

JMS

JMS є ще однією технологією створення розподілених програм, заснованих на моделі обміну повідомленнями.

Вступ

JMS (Java Messaging System) представляє собою інтерфейс до зовнішніх систем, орієнтований на роботу через повідомлення. JMS є «старою» технологією - перша специфікація була опублікована в 1998р. На даний момент пакет `javax.jms` входить в комплект `jdk`, а Sun Application Server реалізує підтримку JMS в якості одного із сервісів.

При розробці JMS в якості основного завдання розглядалося створення узагальненого Java API для програм, орієнтованих на роботу з повідомленнями (message-oriented application programming) та забезпечення незалежності від конкретних реалізацій відповідних служб обробки повідомлень. Таким чином, програма, написана з використанням JMS, буде коректно працювати з будь-якою системою повідомлень, що підтримує цю специфікацію (або має відповідні інтерфейси).

Оскільки JMS є лише оболонкою або інтерфейсом, що описує доступні для програми методи, для роботи програми знадобиться певна реалізація цих інтерфейсів JMS, звана провайдером JMS. Такі реалізації створюються незалежними виробниками, і в даний час таких реалізацій існує досить багато (в тому числі, наприклад, реалізація, включена в Sun Application Server і розповсюджується разом з J2EE, а також MQSeries від IBM, служба JMS WebLogic від BEA, SonicMQ від Progress та інші).

Модель обміну повідомленнями (і JMS) зручно використовувати в тому випадку, якщо розподілена програма має наступні характеристики:

- взаємодія між компонентами є асинхронною; інформація (повідомлення) повинна передаватися кільком чи навіть всім компонентам системи (семантика передачі від одного до багатьох);
- інформація, що передається, використовується багатьма зовнішніми системами, частина з яких не відома на момент проектування системи або інтерфейси яких схильні до частих змін (концепція ESB - Enterprise Service Bus);
- компоненти, що обмінюються інформацією (повідомленнями), виконуються в різний час, що вимагає наявності посередника для проміжного зберігання переданої інформації.

Цілі JMS

Для кращого розуміння JMS потрібно знати цілі, які ставили перед собою творці специфікації JMS.

На даний момент на ринку існує безліч корпоративних систем обміну повідомленнями, і декілька компаній, які виробляють ці системи, були залучені в розробці JMS.

Ці наявні системи розрізняються за можливостями і функціональністю. Автори знали, що JMS буде занадто складною і громіздкою, якщо об'єднає всі функціональні можливості всіх існуючих систем. Також вони вважали, що не повинні обмежуватись лише функціями, спільними для всіх систем.

Автори вважали, що важливо було включити в JMS всю функціональність, необхідну для реалізації "удосконалених корпоративних програм".

Цілями JMS, як стверджується в специфікації, є:

- Визначити загальний набір концепцій і можливостей системи обміну повідомленнями.
- Мінімізувати концепції, які повинен вивчити програміст для використання корпоративної системи обміну повідомленнями.
- Максимізувати переносимість програм, що працюють з системою обміну повідомленнями.
- Мінімізувати роботу, необхідну для реалізації провайдера.

Надати клієнтські інтерфейси для обох доменів: "точка-точка" і `pub / sub`. "Домени" - це термін JMS для позначення моделей обміну повідомленнями, розглянутих раніше. Примітка: Провайдер не зобов'язаний реалізувати обидва домена.

Чого не забезпечує JMS

Деякі функціональні можливості, властиві MOM-продуктам (message oriented middleware), не розглядаються в JMS-специфікації. Ці можливості хоча і визнані авторами JMS важливими для розробки стійких програм систем обміну повідомленнями, але вважаються залежними від JMS-провайдера.

JMS-провайдери можуть реалізовувати такі функціональні можливості будь-яким бажаним способом:

- Розподіл навантаження і відмовостійкість.
- Система повідомлень і повідомлень про помилки та підказки.
- Адміністрування.
- Захист.
- Протокол зв'язку.
- Репозиторій типів повідомлень.

Архітектура JMS

Архітектура JMS виглядає наступним чином (Рис. 1.):

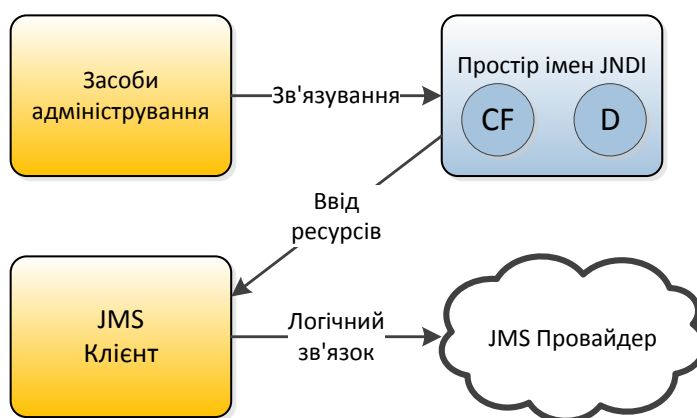


Рис. 1. Архітектура JMS

- Прикладні програми Java, які використовують JMS, називаються клієнтами JMS (JMS client);
- Система обробки повідомлень, яка керує маршрутизацією і доставкою повідомлень, називається JMS-провайдером (JMS provider);
- Програма JMS (JMS application) - це прикладна система, що складається з декількох JMS клієнтів, і, як правило, одного JMS-провайдера. JMS-клієнт, який посилає повідомлення, називається постачальником (producer). JMS-клієнт, що приймає повідомлення, називається споживачем (consumer). Один і той же JMS клієнт може бути одночасно і постачальником та споживачем у різних актах взаємодії;
- Повідомлення (Messages) - це об'єкти, що передаються і приймаються компонентами (клієнтами JMS);
- Засоби адміністрування (Administrative tools) - засоби управління ресурсами, що використовуються клієнтами.

JMS надає два підходи до передачі повідомлень. Перший називається «видання-підписка» (publish and subscribe) (Рис. 3.) і використовується в тому випадку, якщо повідомлення, відправлене одним клієнтом повинно бути отримано кількома.

Другий підхід називається «точка-точка» (point to point) (Рис. 2.) і служить для реалізації обміну повідомленнями між двома компонентами.

Специфікація JMS називає ці два підходи зонами повідомлень (messaging domains).

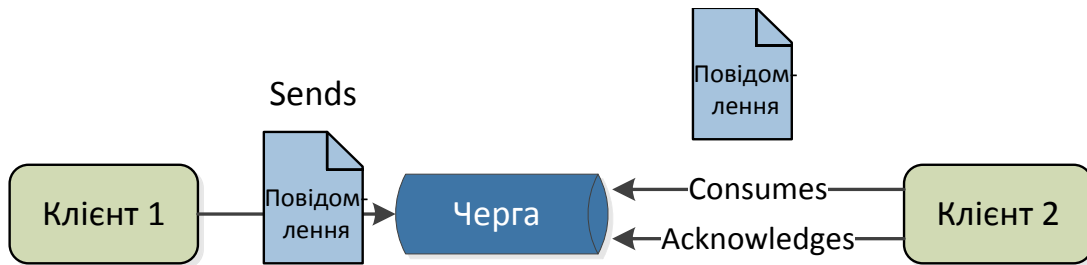


Рис. 2. Модель взаємодії «точка-точка»

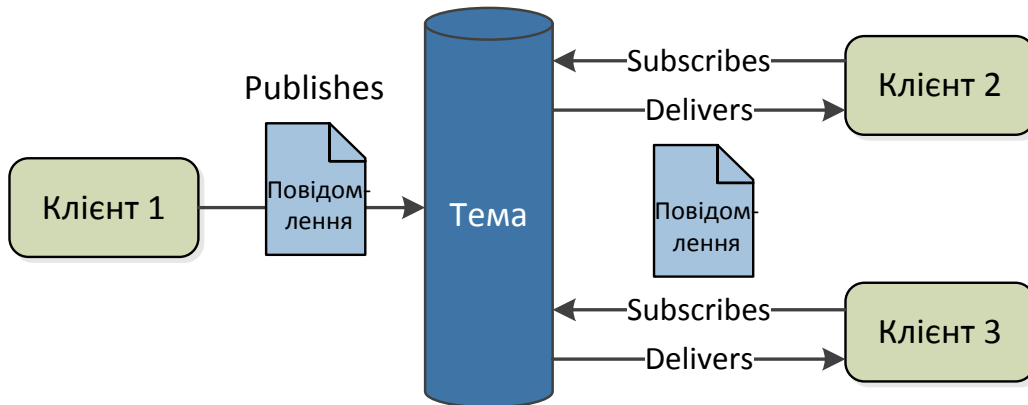


Рис. 3. Модель взаємодії «видання-підписка»

Модель взаємодії точка-точка

Модель передачі повідомлень «точка-точка» надає можливість клієнтам JMS посилати і приймати повідомлення (як синхронно, так і асинхронно) через віртуальні канали, які називаються чергами (queues). Модель передачі повідомлень «точка-точка» ґрунтується на методі опитування, при якому повідомлення явно запитуються (зчитуються) клієнтом з черги. Незважаючи на те, що читання з черги можуть здійснювати декілька клієнтів, кожне повідомлення буде прочитано тільки один раз - провайдер JMS це гарантує.

Модель взаємодії видання-підписка

При використанні моделі взаємодії «видання-підписка» один клієнт (постачальник) може посилати повідомлення багатьом клієнтам (споживачам) через віртуальний канал, який називається темою (topic). Споживачі можуть вибрати підписку (subscribe) на будь-яку тему. Всі повідомлення, що направляються в тему, передаються всім споживачам даної теми. Кожен споживач приймає копію кожного повідомлення. Модель передачі повідомлень видання-підписка, по суті, являє собою модель сервера, який ініціює з'єднання і «проштовхує» інформацію на клієнта. В JMS ця концепція реалізується за допомогою спеціальних «слухачів» (ліснерів), що реєструються в системі. При виникненні нової події ліснер, закріплений за даною темою, збуджується.

Слід зазначити, що при використанні моделі «видання-підписка» клієнти JMS можуть встановлювати довготривалі підписки, що дозволяють споживачам від'єднатися і пізніше знову підключитися і отримувати повідомлення, що надійшли під час відключення зв'язку.

19. Почтова система. Деяке «видання» відправляє лист всім своїм підписчикам. Модель передачі повідомлень «видання-підписки». Тип повідомлень `TextMessage`.
20. Система для пересилання файлів. Пересилається текстовий файл з довільної директорії, та зберігається на іншому клієнті в довільну директорію. Модель передачі повідомлень «точка-точка». Тип повідомлень `BytesMessage`.
21. Система для пересилання файлів. Пересилається текстовий файл з довільної директорії, та зберігається на іншому клієнті в довільну директорію. Модель передачі повідомлень «точка-точка». Тип повідомлень `MapMessage`.
22. Система для пересилання файлів. Пересилається текстовий файл з довільної директорії, та зберігається на іншому клієнті в довільну директорію. Модель передачі повідомлень «точка-точка». Тип повідомлень `ObjectMessage`.
23. Система для пересилання файлів. Пересилається текстовий файл з довільної директорії, та зберігається на іншому клієнті в довільну директорію. Модель передачі повідомлень «точка-точка». Тип повідомлень `StreamMessage`.
24. Система для пересилання файлів. Пересилається текстовий файл з довільної директорії, та зберігається на іншому клієнті в довільну директорію. Модель передачі повідомлень «точка-точка». Тип повідомлень `TextMessage`.
25. Система організації новинної розсилки. Видання повідомляє новину усім своїм підписчикам. Модель передачі повідомлень «видання-підписки». Тип повідомлень `BytesMessage`.
26. Система організації новинної розсилки. Видання повідомляє новину усім своїм підписчикам. Модель передачі повідомлень «видання-підписки». Тип повідомлень `MapMessage`.
27. Система організації новинної розсилки. Видання повідомляє новину усім своїм підписчикам. Модель передачі повідомлень «видання-підписки». Тип повідомлень `ObjectMessage`.
28. Система організації новинної розсилки. Видання повідомляє новину усім своїм підписчикам. Модель передачі повідомлень «видання-підписки». Тип повідомлень `StreamMessage`.
29. Система організації новинної розсилки. Видання повідомляє новину усім своїм підписчикам. Модель передачі повідомлень «видання-підписки». Тип повідомлень `TextMessage`.