# API Документация

## 1. Предисловие

Я решил обойтись одним общим API и, пересмотрев концепцию, пришёл к заключению, что:

- Сделав приложение "портативным", оно будет чрезмерно сложным для установки клиентами;

- Вариант сервиса по подписке выглядит более привлекательно с финансовой точки зрения;

- Разворачивать сервис правильнее на стороне разработчика.

## 2. Сущности

### 2.1. Order (Заказ)

Поля:

- id: Long - уникальный идентификатор;

- storeName - название магазина;

- products - продукты в заказе;

- status: Status - статус заказа.

### 2.2. Product (Товар)

Поля:

- id: Long - уникальный идентификатор;

- name: String - название товара;

- quantity - количество;

### 2.3. Status (Статус)

Возможные значения:

- NEW ("Новый")

- IN\_PROGRESS ("В обработке")

- COMPLETED ("Завершен")

- CANCELLED ("Отменен")

### 2.4. Company (Компания)

Пока не реализовано.

## 3. API Endpoints

### 3.1. Order Controller (/api/order)

#### POST /api/order

- Создает новый заказ

- Тело запроса: JSON объекта CustomerOrder

- Возвращает: созданный заказ (200 OK)

#### GET /api/order

- Получает список всех заказов

- Возвращает: List<CustomerOrder> (200 OK)

#### GET /api/order/{id}

- Получает заказ по ID

- Возвращает: CustomerOrder (200 OK) или 404 Not Found

#### PUT /api/order/{id}

- Обновляет заказ по ID

- Тело запроса: JSON объекта CustomerOrder

- Возвращает: обновленный заказ (200 OK) или 404 Not Found

#### DELETE /api/order/{id}

- Удаляет заказ по ID

- Возвращает: 204 No Content или 404 Not Found

#### GET /api/order/{id}/getStatus

- Получает статус заказа по ID

- Возвращает: объект Status (200 OK) или 404 Not Found

#### PATCH /api/order/{id}/setStatus

- Устанавливает статус заказа

- Параметр запроса: status (значение из Enum Status)

- Возвращает: обновленный заказ (200 OK) или 404 Not Found

### 3.2. Product Controller (/api/product)

#### POST /api/product

- Создает новый товар

- Тело запроса: JSON объекта Product

- Возвращает: созданный товар (200 OK)

#### GET /api/product

- Получает список всех товаров

- Возвращает: List<Product> (200 OK)

#### GET /api/product/{id}

- Получает товар по ID

- Возвращает: Product (200 OK) или 404 Not Found

#### PUT /api/product/{id}

- Обновляет товар по ID

- Тело запроса: JSON объекта Product

- Возвращает: обновленный товар (200 OK) или 404 Not Found

#### DELETE /api/product/{id}

- Удаляет товар по ID

- Возвращает: 204 No Content или 404 Not Found

### 3.3. QR Code Controller (/api/qr)

#### POST /api/qr/generate

- Генерирует QR-код для заказа

- Тело запроса: JSON объекта CustomerOrder

- Возвращает: PNG изображение QR-кода (200 OK) или 500 Internal Server Error

- QR-код содержит ссылку вида: https://example.com/orders?orderId={orderId}

### 3.4. Status Controller (/api/status)

#### GET /api/status/getStatuses

- Получает список всех возможных статусов

- Возвращает: список объектов с полями name и displayName (200 OK)

Пример:

[

{"name": "NEW", "displayName": "Новый"},

{"name": "IN\_PROGRESS", "displayName": "В обработке"},

...

]

## 4. Примеры запросов

Добавлю чуть позже.