Осуществлена реализация 3-х микросервисов, а также реализация сервиса notification (rabbitmq).

Изображение выглядит как графическая вставка, Графика, символ, дизайн

Автоматически созданное описаниеДля каждого миркросервиса подключается своя база данных:

Изображение выглядит как Шрифт, логотип, Графика, дизайн

Автоматически созданное описание- postgresql для productService, stooreroomService;

- mongodb для orderService.



Также есть реализация брокера сообщений rabbitmq.

Изображение выглядит как Шрифт, логотип, Графика, текст

Автоматически созданное описаниеЕсть связь по gRPC между orderService и storeroomService.

Каждый микросервис, а также база данных, брокер сообщений, gRPC работают на своих портах:

- productService 18080, postgresql 5432;

- orderService 18081, mongodb 27017, rabbitmq 5672, gRPC 50054;

- storeroomService 18082, postgresql 5433.



Для проверки работоспособности был выбран postman.

Изображение выглядит как Шрифт, логотип, Цвет электрик, символ

Автоматически созданное описаниеДля быстрого поднятия базы данных и брокера сообщений использовался docker.

Поднятие postgres для productService, storeroomService:





Поднятие mongodb для orderService:



Поднятие rabbitmq для orderService: 

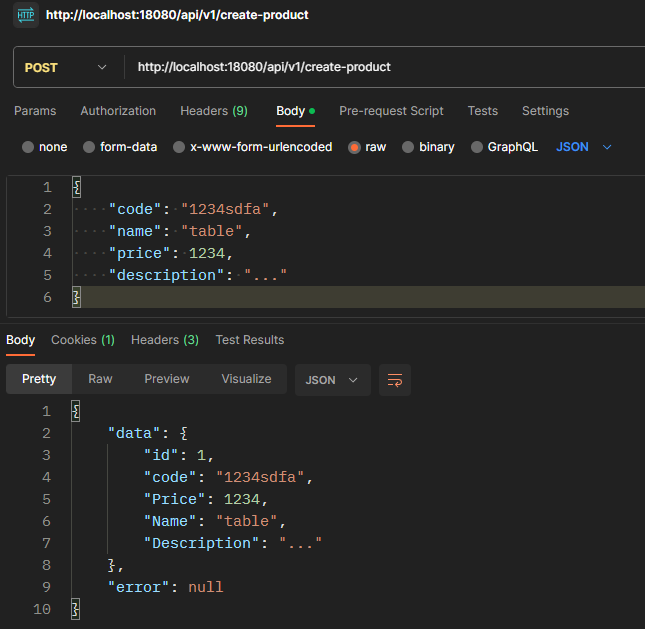
Как это выглядит в docker desktop:

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

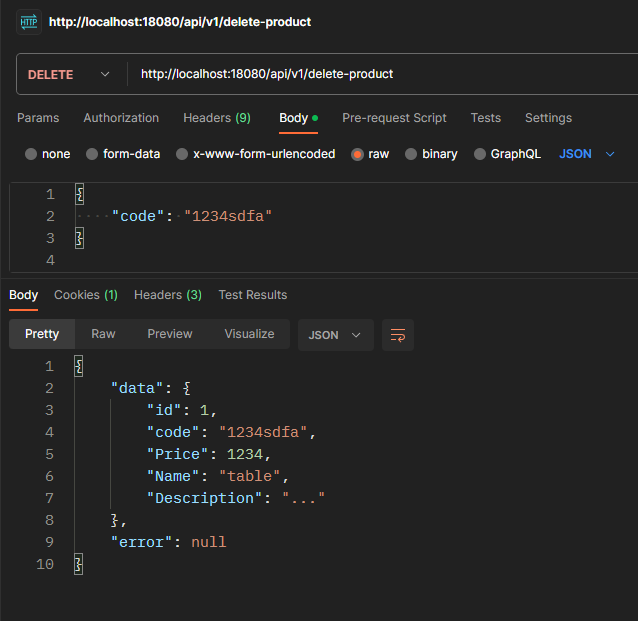
Автоматически созданное описание

Пример работы productService:

Создание продукта:

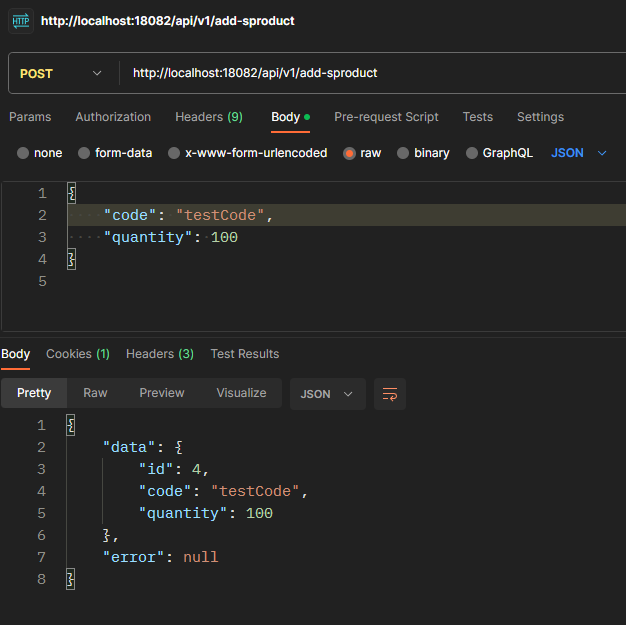


удаление продукта:

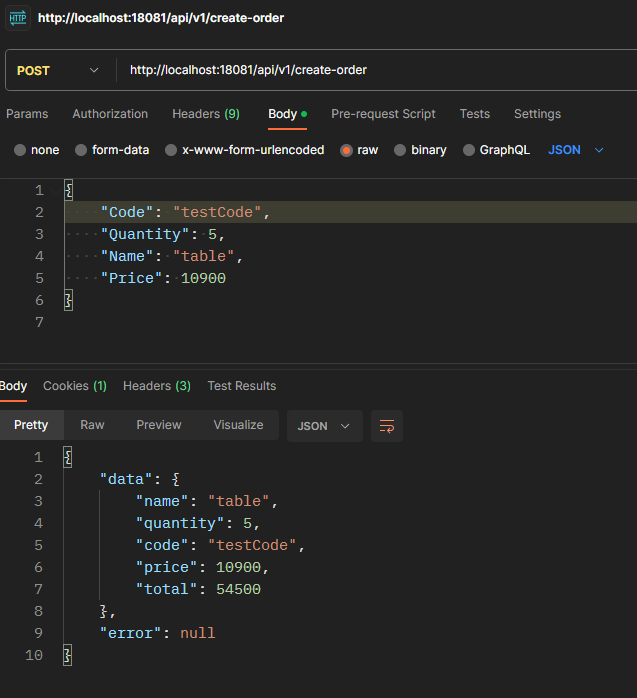


Пример работы orderService + storeroomService (общение через gRPC, отправка сообщение rabbitmq):

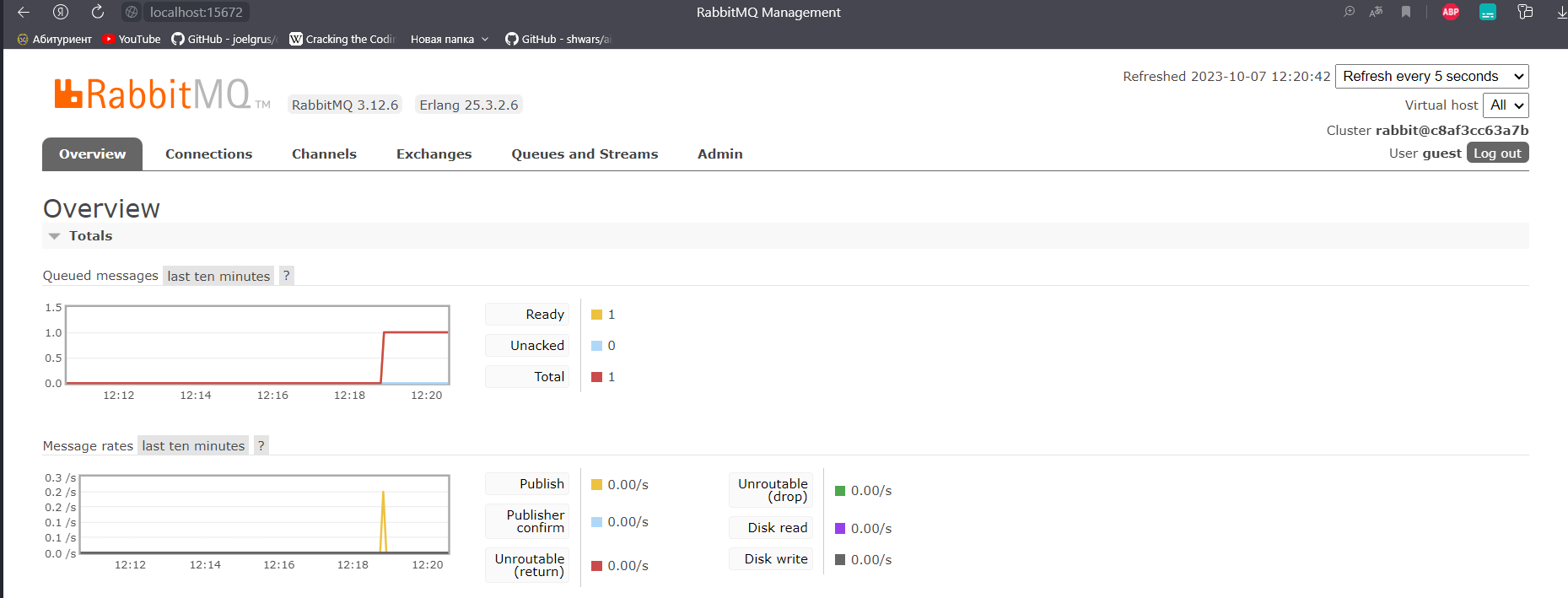
Добавление товара на склад:



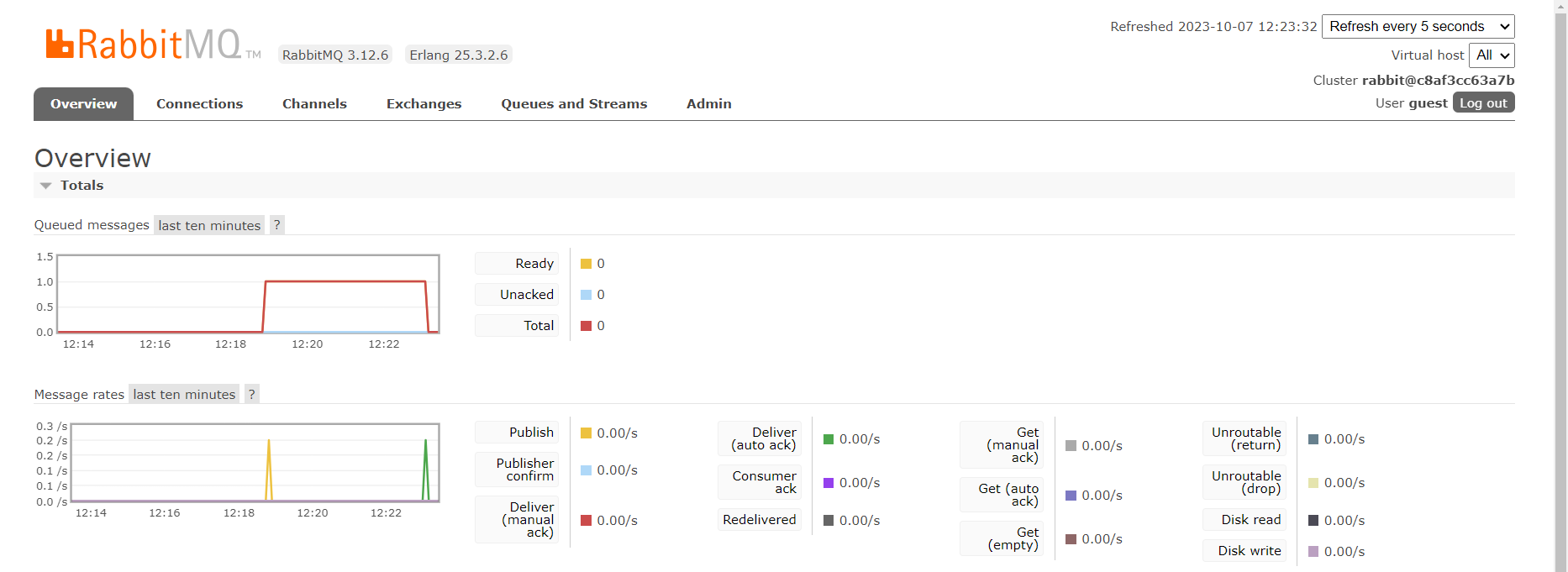
Создание заказа (идет проверка на складе через gRPC, отправка сообщение rabbitmq):



Просмотр отправленного сообщение rabbitmq:



Получатель сообщение из очереди rabbitmq:



При заказе больше, чем есть на складе заказ не создается:

