

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №3

Выполнил:

Борисова Элина

Группа К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- реализовать автодокументирование средствами swagger;
- реализовать документацию API средствами Postman.

Ход работы

Файл [swagger.ts](#)

```
import swaggerJSDoc from "swagger-jsdoc";

const options = {

  definition: {

    openapi: "3.0.0",

    info: {

      title: "Recipe API",

      version: "1.0.0",

      description: "Документация REST API для проекта с рецептами",

    },

    servers: [{ url: "http://localhost:3000" }],

  },

  apis: [".src/routes/*.ts", ".src/controllers/*.ts"], };

export const swaggerSpec = swaggerJSDoc(options);
```

Пример изменений в routes:

```
import { Router } from "express";
import { createComment, getComments, getCommentById, deleteComment } from
const router = Router();

/**
 * @openapi
 * /comments:
 *   post:
 *     summary: Создать комментарий
 *     tags:
 *       - Comments
 *     requestBody:
 *       required: true
```

```

*       content:
*         application/json:
*           schema:
*             type: object
*             properties:
*               text:
*                 type: string
*               userId:
*                 type: string
*               recipeId:
*                 type: string
*       responses:
*         201:
*           description: Комментарий создан
*/
router.post("/", createComment);

/**
 * @openapi
 * /comments:
 *   get:
 *     summary: Получить список всех комментариев
 *     tags:
 *       - Comments
 *     responses:
 *       200:
 *         description: Список комментариев
 */
router.get("/", getComments);

/**
 * @openapi
 * /comments/{id}:
 *   get:
 *     summary: Получить комментарий по ID
 *     tags:
 *       - Comments
 *     parameters:
 *       - name: id
 *         in: path
 *         required: true
 *         schema:
 *           type: string
 *     responses:
 *       200:
 *         description: Комментарий найден
 *       404:
 *         description: Комментарий не найден
 */
router.get("/:id", getCommentById);

/**
 * @openapi
 * /comments/{id}:
 *   delete:

```

```

*      summary: Удалить комментарий
*      tags:
*        - Comments
*      parameters:
*        - name: id
*          in: path
*          required: true
*          schema:
*            type: string
*      responses:
*        204:
*          description: Комментарий удалён
*/
router.delete("/:id", deleteComment);

export default router;

```

Swagger:

Recipe API 1.0.0 OAS 3.0

Документация REST API для проекта с рецептами

Servers

http://localhost:3000

Comments

POST /comments Создать комментарий

GET /comments Получить список всех комментариев

GET /comments/{id} Получить комментарий по ID

DELETE /comments/{id} Удалить комментарий

Ingredients

POST /ingredients Создать ингредиент

GET /ingredients Получить список всех ингредиентов

recipeSteps

POST	/recipe-steps	Создать шаг рецепта
GET	/recipe-steps/recipe/{recipeId}	Получить шаги рецепта по ID рецепта
PUT	/recipe-steps/{id}	Обновить шаг рецепта по ID
DELETE	/recipe-steps/{id}	Удалить шаг рецепта по ID

Subscriptions

POST	/subscriptions	Подписаться на пользователя
GET	/subscriptions	Получить все подписки
DELETE	/subscriptions	Отписаться от пользователя

Users

GET	/users	Получить список пользователей
POST	/users/login	Вход пользователя
POST	/users/register	Регистрация нового пользователя
PUT	/users/{id}	Обновить данные пользователя
DELETE	/users/{id}	Удалить пользователя

Postman:

The screenshot shows the Postman interface with a workspace named "Recipe API". The left sidebar lists collections, including "ingredients". The main panel shows a GET request to `{{baseUrl}}/ingredients`. The response is a 200 OK status with a JSON body containing an array of ingredients.

```
1 [
2   {
3     "id": "a1ea91b2-5236-42c7-b921-0a6407bbc6c0",
4     "name": "Мяка"
5   },
6   {
7     "id": "6405f4b9-307a-43c3-81f6-461255f1e8d0",
8     "name": "Сaxap"
9   },
10  {
11    "id": "a96a409f-6ca4-43d5-b28f-0d88fc6222bf",
12    "name": "Parmesan Cheese"
13  }
14 ]
```

Вывод

В данной работе были реализованы автодокументирование средствами swagger и документация API средствами Postman.