

УДК 004.75

*Голуб Д. К., магістрант,  
Вишняк М. Ю., к.т.н., доцент  
Харківський національний університет радіоелектроніки*

## **МЕТОДИ КОРЕЛЯЦІЇ ЗАПИТІВ ТА ВІДПОВІДЕЙ ПРИ АСИНХРОННІЙ ВЗАЄМОДІЇ З APACHE KAFKA**

Сучасна розробка програмного забезпечення все частіше базується на мікросервісній архітектурі [1], яка передбачає розбиття системи на набір незалежних сервісів. Ключовим викликом при проектуванні таких систем є організація надійної та ефективної комунікації між ними. Популярним патерном є асинхронний request-reply, який поєднує переваги синхронної взаємодії з перевагами асинхронності. Однак, такі брокери повідомлень як Apache Kafka, не мають нативної підтримки цього патерну, що ускладнює його використання в проектах, де Kafka вже задіяна для інших цілей (наприклад, обробка подій).

Apache Kafka працює за моделлю "publish-subscribe" [2], де споживачі читають повідомлення з топіків у власному темпі, і не передбачає прямого зв'язку між відправником конкретного повідомлення та його отримувачем. Для подолання цього обмеження було розроблено метод, що базується на кількох ключових компонентах.

Основа методу – присвоєння унікального ідентифікатора (UUID v4) кожному вихідному запиту. Цей correlationId додається до заголовків Kafka-повідомлення. Окрім цього, в заголовки додається параметр replyTo, який містить назву топіка, куди сервіс-виконавець має надіслати відповідь. Такий підхід дозволяє будь-якому споживачу, який отримав запит, зрозуміти, куди і з яким ідентифікатором потрібно відправити результат. Структурно механізм взаємодії наведено на рисунку 1 у вигляді UML Sequence Diagram.

Клієнтський компонент ініціює виклик методу сервісу Service A, передаючи необхідні вхідні дані. Після отримання запиту сервіс генерує унікальний ідентифікатор кореляції, який використовується для подальшого зіставлення відповіді із відповідним запитом. Одночасно створюється об'єкт Promise [3], що представляє майбутній результат виконання операції, і посилання на нього зберігається у внутрішній структурі даних типу Map<correlationId, Promise>. Після цього сервіс формує повідомлення-запит, яке містить ідентифікатор кореляції, назву топіка для відповіді та дані запиту, і публікує його до відповідного топіка запитів у брокері повідомлень.

Сервіс Service B у режимі безперервного споживання повідомлень отримує запит із топіка, виконує необхідну бізнес-логіку обробки та формує повідомлення-відповідь, що містить той самий correlationId і результат виконання. Це повідомлення публікується до топіка відповідей. Сервіс Service A, який постійно виконує цикл споживання повідомлень з цього топіка, отримує відповідь, зчитує ідентифікатор кореляції та знаходить відповідний об'єкт Promise у своїй внутрішній структурі. Після цього виконується операція resolve, у результаті чого початковий виклик методу отримує результат виконання запиту. Такий підхід дозволяє поєднати асинхронну взаємодію через брокер повідомлень із семантикою класичної моделі запиту та відповіді, забезпечуючи слабке зв'язування між мікросервісами та ефективну кореляцію повідомлень.

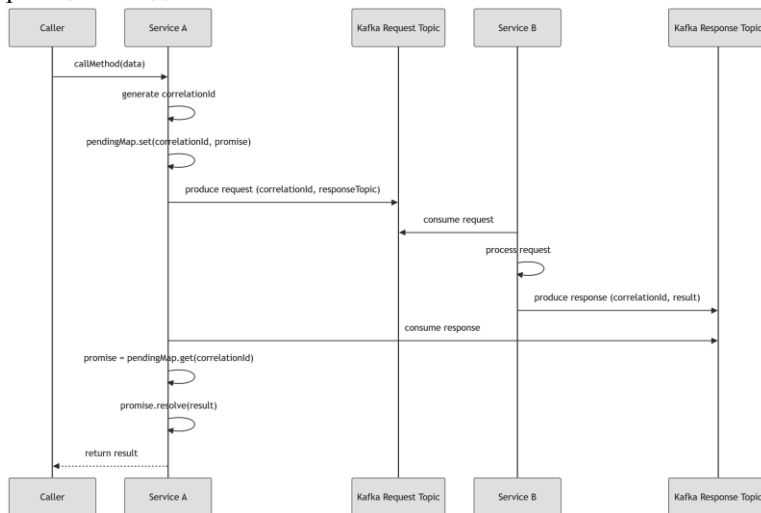


Рис.1. UML Sequence Diagram процесу виконання запиту

**Список використаних джерел:**

1. Richards M., Ford N. Fundamentals of Software Architecture. Sebastopol : O'Reilly Media, 2020. 383 с.
2. Documentation for Apache Kafka. URL: <https://kafka.apache.org/41/getting-started/introduction/> (дата звернення: 14.03.2026).
3. Pasquali S. Mastering Node.js. Birmingham : Packt Publishing Ltd., 2013. 347 с.