

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

з предмету «Проектування розподілених систем»

Виконав:

студент групи ІМ-31мн,

Онищук Микола

Київ 2024

## Завдання

- Реалізувати Enterprise паттерн асинхронної комунікації:
  - Event Sourcing модель з механізмом матеріалізації сутності АБО
  - 2-phase / 3-phase commit з використанням Брокера Повідомлень і 2ма Постачальниками Сервісу

## Виконання

Лабораторну роботу було виконано на мові Golang та розгорнуто у Docker. Також було використано RabbitMQ в якості брокера повідомлень для комунікації між споживачем та постачальником та PostgreSQL у якості бази даних.

Розроблена система містить по одному інстансу споживача та постачальника, а також брокер повідомлень для спілкування між ними та сервіс бази даних.

В даній роботі Enterprise патерн асинхронної комунікації було реалізовано за допомогою моделі Event Sourcing.

Сервіс споживача має 3 ендпоінти - “/post\_events” для створення та публікації нової події, “/get\_events” для отримання історії подій та “/get\_orders\_details” для перегляду актуального стану замовлень, створених чи оновлених на основі подій. Постачальник не має ендпоінтів, адже спілкування між ним та споживачем відбувається за допомогою Event Sourcing моделі: споживач записує події у базу даних і публікує їх у чергу RabbitMQ, а постачальник отримує ці події з черги, обробляє їх і оновлює стан своїх даних відповідно до отриманих змін.

Протестуємо роботу системи, надсилаючи запити за допомогою “Postman”.

Надішлемо POST-запит за адресою [http://localhost:5050/post\\_events](http://localhost:5050/post_events) та отримаємо результат у вигляді створеної відповідної події та повідомлення про її створення (рис. 1). Тіло запиту повинно містити тип події, а також ідентифікатори, замовлення, товару та кількість одиниць товару.

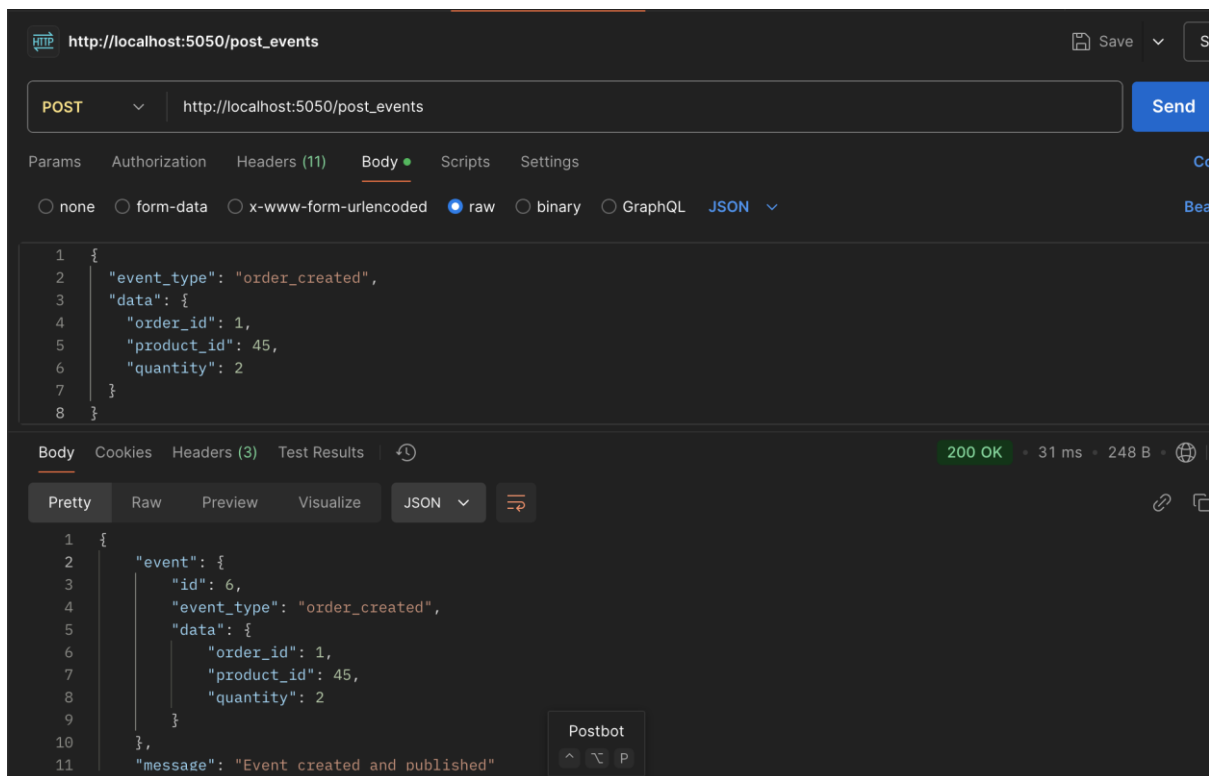


Рисунок 1 - Відправка запиту про завдання.

Після відправки кількох таких запитів (рис. 2), надішлемо GET-запит на [http://localhost:5050/get\\_events](http://localhost:5050/get_events) (рис. 3). Так ми отримаємо список усіх подій, що відбулися.

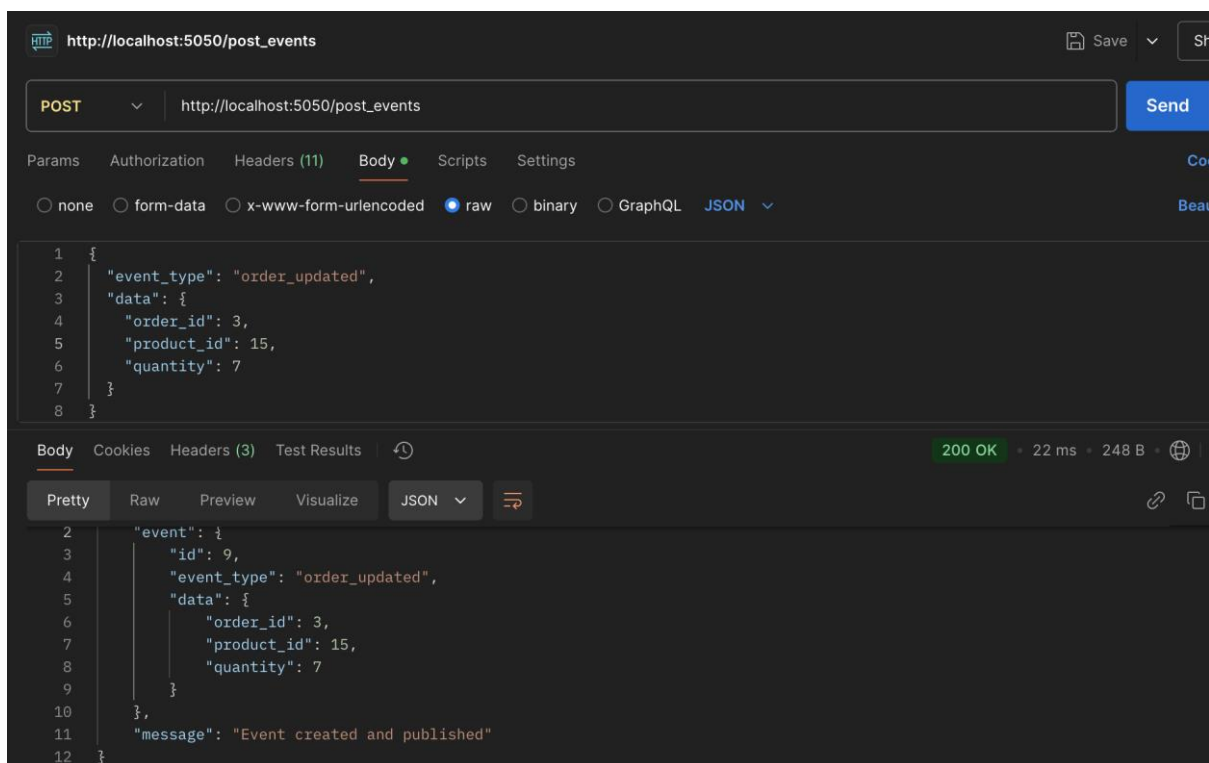


Рис. 3 - Відправка події типу "order\_updated".

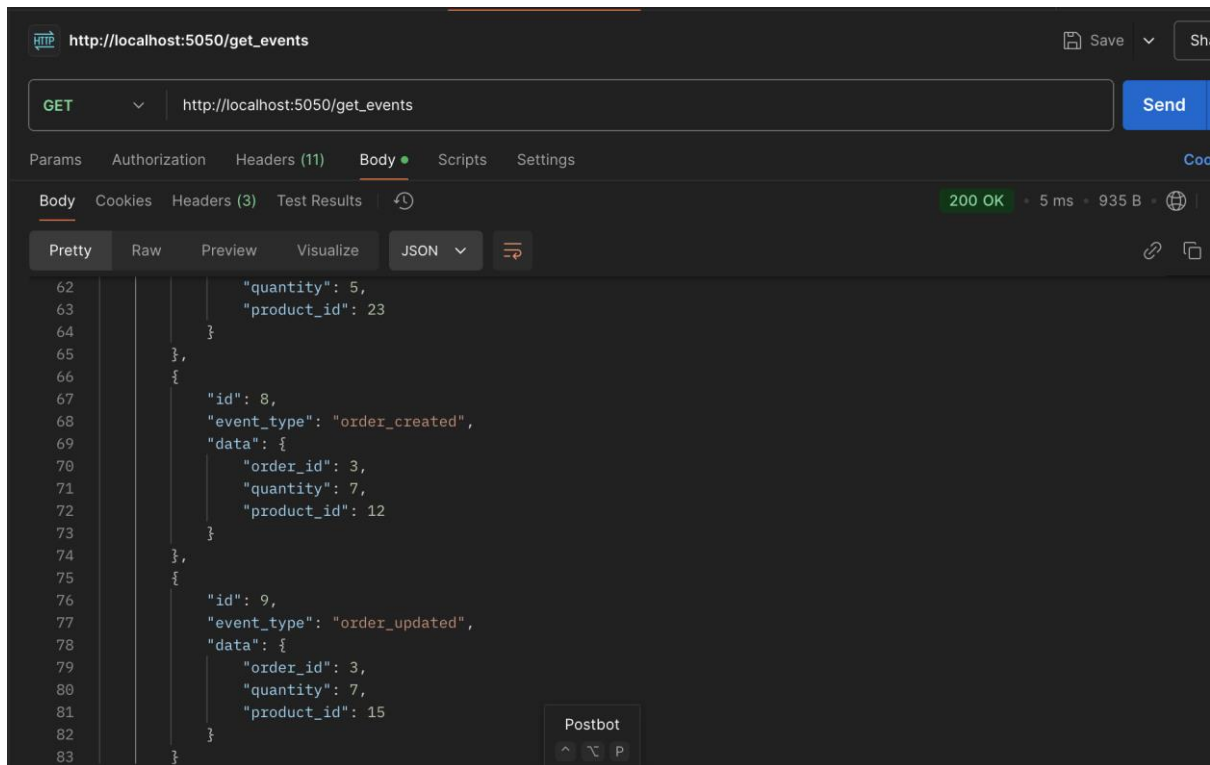


Рисунок 4 - Відправка запиту про отримання подій.

Тепер відправимо GET-запит на `http://localhost:5050/get_orders_details` для отримання поточних даних про всі замовлення (рис. 5).

