

## АСИНХРОНІСТЬ В SALESFORCE APEX

Асинхронний процес - це процес або функція, яка виконує завдання «у фоновому режимі» та без необхідності користувачеві чекати завершення операції. Зазвичай асинхронний Apex використовується для запитів до зовнішніх систем, операцій, що вимагають використання великої кількості ресурсів, і коду, який потрібно запускати в певний час із визначеним періодом повторення [1].

Salesforce.com використовує асинхронну структуру обробки на основі черги (рис.1) [2]. Цей фреймворк використовується для управління асинхронними запитами для декількох організацій у кожному екземплярі системи.

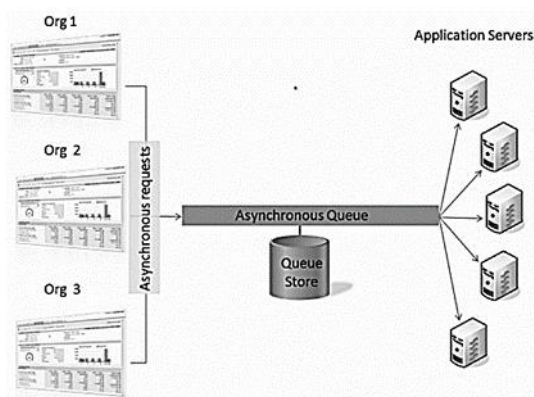


Рис. 1. Схема асинхронної системи обробки Salesforce на основі черги

Ключові переваги асинхронної обробки охоплюють наступні чинники:

- User efficiency. Скажімо, існує процес, який робить багато обчислень на дочірньому кастомному об'єкті щоразу, коли створюється parent-запис. Час, необхідний для виконання цих обчислень, може коливатися від незначного завантаження до повного блокування інтерфейсу користувача. За допомогою асинхронної обробки користувач може продовжувати свою роботу, поки обробка буде виконуватися у фоновому режимі.

- Масштабованість. Кожен екземпляр Salesforce має обмежений набір ресурсів. Ресурсами системи можна ефективно керувати, делегуючи виконання деякої логіки платформи на певний момент у майбутньому, коли її ресурси стануть доступними.

- Вищі governor-limits. Всі наступні асинхронні процеси в черзі запускаються у новому потоці, з вищими governor та execution лімітами. Це дозволяє уникати написання надлишкового коду та надмірного поглинання хмарного простору.

Наприклад, кількість запитів SOQL подвоюється зі 100 до 200 запитів при використанні асинхронних викликів. Загальний heap size і максимальний CPU час набагато більший для асинхронних викликів.

Асинхронний Apex можна використовувати в різних варіаціях, а саме:

- Future Methods - запускаються у власному потоці і не виконуються, поки ресурси системи не стануть доступними, зазвичай використовуються для здійснення web service callouts з Apex-тригерів.

- Batch Apex – використовується для написання громіздкої логіки, яка зазвичай перевищує стандартні ліміти обробки даних в окремій транзакції. Типові сценарії використання це - очищення даних або архівування записів. Варто розглянути можливість використання Batch Apex замість методів @future для асинхронної обробки великої кількості записів. Це буде ефективніше, ніж створення запиту @future для кожного запису.

- Queueable Apex – подібний до future methods, але забезпечує можливість додаткового job chaining і дозволяє використовувати більш складні типи даних для обробки. Може бути імплементовано для виконання послідовних операцій обробки із зовнішніми веб-службами. Однією з відмінних рис є можливість відстежувати стан виконання кожного окремого процесу за допомогою його унікального ідентифікатора.

- Scheduled Apex – автоматично планує запуск Apex логіки у визначений користувачем час, використовується для щоденних або щотижневих повторюваних завдань.

При роботі з асинхронними анотаціями в Salesforce важливо пам'ятати, що якщо в черзі перебуває понад 2000 необроблених запитів від однієї організації, будь-які додаткові запити від цієї самої організації будуть відкладені, поки черга обробляє запити від інших організацій. Варто мінімізувати кількість поданих пакетів одночасно, щоб мати гарантію, що подані партії не затримуються в спільній черзі.

### Список літератури

1. Write more efficient Apex code with asynchronous processing [Electronic resource]. – Access mode: [https://trailhead.salesforce.com/content/learn/modules/asynchronous\\_apex](https://trailhead.salesforce.com/content/learn/modules/asynchronous_apex)

2. Asynchronous Apex, The Savior from Salesforce Governor Limits [Electronic resource]. – Access mode: <https://salesforcenextgen.com/asynchronous-apex>