

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 3

Выполнил:

Захарчук Александр
К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

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

Ход работы

Для настройки документации был добавлен файл `swagger.ts`. Его содержание представлено в листинге 1. В данном файле также добавляются 2 эндпоинта – для страницы документации и для получения сырого файла спецификации.

Листинг 1 – файл `swagger.ts`

```
import * as swaggerJSDoc from "swagger-jsdoc";
import * as swaggerUi from "swagger-ui-express";
import { Express } from "express";

const options: swaggerJSDoc.Options = {
  definition: {
    openapi: "3.0.0",
    info: {
      title: "Recipe API",
      version: "1.0.0",
      description: "API для управления рецептами",
    },
    components: {
      securitySchemes: {
        bearerAuth: {
          type: "http",
          scheme: "bearer",
          bearerFormat: "JWT",
        },
      },
    },
    schemas: {
      User: {
        type: "object",
        properties: {
          username: { type: "string" },
          email: { type: "string" },
          profile_picture: { type: "string" },
          bio: { type: "string" },
        },
      },
      Recipe: {
        type: "object",
        properties: {
          recipe_id: { type: "integer" },
          title: { type: "string" },
          description: { type: "string" }
        },
      },
    },
  },
};
```



```
*/  
commentRouter.get("/", [checkJwt], async function (req: Request, res:  
Response) {  
    const comments = await commentRepository  
        .createQueryBuilder("comment")  
        .select("comment.comment_id")  
        .addSelect("comment.text")  
        .addSelect("comment.created_at")  
        .addSelect("comment.updated_at")  
        .leftJoinAndSelect("comment.recipe", "recipe")  
        .leftJoinAndSelect("comment.user", "user")  
        .getMany();  
    res.json(comments);  
})
```

После запуска сервера документация доступна по ссылке <http://localhost:3000/docs>. Документация представлена на рисунках 1 и 2.

Comments API для управления комментариями			^
POST	/comments	Создать комментарий	🔒 ▼
GET	/comments	Получить все комментарии	🔒 ▼
GET	/comments/{id}	Получить комментарий по ID	🔒 ▼
PUT	/comments/{id}	Обновить комментарий	🔒 ▼
DELETE	/comments/{id}	Удалить комментарий	🔒 ▼
Ingredients API для управления ингредиентами			^
POST	/ingredients	Создать ингредиент	🔒 ▼
GET	/ingredients	Получить список всех ингредиентов	🔒 ▼
GET	/ingredients/{id}	Получить ингредиент по ID	🔒 ▼
PUT	/ingredients/{id}	Обновить ингредиент	🔒 ▼
DELETE	/ingredients/{id}	Удалить ингредиент	🔒 ▼
Likes API для управления лайками рецептов			^
POST	/likes	Поставить лайк рецепту	🔒 ▼
GET	/likes	Получить лайки текущего пользователя	🔒 ▼
DELETE	/likes/{id}	Удалить лайк по ID	🔒 ▼
RecipeIngredients API для управления ингредиентами рецептов			^
POST	/recipes-ingredients	Добавить ингредиент в рецепт	🔒 ▼
GET	/recipes-ingredients	Получить все ингредиенты рецептов	🔒 ▼
GET	/recipes-ingredients/{id}	Получить ингредиент рецепта по ID	🔒 ▼
PUT	/recipes-ingredients/{id}	Обновить ингредиент рецепта по ID	🔒 ▼
DELETE	/recipes-ingredients/{id}	Удалить ингредиент из рецепта по ID	🔒 ▼
Recipes			^
POST	/recipes	Создать новый рецепт	🔒 ▼
GET	/recipes	Получить все рецепты	🔒 ▼
GET	/recipes/{id}	Получить рецепт по ID	🔒 ▼
PUT	/recipes/{id}	Обновить рецепт по ID	🔒 ▼
DELETE	/recipes/{id}	Удалить рецепт по ID	🔒 ▼
RecipeSteps			^
POST	/recipe-steps	Создать новый шаг для рецепта	🔒 ▼
GET	/recipe-steps/{id}	Получить шаг рецепта по ID	🔒 ▼
PUT	/recipe-steps/{id}	Обновить шаг рецепта по ID	🔒 ▼
DELETE	/recipe-steps/{id}	Удалить шаг рецепта по ID	🔒 ▼
SavedRecipes			^
POST	/saved-recipes	Сохранить рецепт	🔒 ▼
GET	/saved-recipes	Получить список сохраненных рецептов	🔒 ▼
DELETE	/saved-recipes/{id}	Удалить сохраненный рецепт по ID	🔒 ▼
Subscriptions			^
POST	/subscriptions	Подписаться на нового пользователя	🔒 ▼

Рисунок 1 - документация API



Рисунок 2- схемы API

Полученный файл спецификации был загружен в Postman для получения документации API. Полученный результат представлен на рисунке 3.

