

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа № 4
“Документирование API из ЛР2 средствами Swagger,
Postman”

Выполнил:

Коротин А.М.

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Задokumentировать реализованный ранее REST API в формате OpenAPI 3.0.0 и выше.

Ход работы

Для автоматической генерации документации в формате OpenAPI на платформе Node.js были выбраны библиотеки swagger-jsdoc и swagger-ui-express. Их использование позволяет автоматически генерировать документацию на основе комментариев в файлах проекта в формате JSDoc.

Первоначальная настройка документации приведена на рисунке 1.

```

const swaggerOptions: swaggerJsDoc.Options = {
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Account API',
      description: "User account API for Smart Device App",
      version: '1.0.0-BETA',
      contact: {
        name: "Alexey Korotin K33392"
      }
    },
    externalDocs: {
      description: "swagger.json",
      url: '/swagger.json'
    },
    servers: [
      {
        url: `http://${appHost}:${appPort}/v1`
      }
    ]
  },
  apis: ['**/presentation/http/**/*.ts']
};

const docs :Object = swaggerJsDoc(swaggerOptions);

app.use('/swagger-ui', swaggerUi.serve, swaggerUi.setup(docs));

```

Рисунок 1 — Первоначальная настройка документации

В свойстве `apis` объекта `swaggerOptions` указан путь, из которого будут извлечены комментарии в формате JSDoc для дальнейшего преобразования в формат OpenAPI 3.0.0. Пример соответствующего комментария приведен на рисунке 2.

```

75  /**
76  * @swagger
77  * /auth/login:
78  *   post:
79  *     tags: [Auth]
80  *     summary: Log in into the existing account
81  *     requestBody:
82  *       description: Authentication credentials
83  *       required: true
84  *       content:
85  *         application/json:
86  *           schema:
87  *             $ref: '#components/schemas/Credentials'
88  *     responses:
89  *       200:
90  *         description: Ok
91  *         content:
92  *           application/json:
93  *             schema:
94  *               $ref: '#components/schemas/JWT'
95  *       400:
96  *         description: Bad request
97  *       401:
98  *         description: Invalid credentials
99  */
100 login = async (req: Request, res: Response) : Promise<Response<...>> => {

```

Рисунок 2 — Комментарий JSDoc для эндпоинта логина

Пример сгенерированной документации приведен на рисунке 3.

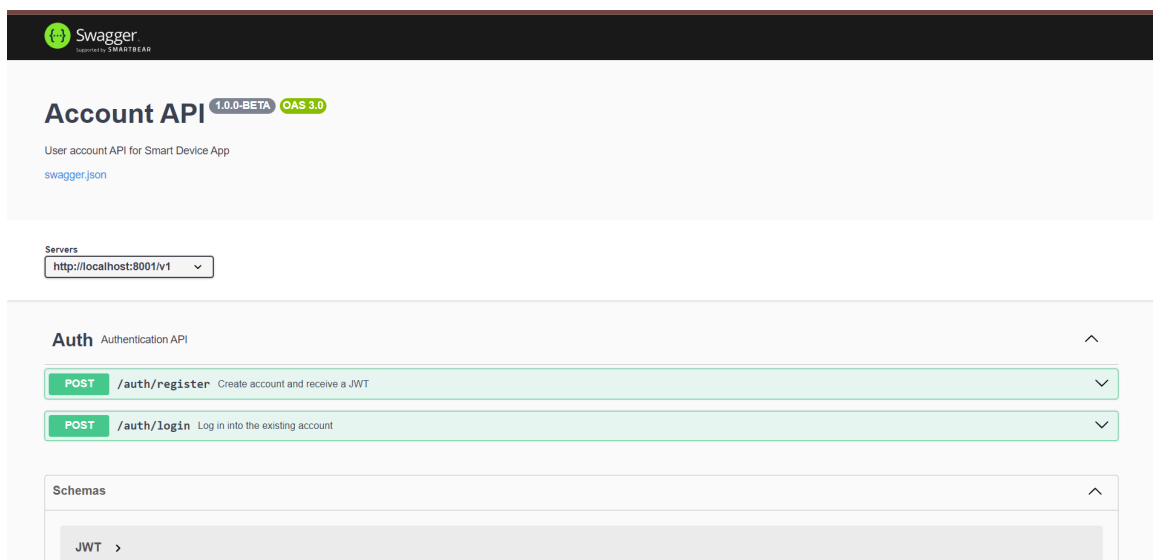


Рисунок 3 — Сгенерированная документация Swagger UI

Вывод

В ходе выполнения практической работы были изучены способы документирования REST API на платформе Node.js с использованием фреймворка Express.

Также был задокументирован ранее реализованный REST API для управления девайсами умного дома.