

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа № 4

Выполнил:

Кононов Степан

Группа

К32392

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

[Open request →](#)

`http://localhost:3000/users`

**Запрос:** Создать нового пользователя.

**Принимаемые данные:**

- `name` : 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.
- `name` : ИМЯ пользователя.
- `email` : email пользователя.
- `createdAt` : дата и время создания пользователя.
- `updatedAt` : дата и время последнего обновления пользователя.

**Body** raw (json)

json



```
{ "name": "John Doe", "email": "john.doe@example.com", "password": "password123" }
```

## 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.
- `name` : 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:
*       200:
*         description: Информация о пользователе
*         content:
*           application/json:
*             schema:
*               $ref: '#/components/schemas/User'
*       404:
*         description: Пользователь не найден
* /
router.get('/users/:identifier', async (req, res) => {
  const identifier = req.params.identifier;
  try {
    let user;
    if (isNaN(identifier)) {
      user = await User.findOne({ where: { email: identifier } });
    } else {
      user = await User.findPk(identifier);
    }

    if (user) {
      res.json(user);
    } else {
      res.status(404).json({ error: 'User not found' });
    }
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

```

User

POST /users Создать нового пользователя

Parameters

Cancel

No parameters

Request body required

application/json

```
{
  "id": 1,
  "name": "John Doe",
  "email": "john.doe@example.com",
  "password": "password123"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:3000/users' \
  -H 'accept: application/json' \
  -H 'Content-type: application/json' \
  -d '{
    "id": 1,
    "name": "John Doe",
    "email": "john.doe@example.com",
    "password": "password123"
  }'
```

Request URL

http://localhost:3000/users

Server response

Code

Details

201

Response body

```
{
  "id": 1,
  "name": "John Doe",
  "email": "john.doe@example.com",
  "password": "password123",
  "updatedAt": "2024-09-29T18:33:33.984Z",
  "createdAt": "2024-09-29T18:33:33.984Z"
}
```

Download

Response headers

```
connection: keep-alive
content-length: 160
content-type: application/json; charset=utf-8
date: Sun, 29 Sep 2024 18:33:33 GMT
etag: W/"a0-m7eb23598dTv571faxkPKdYc"
keep-alive: timeout=5
x-powered-by: Express
```

Responses

GET /users/{identifier} Получить пользователя по ID или email

PUT /users/{id} Обновить информацию о пользователе

DELETE /users/{id} Удалить пользователя

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

## Вывод

В ходе работы я создал документацию в Swagger и Postman

Документация

5