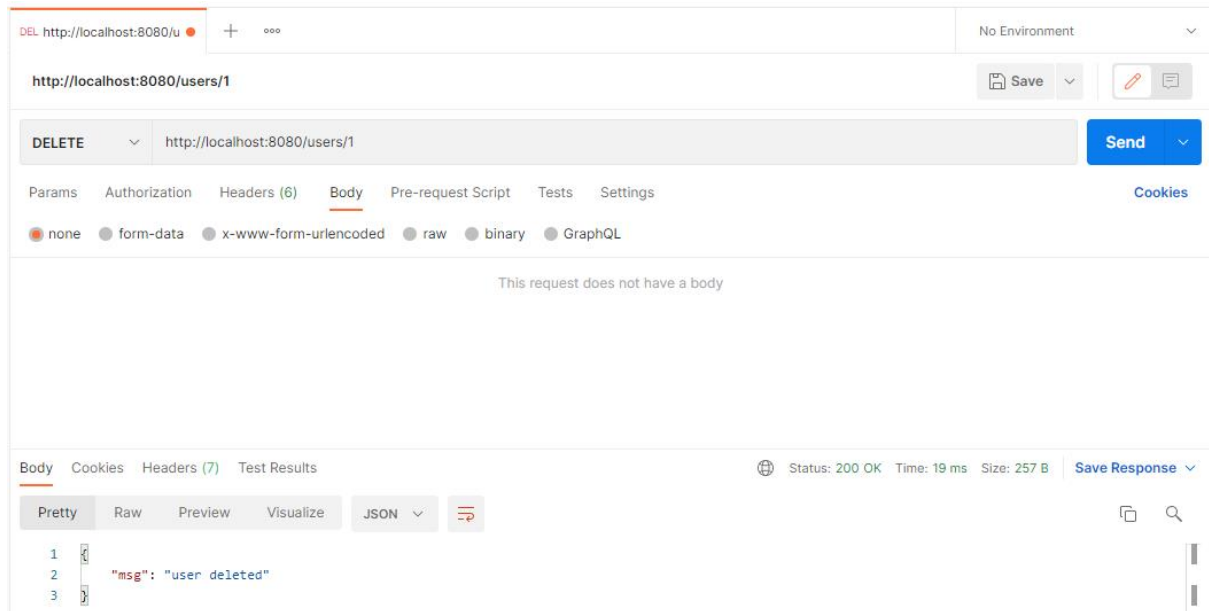
This request does not have a body

Body Cookies Headers (7) Test Results Status: 200 OK Time: 8 ms Size: 417 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "firstName": "Nikolay",
4   "lastName": "Prokhorov",
5   "middleName": "Igorovich",
6   "email": "mail@prokhn.ru",
7   "createdAt": "2022-03-30T01:39:45.816Z",
8   "updatedAt": "2022-03-30T01:41:43.851Z"
9 }
```

Удаление пользователя:



Выводы:

Были освоены основы фреймворка Express и Sequelize ORM. Typescript, в свою очередь, значительно улучшает читаемость кода, обеспечивая подсветку ошибок и синтаксиса.