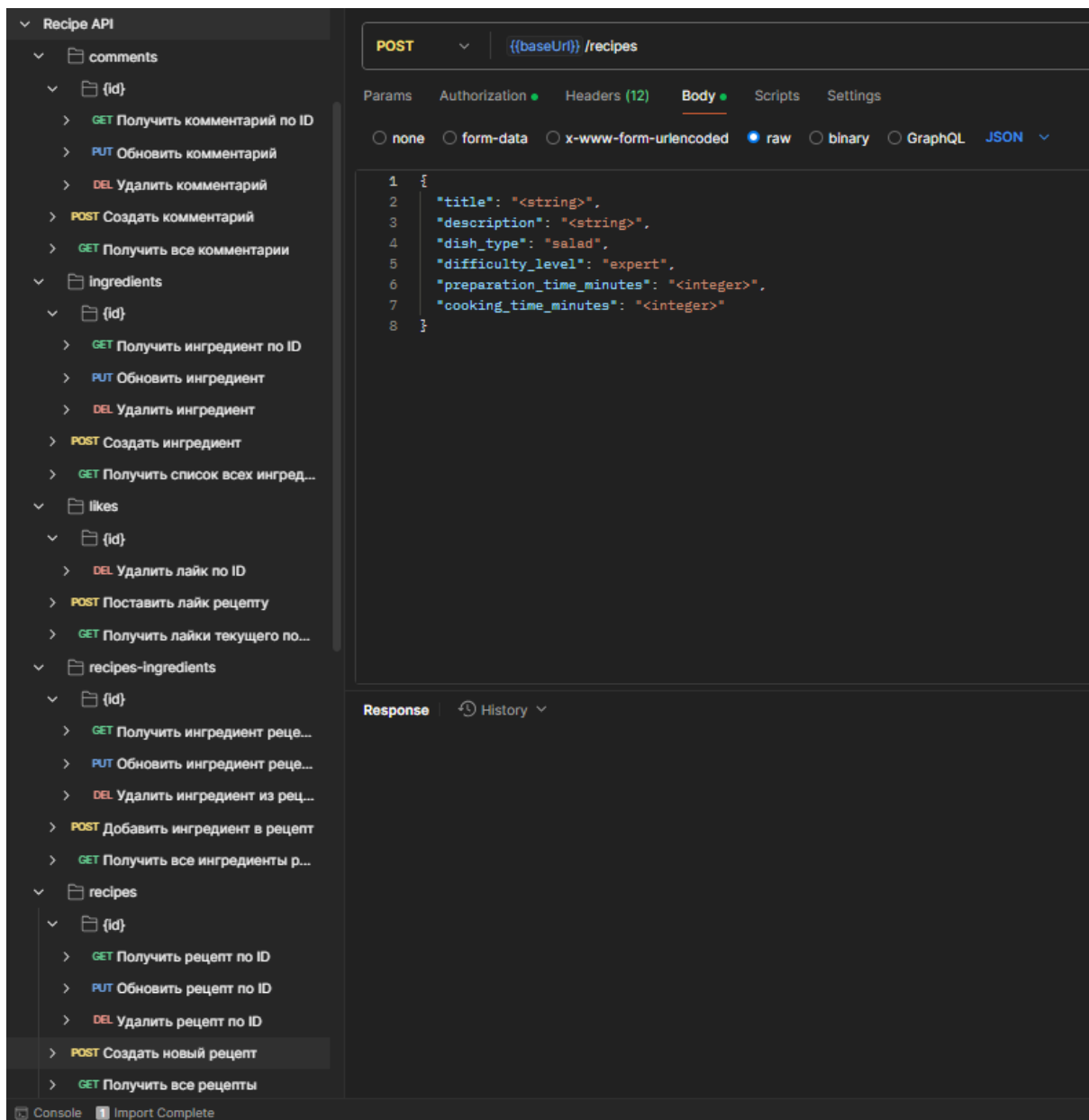


Рисунок 3 - коллекция в Postman

Вывод

В ходе выполнения данной работы были изучены и отработаны на практике автоматическое документирование API с помощью swagger-jsdoc, а также генерация коллекции в Postman. Данные подходы помогают существенно упростить тестирование API, и уменьшают ошибки в схемах запросов и ответов.