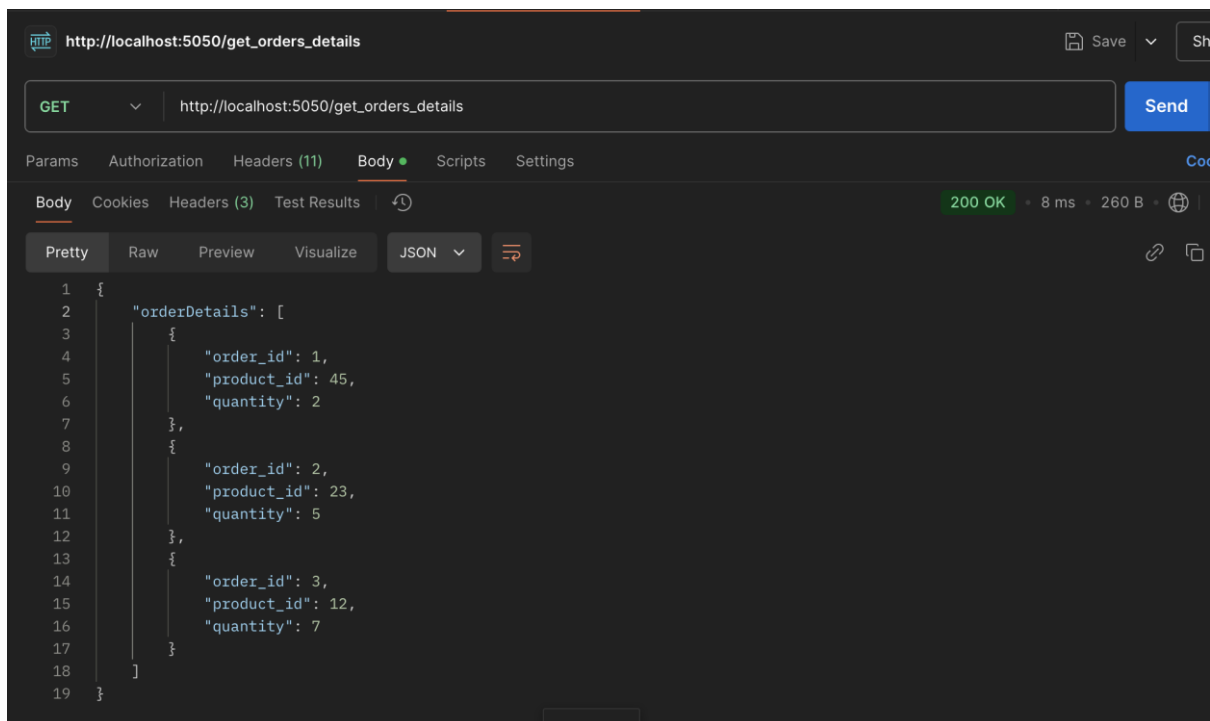


Рисунок 5 - Відправка запиту для отримання поточної інформації про замовлення.

Внаслідок надісланих запитів було отримано наступні логи (рис. 6).

```
provider-1 | [1.884ms] [rows:0] SELECT * FROM "order_details" WHERE "order_details"."order_id" = 3 ORDER BY "order_details"."order_id" LIMIT 1
provider-1 | 2025/01/05 16:52:09 Order details created: {OrderID:3 ProductID:12 Quantity:7}
rabbitmq-1 | 2025-01-05 16:53:18.101933+00:00 [info] <0.2414.0> accepting AMQP connection <0.2414.0> (172.19.0.5:51438 -> 172.19.0.3:5672)
rabbitmq-1 | 2025-01-05 16:53:18.107913+00:00 [info] <0.2414.0> connection <0.2414.0> (172.19.0.5:51438 -> 172.19.0.3:5672): user 'lab3' authenticated and granted access to vhost '/'
consumer-1 | 2025/01/05 16:53:18 Event published: {"data":{"order_id":3,"product_id":15,"quantity":7},"event_type":"order_updated"}
provider-1 | 2025/01/05 16:53:18 Received event: {order_updated map[order_id:3 product_id:15 quantity:7]}
rabbitmq-1 | 2025-01-05 16:53:18.114056+00:00 [info] <0.2414.0> closing AMQP connection <0.2414.0> (172.19.0.5:51438 -> 172.19.0.3:5672, vhost: '/', user: 'lab3')
provider-1 | 2025/01/05 16:53:18 Order details updated: {OrderID:3 ProductID:12 Quantity:7}
postgres-1 | 2025-01-05 16:56:13.224 UTC [14] LOG: checkpoint starting: time
postgres-1 | 2025-01-05 16:56:13.756 UTC [14] LOG: checkpoint complete: wrote 6 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.518 s, sync=0.012 s, total=0.532 s; sync files=6, longest=0.007 s, average=0.002 s; distance=2 kB, estimate=2 kB
```

Рисунок 6 - Логи при виконанні запитів.