

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №3

Выполнил:

Цой Степан

Группа
К3340

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Реализовать автодокументирование средствами swagger.

Ход работы

Recipe API 1.0.0 OAS 3.0

API для управления рецептами и пользователями

[Contact API](#) [Support](#)

Servers

http://localhost:4000/api - Development server

Authorize 

Authentication Эндпоинты для аутентификации

POST /auth/register Регистрация нового пользователя

 ^

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "username": "john_doe",
  "email": "john@example.com",
  "password": "Password123!"
}
```

Responses

Code	Description	Links
201	<p>Пользователь успешно зарегистрирован</p> <p>Media type: application/json</p> <p>Controls: Accept header.</p> <p>Example Value Schema</p> <pre>{ "message": "User registered successfully" }</pre>	No links
400	<p>Неверные данные или пользователь уже существует</p> <p>Media type: application/json</p> <p>Example Value Schema</p> <pre>{ "message": "User already exists" }</pre>	No links

GET /users Получить список всех пользователей

Доступно только администраторам

Parameters Try it out

Name	Description
email string (query)	Фильтр по email <input type="text" value="email"/>
username string (query)	Фильтр по username <input type="text" value="username"/>

Responses

Code	Description	Links
200	Список пользователей Media type: application/json Controls Accept header. Example Value Schema <pre>{ "user_id": 1, "username": "john_doe", "email": "john@example.com", "created_at": "2023-01-01T00:00:00.000Z", "role": "user"}</pre>	No links
401	Не авторизован	No links
403	Недостаточно прав	No links

GET /users/search Поиск пользователя по email или username

Parameters Try it out

Name	Description
email string(email) (query)	Email пользователя <input type="text" value="email"/>
username string (query)	Username пользователя <input type="text" value="username"/>

Responses

Code	Description	Links
200	Найденный пользователь Media type: application/json Controls Accept header. Example Value Schema <pre>{ "user_id": 1, "username": "john_doe", "email": "john@example.com", "created_at": "2023-01-01T00:00:00.000Z", "role": "user"}</pre>	No links
400	Не указаны параметры поиска	No links
404	Пользователь не найден	No links

PUT

/users/{id}

Обновить данные пользователя

Parameters

Try it out

Name	Description
id <small>* required</small> integer (path)	ID пользователя <input type="text" value="id"/>

Request body required

application/json

Example Value | Schema

```
{  "username": "new_username",  "email": "new@example.com",  "bio": "Новая биография пользователя"}
```

Responses

Code	Description	Links
200	Обновленный пользователь <div>Media type application/json</div> <div>Controls Accept header</div> <div>Example Value Schema</div> <div><pre>{ "user_id": 1, "username": "john_doe", "email": "john@example.com", "profile_photo": "https://example.com/photo.jpg", "bio": "John's romances", "role": "user", "created_at": "2023-01-01T00:00:00.000Z"}</pre></div>	No links
404	Пользователь не найден	No links

CRUD Базовые CRUD операции для всех сущностей

GET /crud/{entity} Получить список сущностей

Try it out

Name	Description
entity <small>required</small> string (path)	Тип сущности (users, recipes, ingredients, etc.) <input type="text" value="entity"/>
page integer (query)	Номер страницы Default value : 1 <input type="text" value="1"/>
limit integer (query)	Количество элементов на странице Default value : 10 <input type="text" value="10"/>

Responses

Code	Description	Links
200	Список сущностей <div>Media type application/json Controls Accept header. Example Value Schema <pre>{ "data": [()], "pagination": { "page": 1, "limit": 10, "total": 100, "pages": 10 }}</pre></div>	No links

POST /crud/{entity} Создать новую сущность

Доступно только администраторам

Try it out

Name	Description
entity <small>required</small> string (path)	Тип сущности <input type="text" value="entity"/>

Request body required

application/json

Example Value | Schema

```
()
```

Responses

Code	Description	Links
201	Сущность создана	No links
400	Неверные данные	No links
403	Недостаточно прав	No links

GET

/crud/{entity}/{id}

Получить сущность по ID

Parameters

Try it out

Name	Description
entity <small>required</small> string <i>(path)</i>	Тип сущности
id <small>required</small> integer <i>(path)</i>	ID сущности

Responses

Code	Description	Links
200	Сущность найдена	No links
404	Сущность не найдена	No links

PUT

/crud/{entity}/{id}

Обновить сущность

Доступно только администраторам

Parameters

Try it out

Name	Description
entity <small>required</small> string <i>(path)</i>	Тип сущности
id <small>required</small> integer <i>(path)</i>	ID сущности

Request body required

application/json

Example Value | Schema

Responses

Code	Description	Links
200	Сущность обновлена	No links
403	Недостаточно прав	No links
404	Сущность не найдена	No links

DELETE /crud/{entity}/{id} Удалить сущность

Доступно только администраторам

Parameters

Try it out

Name	Description
entity * required string (path)	Тип сущности
id * required integer (path)	ID сущности

Responses

Code	Description	Links
204	Сущность удалена	No links
403	Недостаточно прав	No links
404	Сущность не найдена	No links

Schemas

User {
 user_id > [...]
 username > [...]
 email > [...]
 profile_photo > [...]
 bio > [...]
 role > [...]
 created_at > [...]
}

UserProfile {
 user_id > [...]
 username > [...]
 email > [...]
 created_at > [...]
 role > [...]
}

Pagination {
 page > [...]
 limit > [...]
 total > [...]
 pages > [...]
}

ErrorResponse {
 message > [...]
 error > [...]
}

Вывод

Нами было реализовано автодокументирование средствами swagger.