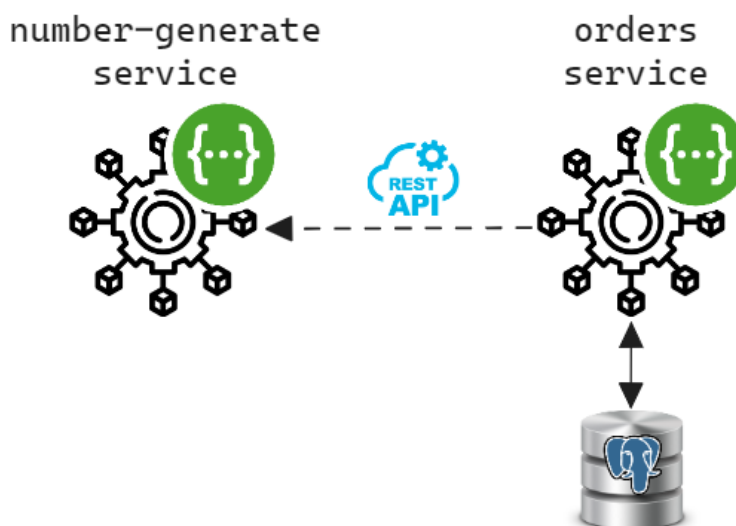


## Тестовое задание

Необходимо разработать 2 сервиса на Spring Boot. Первое приложение должно реализовывать функционал управления заказами и предоставлять REST API для клиента. Второе приложение должно реализовывать функционал для генерации и выдачи номера заказа и предоставлять REST API для сервиса orders.

Схема взаимодействия сервисов выглядит следующим образом:



### Требования к сервису по управлению заказами (orders-service)

Должно быть реализовано следующее API (/orders):

1. Создание заказа
2. Получение заказа по его идентификатору
3. Получение заказа за заданную дату и больше заданной общей суммы заказа
4. Получение списка заказов, не содержащих заданный товар и поступивших в заданный временной период.

### Требования к сервису генерации номер заказа (number-generate-service)

Реализовать метод (/numbers), который будет возвращать сгенерированный номер заказа (5 символов) и текущую дату и время (формат YYYYMMDD), пример 1111120241212. Алгоритм генерации номера продумывает сам кандидат, основное требование к номеру заказа, чтобы не было дублирования. Для хранения можно использовать структуры данных.

## Результаты тестового задания

### Начальный уровень:

1. Конфигурация производится стандартным образом через YAML-файл
2. Данные о заказах хранятся в реляционной базе данных - например, PostgreSQL, H2. Для работы с ней используется Spring JDBC
3. Версия Spring Boot – самая свежая стабильная
4. Приложение покрыто Unit-тестами
5. Написать Swagger-контракт для приложения и поместить его в ресурсы приложения
6. Приложение собирается с помощью Maven 3 (Gradle).

## Продвинутый уровень:

1. Приложение должно быть автоматически собрано в Docker-образ - можно применить соответствующий плагин для Maven, написан docker-compose
2. Использовать NoSQL базу данных для хранения сгенерированных номеров – например, Redis или MongoDB
3. Реализовать пользовательский интерфейс для удобного отображения REST API приложения в браузере. Выбор способа реализации свободный, применять ли Bootstrap, JS-фреймворки, шаблонизаторы типа Thymeleaf/Mustache – всё индивидуально

### Приложение 1. Структура базы данных

Описание	Тип
----------	-----

Заказы (Таблица 1)	
ID (PK)	Числовое
Номер заказа	Строка
Общая сумма заказа	Числовое
Дата заказа	Дата
Получатель	Строка
Адрес доставки	Строка
Тип оплаты (карта, наличные)	Строка
Тип доставки (самовывоз, доставка до двери)	Строка

Детали заказа (Таблица 2)	
ID (PK)	Числовое
Артикул товара	Числовое
Название товара	Строка
Количество	Числовое
Стоимость единицы товара	Числовое
ID заказа (ссылка на Таблицу 1)	Числовой