САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №3

Выполнил:

Ананьев Никита

К3340

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Документирование API средствами swagger и Postman

Ход работы

Для подключения сваггера к приложению на express была добавлена следующая функция (рис. 1). Она принимает объект app как аргумент, настраивает схему сваггера, добавляет соответствующий роут в приложение и возвращает обновленный app:

```
import { Express } from 'express';
import { getMetadataArgsStorage, RoutingControllersOptions} from 'routing-controllers';
import { routingControllersToSpec } from 'routing-controllers-openapi';
import * as swaggerUi from 'swagger-ui-express';
import { validationMetadatasToSchemas } from 'class-validator-jsonschema';

export function useSwagger(app: Express, options: RoutingControllersOptions): Express {
    try {
        const schemas = validationMetadatasToSchemas({
            refPointerPrefix: '#/definitions/',
        });

        const storage = getMetadataArgsStorage();
        const spec = routingControllersToSpec(storage, options, {
            components: {
                schemas,
                securitySchemes: {
                    bearerAuth: {
                        type: 'http',
                        scheme: 'bearer',
                        bearerFormat: 'JWT',
                    },
                },
            },
            info: {
                title: 'Boilerplate API documentation',
                description: 'API documentation for boilerplate',
                version: '1.0.0',
            },
            security: [{ bearerAuth: [] }]
        });
        app.use('/docs', swaggerUi.serve, swaggerUi.setup(spec));
        return app;
    } catch (error) {
        console.error('Swagger set up error:', error);
        return app;
    }
```

Рисунок 1 - функция для настройки swagger

Теперь, необходимо обновить контроллеры, чтобы в документации swagger описание роута, примеров входных/выходных данных и статус-кодов было корректным (рис. 2):

