САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа № 4

Выполнил:

Кононов Степан

Группа

K32392

Проверил: Добряков Д. И.

Санкт-Петербург

2024г.

Документация

Задача

Документирование API из ЛР2 средствами Swagger, Postman

Ход работы

Выполнил документацию в Postman (пример запроса)

POST Создать пользователя http://localhost:3000/users Запрос: Создать нового пользователя. Принимаемые данные: пате: строка, имя пользователя (обязательное поле). email: строка, email пользователя (обязательное поле, должен быть уникальным). password: строка, пароль пользователя (обязательное поле). Возвращаемые данные: id: уникальный идентификатор пользователя. паме: имя пользователя. email: email пользователя. createdAt: дата и время создания пользователя. updatedAt: дата и время последнего обновления пользователя. Body raw (json) 6 json

GET Получить пользователя по ID

Open request →

http://localhost:3000/users/1

Запрос: Получить пользователя по ID.

Принимаемые данные:

• identifier: идентификатор пользователя (число).

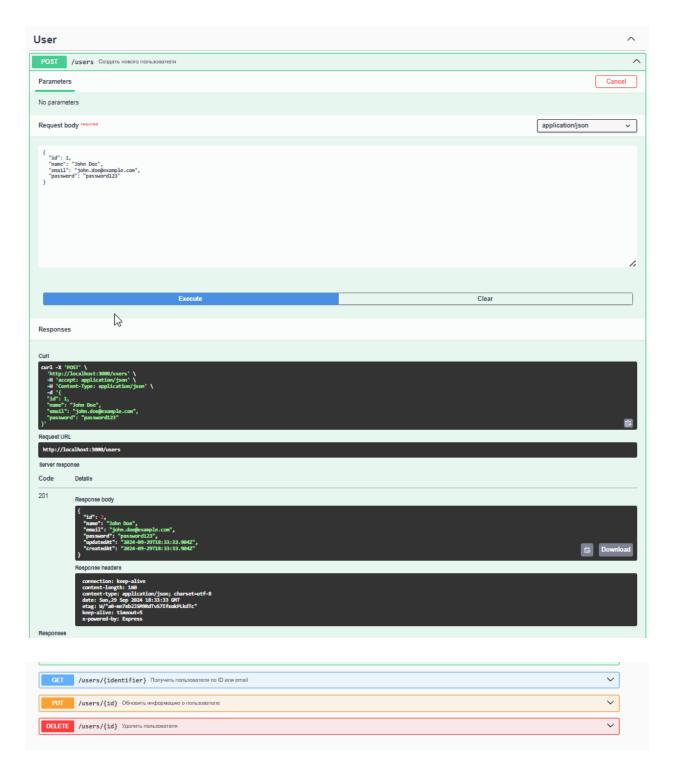
Возвращаемые данные:

- id: уникальный идентификатор пользователя.
- паме: имя пользователя.
- email: email пользователя.
- createdAt: дата и время создания пользователя.
- updatedAt: дата и время последнего обновления пользователя.

Статус 404: Если пользователь с указанным ID не найден.

Документация через Swagger

```
@swagger
  /users/{identifier}:
    get:
      summary: Получить пользователя по ID или email
      tags: [User]
      parameters:
        - in: path
          name: identifier
          schema:
            type: string
          required: true
          description: ID или email пользователя
      responses:
          description: Информация о пользователе
          content:
            application/json:
                 $ref: '#/components/schemas/User'
        404:
          description: Пользователь не найден
router.get('/users/:identifier', async (reg, res) => {
   const identifier = req.params.identifier;
   try {
     let user;
     if (isNaN(identifier)) {
       user = await User.findOne({ where: { email: identifier } });
     } else {
       user = await User.findByPk(identifier);
     }
     if (user) {
       res.json(user);
     } else {
       res.status(404).json({ error: 'User not found' });
    } catch (error) {
     res.status(500).json({ error: error.message });
 });
```



(не стал приводить все методы так как слишком много шаблонного кода)

Вывод

В ходе работы я создал документацию в Swagger и Postman