**Задание:**

Необходимо создать простое API для управления списком готовой продукции и связей между ними. API должно поддерживать операции CRUD (создание, чтение, обновление и удаление) для продукции и связей.

Пример

**Бэкенд:**

1. Создайте базу данных с следующими таблицами:
   * "Nomenklature" (Поля: Название, Цена)
   * "Links" (Поля: “Ссылка на текущую продукцию”, “ссылка на вышестоящую продукцию”, “количество текущих изделий, входящих в вышестоящее”)

Пример БД

Nomenklature

| Id | name | price |
| --- | --- | --- |
| 1 | Продукт 1 | 100 |
| 2 | Продукт 2 | 200 |
| 3 | Продукт 3 | 300 |
| 4 | Продукт 5 | 400 |

Links

| Id | nomenklatureId | parentId | Kol |
| --- | --- | --- | --- |
| 4 | 1 |  | 1 |
| 1 | 2 | 1 | 2 |
| 2 | 3 | 2 | 1 |
| 3 | 4 | 2 | 3 |

1. Реализуйте следующие API-маршруты (endpoints) для взаимодействия с базой данных:
   * GET /api/nomenklatures - Получить список всей продукции (плоский список)
   * GET /api/nomenklatures/{id} - Получить информацию о продукции с определенным Id, включая все подчиненные продукты и подсчитать стоимость продукции и количество на каждом уровне. Количество продукции на первом уровне принимаем равным 1. Стоимость продукции - количество по продукту, умноженное на цену по продукту плюс стоимость всех нижестоящих уровней, пример

| Продукт | Количество | Цена | Стоимость |
| --- | --- | --- | --- |
| Продукт 1 | 1 | 100 | 3400 + 100 = 3500 |
| Продукт 2 | 2 | 200 | (200 + 1500) \* 2= 3400 |
| Продукт 3 | 1 | 300 | 300 \* 1 = 300 |
| Продукт 4 | 3 | 400 | 400 \* 3 = 1200 |

* + POST /api/nomenklatures - Создать новый продукт
  + PUT /api/nomenklatures/{id} - Обновить продукт с определенным Id
  + DELETE /api/nomenklatures/{id} - Удалить продукт с определенным Id
  + POST /api/nomenklatures/link – Связать 2 продукта
  + DELETE /api/nomenklatures/link – Удалить связь между 2 продуктами

1. Используйте Entity Framework для работы с базой данных.

**Дополнительное задание (Тесты):**

1. Напишите тесты проверяющие различные сценарии использования API и граничные случаи.

**Дополнительное задание (Docker):**

1. Создайте Docker-контейнер для вашего приложения. Инструкции по созданию Docker-контейнера должны быть описаны в отдельном файле (например, Dockerfile).

**Инструкции:**

* Результат выполнения задания должен быть предоставлен в виде исходного кода проекта.
* Вы можете использовать любые библиотеки, фреймворки или инструменты, с которыми вам удобно работать.
* Документация к API не требуется, но приветствуется комментарии в коде для объяснения назначения и использования различных частей приложения.

Удачи в выполнении задания! Если у вас возникнут дополнительные вопросы, пожалуйста, не стесняйтесь задавать их.