### ****Структура документа****

#### ****1. Описание задачи****

* **Цель** : Создать серверную часть REST API для системы управления товарами.
* **Тематика** : Тематика сайта определяется из прошлого задания.
* **Основная задача** : Реализовать базовые операции CRUD (Create, Read, Update, Delete) для товаров.
* Товары должны быть доступны через REST API с соответствующими маршрутами для выполнения CRUD-операций.

#### ****2. Требуемые методы****

Необходимо реализовать следующие маршруты для работы с товарами:

|  |  |  |
| --- | --- | --- |
| GET | /products | Получить список всех товаров. |
| GET | /products/:id | Получить информацию о конкретном товаре по его ID. |
| POST | /products | Создать новый товар. |
| PUT | /products/:id | Обновить информацию о товаре по его ID. |
| DELETE | /products/:id | Удалить товар по его ID. |

#### ****3. Подробное описание каждого метода****

##### ****3.1. GET /products****

* Возвращает список всех товаров в формате JSON.

##### ****3.2. GET /products/:id****

* Возвращает информацию о конкретном товаре по его ID.
* Если товар не найден, возвращается ошибка 404.

##### ****3.3. POST /products****

* Создает новый товар на основе переданных данных в теле запроса.
* В ответе возвращается созданный товар с уникальным ID.

##### ****3.4. PUT /products/:id****

* Обновляет информацию о товаре по его ID.
* Если товар не найден, возвращается ошибка 404.

##### ****3.5. DELETE /products/:id****

* Удаляет товар по его ID.
* В ответе возвращается сообщение об успешном удалении или ошибка 404, если товар не найден.

#### ****4. Технические требования****

* Использовать фреймворк Express.js для создания сервера.
* Хранить данные товаров можно в памяти (например, в массиве объектов) или использовать легковесную базу данных, такую как SQLite или MongoDB.
* Сервер должен запускаться на порту 3000 по умолчанию.

#### ****5. Инструкции для тестирования****

* Использовать инструменты, такие как Postman или curl, для тестирования API.
* Проверить работу каждого метода, отправляя соответствующие запросы.

#### ****Примеры тестовых запросов**** :

* Получить все товары: GET /products
* Создать новый товар: POST /products
* Обновить товар: PUT /products/:id
* Удалить товар: DELETE /products/:id

#### ****6. Дополнительные задания (по желанию)****

* Реализовать возможность фильтрации товаров по категории или цене.
* Добавить сортировку товаров по цене или названию.
* Настроить пагинацию для вывода большого списка товаров.

#### ****7. Критерии оценки****

* Все основные CRUD-маршруты работают корректно.
* Сервер возвращает правильные ответы в соответствии с запросами.
* Код структурирован и читаем.

#### ****8. Заключение****

Данное задание поможет освоить основы работы с Node.js и Express, а также научиться создавать REST API.