ДИСЦИПЛИНА	Интеграция информационных систем с использованием API и микросервисов
ПОДРАЗДЕЛЕНИЕ	ПИШ СВЧ-электроники
ВИД УЧЕБНОГО МАТЕРИАЛА	Методические указания к практическим занятиям
ПРЕПОДАВАТЕЛЬ	Астафьев Рустам Уралович
СЕМЕСТР	1 семестр, 2024/2025 уч. год

Практическое занятие №6. REST API 2
Введение в REST API 2
Спецификация API 2
Создание простого RESTful API с использованием Node.js и Express. ... 2
Документирование API 4
Базовое задание №3 4

Практическое занятие №6. REST API

Введение в REST API

Предварительно обсудим основную концепцию API и его роль в веб-разработке. Так же вспомни основные принципы REST:

- Клиент-серверная архитектура
- Безсостояние (stateless)
- Кеширование
- Единообразие интерфейса
- Слои системы
- HTTP методы: GET, POST, PUT, DELETE

Спецификация API

Введение в спецификацию API (OpenAPI/Swagger). Структура спецификации:

- Заголовок (info)
- Путь (paths)
- Методы (GET, POST и т.д.)
- Параметры и ответы

Пример спецификации для простого API (например, для управления задачами). Располагается по ссылке в гитхаб.

Далее перейдём к примеру, который ляжет в основу базового задания.

Создание простого RESTful API с использованием Node.js и Express.

1. Создадим
2. Инициализируйте новый проект Node.js:

   npm init -y
3. Установите необходимые зависимости:

   npm install express body-parser

4. Создайте файл server.js в корне проекта.

5. Откройте server.js и добавьте аналогичный код из примера.

6. Запустите сервер:

node server.js

7. Вы должны увидеть сообщение в терминале:

Server is running on http://localhost:3000

8. Теперь мы можем протестировать наше API с помощью Postman или curl. Вот некоторые примеры запросов.

- Получить список задач
  • Метод: GET
  • URL: http://localhost:3000/tasks
- Создать новую задачу
  • Метод: POST
  • URL: http://localhost:3000/tasks
  • Тело запроса (JSON):

```
{
  "title": "Learn REST API",
  "completed": false
}
```

- Получить задачу по ID
  • Метод: GET
  • URL: http://localhost:3000/tasks/1
- Обновить задачу по ID
  • Метод: PUT
  • URL: http://localhost:3000/tasks/1
- Тело запроса (JSON):

```
{
  "title": "Learn REST API Basics",
  "completed": true
```

}

- Удалить задачу по ID

  • Метод: DELETE

  • URL: http://localhost:3000/tasks/1

Документирование API

Для документирования API вы можете использовать Swagger. Для этого вам нужно будет установить дополнительные зависимости и настроить Swagger UI. Вот основные шаги:

1. Установите зависимости для Swagger:

   npm install swagger-ui-express swagger-jsdoc

2. Рассмотрим конфигурацию для файла со Swagger.

3. Теперь вы можете запустить сервер и открыть http://localhost:3000/api-docs, чтобы увидеть документацию вашего API.

Базовое задание №3

В рамках выполнения задания необходимо подготовить http-сервер, который будет функционировать на одном из портов и возвращать содержимое файла index.html. В состав содержимого необходимо включить каталог товаров в виде карточек, содержащих следующие сведения:

- название
- стоимость
- описание.

Эти сведения подтягиваются из json-файла на сервере.

Также необходимо реализовать http-сервер и подготовить для него API-спецификацию. Это приложение будет выступать в роли панели администратора для приложения интернет-магазина. В рамках панели администратора возможно выполнять следующие действия:

- добавление товара или нескольких товаров сразу;

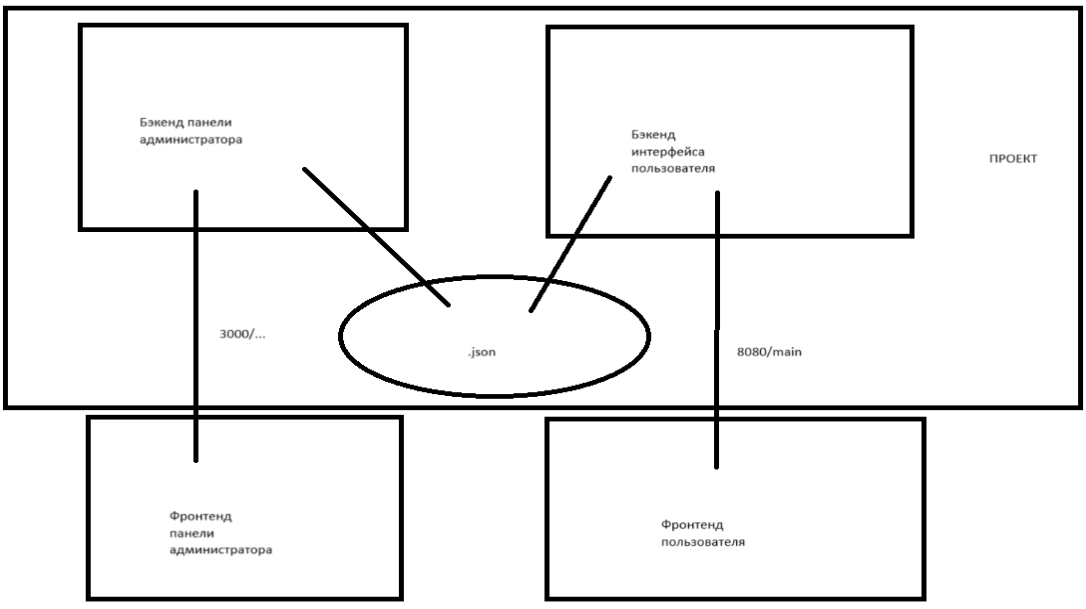
- редактирование информации о товаре по ID;
- удаление товара по ID.

Таким образом, папка проекта будет содержать два бэкенд-приложения, активирующих два локальных порта на устройстве (например, 3000 и 8080), а также фронтенд (может быть просто html страница с кнопками без менеджеров состояния и прочего, то есть обновление происходит с обновлением страницы).

В качестве примера необходимо разработать минимум 5 товаров, описание которых попадает в карточки товаров.

Также товары должны быть разбиты по категориям (минимум две категории, из 5 товаров один находится сразу в двух) для тестирования приложения.

В качестве ответа необходимо представить ссылку на публичный гит-репозиторий, клонирование с которого и выполнение инструкций по установке приведет к запуску приложения локально.



Список источников и ссылки

1. https://github.com/astafiev-rustam/integration-of-information-systems/tree/Practice-1-6