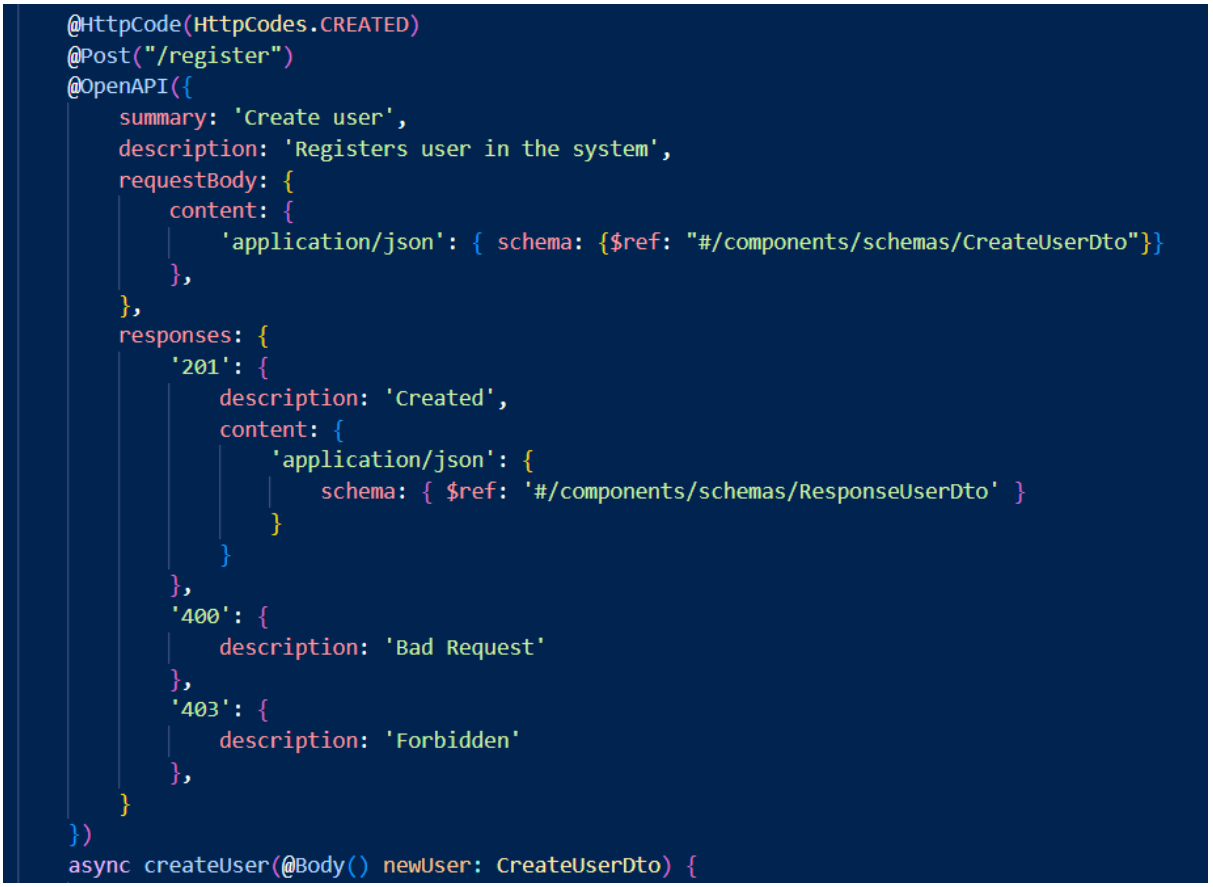

Рисунок 2 - пример описания роута, с использованием декоратора из пакета OpenAPI

Такие декораторы были добавлены ко всем роутам приложения. На рис. 3 представлен внешний вид получившейся документации Swagger:

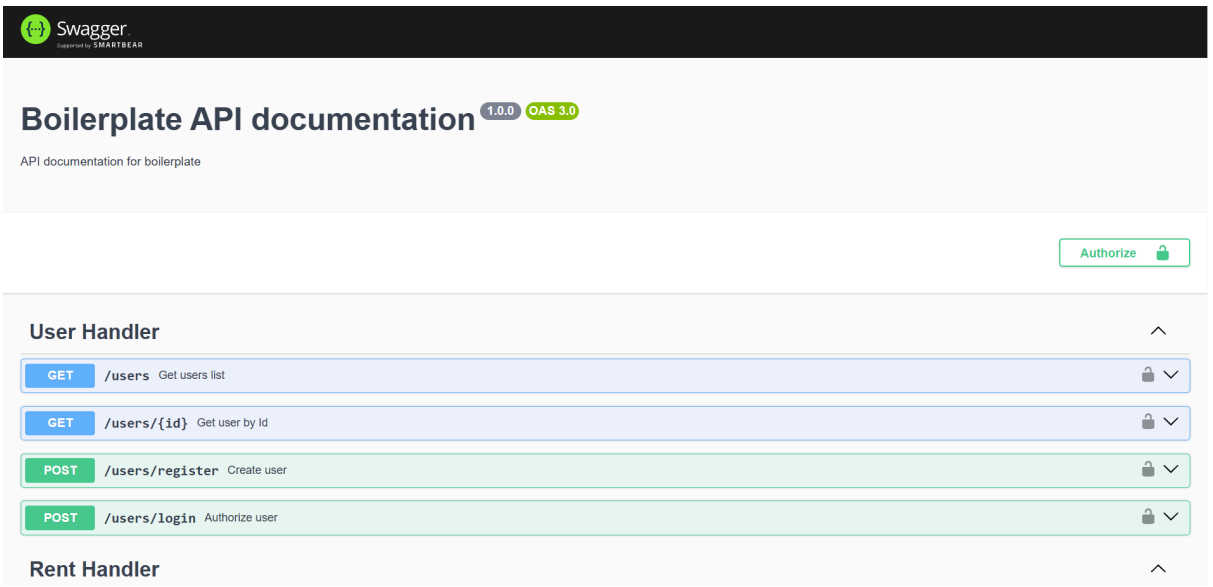

Рисунок 3 - документация API на основе swagger UI

Вывод

Express (+ routing-controllers) позволяет использовать обширный и удобный набор для документирования разрабатываемого веб-приложения. В том числе с помощью OpenAPI и Swagger.