

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

Рекечинський Дмитро

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 завдання

Проект було створено з реалізацію паттерну Enterprise на основі моделі Event Sourcing.

Для виконання завдання було створено 4 сервіси:

- Споживач сервісу consumer-service
- Постачальник сервісу provider-service
- Брокер повідомлень RabbitMQ (rabbitmq)
- Сервіс БД PostgreSQL (postgres)

Проблема з відстеженням статусу сервісу, яка виникла у 2 лабораторній роботі з RabbitMQ, поширилась також на PostgreSQL. Це вирішується за допомогою додавання healthcheck, і в Інтернеті радили використати для цього команду `pg_isready`.

Таким чином, конфігурація для healthcheck в PostgreSQL виглядає так:

healthcheck:

```
test: pg_isready
interval: 10s
timeout: 5s
retries: 10
start_period: 5s
```

Залежні контейнери налаштовуються у такий же спосіб, як описано у 2 лабораторній роботі:

`depends_on:`

`rabbitmq:`

`condition: service_healthy`

`postgres:`

`condition: service_healthy`

Втім, цей healthcheck працює не так, як слід... Кожні секунд 5 виникає помилка «FATAL: role "root" does not exist».

Подальше дослідження показало ось що:

- Якщо в команді `pg_isready` не вказано користувача явно, тоді перевірка виконується для поточного користувача.
- Команди в середовищі контейнера запускаються від користувача `root`, який має розширені права.
- За умовчуванням, PostgreSQL має користувача `postgres`. Але оскільки ми вказали змінну середовища `POSTGRES_USER: user`, то створюється користувач `user`. З цього випливає, що користувача `root` не передбачено жодним чином.

Це можна виправити за допомогою явного вказування користувача та бази даних у команді `healthcheck`:

`healthcheck:`

`test: "pg_isready -d ${POSTGRES_DB} -U ${POSTGRES_USER}"`

`interval: 10s`

```
timeout: 5s
retries: 10
start_period: 5s
```

Демонстрація результатів

```
~
> curl "http://localhost:8000/get/events" | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         0         0             Dload  Upload  Total  Spent  Left  Speed
100  429  100  429    0     0  23726      0 --:--:-- --:--:-- --:--:-- 23833
{
  "events": [
    {
      "id": 2,
      "event_type": "order_created",
      "timestamp": "2024-12-19T01:55:03.105Z",
      "data": {
        "order_id": 1,
        "product_id": 69420,
        "quantity": 1
      },
      "createdAt": "2024-12-19T01:55:03.105Z",
      "updatedAt": "2024-12-19T01:55:03.105Z"
    },
  ],
}
```

Рис. 1 — Відображення всіх подій

```

~
> curl "http://localhost:8000/get/projections" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    512    100    512     0     0   32168      0 --:--:-- --:--:-- --:--:-- 34133
{
  "projections": [
    {
      "order_id": 1,
      "product_id": 69420,
      "quantity": 1,
      "createdAt": "2024-12-19T01:55:03.201Z",
      "updatedAt": "2024-12-19T01:55:03.201Z"
    },
    {
      "order_id": 2,
      "product_id": 1488,
      "quantity": 2,
      "createdAt": "2024-12-19T02:02:48.626Z",
      "updatedAt": "2024-12-19T02:02:48.626Z"
    }
  ],
}

```

Рис. 2 — Відображення всіх сутностей

```

~
> curl -X POST "http://localhost:8000/events" -d '{"event_type": "order_created", "data": {"order_id": 11, "product_id": 11, "quantity": 2 }}' -H "Content-Type: application/json" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    347    100    256    100     91   2073     737 --:--:-- --:--:-- --:--:-- 2821
{
  "message": "Event created and published",
  "event": {
    "timestamp": "2024-12-19T14:53:28.048Z",
    "id": 5,
    "event_type": "order_created",
    "data": {
      "order_id": 11,
      "product_id": 11,
      "quantity": 2
    },
    "updatedAt": "2024-12-19T14:53:28.048Z",
    "createdAt": "2024-12-19T14:53:28.048Z"
  }
}

```

Рис. 3.1 — Створення нової сутності

```

~
> curl "http://localhost:8000/get/projections" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    140    100    140     0     0   40603      0 --:--:-- --:--:-- --:--:-- 46666
{
  "projections": [
    {
      "order_id": 11,
      "product_id": 11,
      "quantity": 2,
      "createdAt": "2024-12-19T15:37:04.425Z",
      "updatedAt": "2024-12-19T15:37:04.425Z"
    }
  ]
}

```

Рис. 3.2 — Результат виконання

```

~
> curl -X POST "http://localhost:8000/events" -d '{"event_type": "order_updated", "data": {"order_id": 11, "product_id": 11, "quantity": 4 }},' -H "Content-Type: application/json" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    347    100    256    100     91   2078     738 --:--:-- --:--:-- --:--:-- 2821
{
  "message": "Event created and published",
  "event": {
    "timestamp": "2024-12-19T15:22:35.350Z",
    "id": 6,
    "event_type": "order_updated",
    "data": {
      "order_id": 11,
      "product_id": 11,
      "quantity": 4
    },
    "updatedAt": "2024-12-19T15:22:35.350Z",
    "createdAt": "2024-12-19T15:22:35.350Z"
  }
}

```

Рис. 4.1 — Оновлення сутності

```

~
> curl "http://localhost:8000/get/projections" | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100    140    100    140     0     0   31574      0 --:--:-- --:--:-- --:--:-- 35000
{
  "projections": [
    {
      "order_id": 11,
      "product_id": 11,
      "quantity": 4,
      "createdAt": "2024-12-19T15:37:04.425Z",
      "updatedAt": "2024-12-19T15:41:29.509Z"
    }
  ]
}

```

Рис. 4.2 — Результат виконання

```

~
> curl -X POST "http://localhost:8000/events" -d '{"event_type": "order_deleted",
"data": {"order_id": 11 }}' -H "Content-Type: application/json" | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100    286    100    228    100     58   1921     488 --:--:-- --:--:-- --:--:-- 2423
{
  "message": "Event created and published",
  "event": {
    "timestamp": "2024-12-19T15:43:08.732Z",
    "id": 13,
    "event_type": "order_deleted",
    "data": {
      "order_id": 11
    },
    "updatedAt": "2024-12-19T15:43:08.732Z",
    "createdAt": "2024-12-19T15:43:08.732Z"
  }
}

```

Рис. 5.1 — Видалення сутності


```
~  
> curl "http://localhost:8000/get/projections" | jq  
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
                                 Dload  Upload  Total  Spent    Left     Speed  
100    18    100    18     0     0    1130      0 --:--:-- --:--:-- --:--:-- 1200  
{  
  "projections": []  
}
```

Рис. 5.2 — Результат виконання

Повна версія коду проекту розміщена за веб-адресою:

<https://github.com/rocket111185/distribution-systems/tree/release/lab3>