

Павел_МельникЗаказы v1.0

Чеклист:

- Убедитесь что требования соответствуют архитектуре: микросервис обращается только к **своей** БД
- Убедитесь, что в требования к бэку не просочились требования к интерфейсу. Помним, **пользователь не имеет доступа к бэку**.
- Работа с заказами
 - Получить список заказов
 - Получить заказ
 - Создать заказ
 - Обновить заказ
- Работа с товарами в заказе
 - Добавить товар
 - Изменить количество товара
 - Удалить товар из заказа
- Приложение 1. Мэппинг данных объектов JSON на модель БД
 - Мэппинг Order
 - Мэппинг Product

Работа с заказами ↗

Получить список заказов ↗

Метод	GET orders/
Описание	Возвращает список всех заказов с поддержкой фильтрации по различным параметрам.
Входные данные	Параметры фильтрации: <ul style="list-style-type: none">• Фильтрация по статусу заказа (например, "New", "Completed")• Фильтрация по ID клиента.• Фильтрация по дате создания заказа (например, все заказы за последний месяц).• Фильтрация по диапазону общей стоимости заказа.
Выходные данные	Массив объектов Order , отфильтрованный по заданным критериям.
Коды ответов	<ul style="list-style-type: none">• Успешный ответ содержит код ответа <code>200 OK</code>• Ошибки:<ul style="list-style-type: none">◦ <code>400 Bad Request</code> — Некорректные параметры фильтрации.◦ <code>500 Internal Server Error</code> — ошибка сервера

Требования по реализации:

- Метод должен возвращать массив всех заказов, соответствующих заданным фильтрам, из базы данных.
- Если фильтры не заданы, возвращаются все заказы.
- В случае ошибки сервер должен вернуть соответствующий код ошибки и описание ошибки.

Получить заказ ↗

Метод	GET orders/{order_id}
Описание	Возвращает данные конкретного заказа по его ID.
Входные данные	Нет
Выходные данные	Объект Order
Коды ответов	<ul style="list-style-type: none">• Успешный ответ содержит код ответа <code>200 OK</code>• Ошибки:<ul style="list-style-type: none">◦ <code>404 Not Found</code> – заказ не найден◦ <code>500 Internal Server Error</code> – ошибка сервера


Требования по реализации:

- Метод должен возвращать данные конкретного заказа по его ID из базы данных.
- В случае ошибки сервер должен вернуть соответствующий код ошибки и описание ошибки.

Создать заказ ↗

Метод	POST orders/
Входные данные	Объект <code>Order</code>
Выходные данные	Объект <code>Order</code>
Коды ответов	<p>Успешный ответ содержит код ответа <code>200 OK</code></p> <p>Ошибки:</p> <ul style="list-style-type: none">• <code>400 Bad Request</code> – ошибки в полях объекта Order

Требования по реализации

 Обратите внимание, что в Вашей БД название таблицы с товарами в заказах может отличаться от ProductInOrder. В этом случае исправьте в требованиях к реализации название таблицы на Ваше.

Должна создаваться новая запись в таблице `order` (см. [Мэппинг Order](#)), при этом:

- дата создания заказа должна заполняться автоматически текущей датой;
- для товаров в заказе должны создаваться записи в таблице ProductInOrder (см. [Мэппинг Product](#)).

Метод должен возвращать в объекте `Order` данные, сохраненные в БД, в том числе `order_id`.

Обновить заказ ↗

Метод	PATCH orders/{order_id}
Входные данные	Объект Order
Выходные данные	Объект Order
Коды ответов	<ul style="list-style-type: none">Успешный ответ содержит код ответа 200 OKОшибки:<ul style="list-style-type: none">400 Bad Request – ошибки в полях объекта Order404 Not Found – редактируемый заказ не найден

Требования по реализации:


- Метод должен обновлять запись в таблице Order и возвращать обновленные данные.

Работа с товарами в заказе ↗

Добавить товар ↗

Метод	POST orders/{order_id}/products
Входные данные	Объект Product
Выходные данные	-
Коды ответов	<p>Успешный ответ содержит код ответа 200 OK</p> <p>Ошибки:</p> <ul style="list-style-type: none">404 Not Found – редактируемый заказ не найден400 Bad Request – ошибки в полях объекта Product409 Conflict - на складе недостаточно остатков, чтобы выполнить запрос

Требования по реализации

 Обратите внимание, что в Вашей БД название таблицы с товарами в заказах может отличаться от ProductInOrder. В этом случае исправьте в требованиях к реализации название таблицы на Ваше. А также название поля, для хранения кол-ва товара в заказе.

- Уменьшить количество остатков для товара используя метод PATCH products/{product_id} (см. [Товары v2.0 | Обновить данные об остатках товара](#)), где Increase = - Product.Amount .
- Если метод вернул результат 200 OK , сохранить информацию о добавленном товаре в таблице ProductInOrder (см. [Мэппинг Product](#)).
Иначе, вернуть ошибку 409 Conflict .

Изменить количество товара ↗

Метод	PATCH orders/{order_id}/products/{product_id}
Входные данные	Increase - содержит целое число, положительное или отрицательное. Если Increase > 0, необходимо увеличить количество товаров в заказе на указанное число единиц. Если Increase < 0 - уменьшить.

Выходные данные	-
Коды ответов	Успешный ответ содержит код ответа <code>200 OK</code> Ошибки: <ul style="list-style-type: none"><code>404 Not Found</code> – редактируемый заказ или товар не найден<code>409 Conflict</code> - на складе недостаточно остатков, чтобы выполнить запрос

Требования по реализации

⚠ Обратите внимание, что в Вашей БД название таблицы с товарами в заказах может отличаться от `ProductInOrder`. В этом случае исправьте в требованиях к реализации название таблицы на Ваше. А также название поля, для хранения кол-ва товара в заказе.

- Изменить на `Increase` данные об остатках товара используя метод `PATCH products/{product_id}` (см. [Товары v2.0 | Обновить данные об остатках товара](#)).
- Если метод вернул результат `200 OK`, значение поля `ProductInOrder.Amount` изменить на `Increase`, где `ProductInOrder.Product_ID = product_id` и `ProductInOrder.Order_ID = order_id`.
Иначе, вернуть ошибку `409 Conflict`.

```
{
  "Increase": 1
}
```

Удалить товар из заказа ↗

Метод	DELETE orders/{order_id}/products/{product_id}
Входные данные	-
Выходные данные	-
Коды ответов	Успешный ответ содержит код ответа <code>200 OK</code> Ошибки: <ul style="list-style-type: none"><code>404 Not Found</code> – редактируемый заказ или товар не найден

Требования по реализации:

- Увеличить количество остатков для товара, используя метод `PATCH products/{product_id}`, где `Increase = ProductInOrder.Amount`.
- Удалить запись о товаре из таблицы `ProductInOrder`.

Описание параметров фильтра для списка заказов ↗

Параметр фильтра	Тип данных	Описание	Пример значения
<code>filter.id</code>	Integer	Уникальный идентификатор заказа, по которому производится фильтрация.	<code>1001</code>
<code>filter.status</code>	String	Статус заказа (например, "Новый", "В обработке", "Доставлен"). Этот параметр фильтрует заказы по	"Новый"

		их текущему статусу.	
<code>filter.date</code>	Date	Дата создания заказа. Этот параметр фильтрует заказы по указанному временному диапазону.	<code>2024-07-01</code>
<code>filter.is_deleted</code>	Boolean	Логическое значение, определяющее, находится ли заказ в архиве (удален) или актуален. Если <code>true</code> , заказ находится в архиве.	<code>false</code> , <code>true</code>
<code>filter.customer_name</code>	String	Имя клиента. Этот параметр фильтрует заказы по имени клиента.	<code>"Иван"</code>
<code>filter.total_amount_min</code>	Decimal	Минимальная сумма заказа, начиная с которой производится фильтрация.	<code>1000.00</code>
<code>filter.total_amount_max</code>	Decimal	Максимальная сумма заказа, до которой производится фильтрация.	<code>5000.00</code>

Приложение 1. Мэппинг данных объектов JSON на модель БД

Мэппинг Order

Поле	Тип данных	Описание
Order_ID	INT (PK)	Идентификатор заказа
FirstName	VARCHAR(255)	Имя
LastName	VARCHAR(255)	Фамилия
Email	VARCHAR(255)	Электронная почта
Phone	VARCHAR(20)	Телефон
Address	VARCHAR(255)	Адрес
OrderDate	DATETIME	Дата создания заказа
TotalAmount	DECIMAL(10, 2)	Общая сумма заказа
Status	VARCHAR(50)	Статус заказа

Мэппинг Product

Поле	Тип данных	Описание
Product_ID	INT (PK)	Идентификатор товара
ProductType_ID	INT	Идентификатор типа товара
MaterialType_ID	INT	Идентификатор материала
VendorCode	VARCHAR(50)	Артикул
Name	VARCHAR(255)	Наименование товара
Equipment	VARCHAR(255)	Комплектация

