

АЛЬТЕРНАТИВИ JAVA SPRING FRAMEWORK

Чорна Анастасія

Науковий керівник: канд.ф.-м. наук, доцент Малежик П.М.

*Національний педагогічний університет імені М.П. Драгоманова, м. Київ,
Україна*

В останнє десятиліття розробка програмного забезпечення на Java пов'язана з використанням Spring Framework. Більшість програмного забезпечення на Java повністю базується на Spring. Та на сьогодні з'являються фреймворки, що можуть стати альтернативою Spring. Такими фреймворками є Quarkus і Micronaut. В статті подано порівняння цих фреймворків та аналізуються основні причини потреби в альтернативних технологіях. Стаття містить опис основних переваг, що отримують Java розробники під час використання фреймворків Quarkus чи Micronaut.

Ключові слова: фреймворк, інтеграція залежностей, інверсія управління, SOLID принципи, контекст програми, рефлексія.

Постановка проблеми. Java — це одна з найпопулярніших мов програмування останні десять-п'ятнадцять років. Вона користується своєю популярністю через ряд інструментів, бібліотек та фреймворків, що в ній реалізовано. Крім цього вона має багато інших переваг у використанні. Якщо проаналізувати університетські та приватні Java курси, а також більшість вакансій та робочих позицій в компаніях з розробки програмного забезпечення, то можна помітити, що частіше за все об'єктом вивчення та вимог для роботи є не сама Java, а Spring.

Spring надає комплексну модель програмування та конфігурації для сучасних корпоративних програм на базі Java. Ключовим елементом Spring є інфраструктурна підтримка на рівні додатків: Spring зосереджується на підтримці корпоративних додатків, щоб команда розробників могла зосередитися на бізнес логіці додатків, без особливих затрат та роботи над середовищами розгортання.

Аналіз досліджень та публікацій. Як показують дослідження більше 80 відсотків розробників так чи інакше використовують Spring у роботі. Основною причиною цього є те, що використання фреймворку дозволяє писати якісний та зрозумілий програмний код. Більшість розробників, що використовують Spring в розробці додатків, відзначають, що це дозволяє їм писати код високої якості, економити час, а також користуватися зручними технологіями тестування коду. Це фактори дозволяють розробникам створювати якісні програмні продукти[1].

Популярність Spring серед розробників виникла також через архітектуру фреймворку, що забезпечує підтримку в програмних продуктах дотримання принципів Inversion of control (IoC) і Dependency Injection (DI). Кодова база самого фреймворку створена з урахуванням цих та інших принципів проектування, таких як SOLID. Програмний додаток на Spring — це проект, що складається з бітів, які існують в обмеженому контексті та мають свій життєвий цикл.

Метою статті є дослідження та порівняння Java фреймворків як альтернативи Spring фреймворку, визначення переваг та недоліків застосування кожного з них, виокремлення меж та цілей їхнього застосування.

Виклад основного матеріалу дослідження. Кілька років тому Spring був фреймворком, що не мав подібних аналогів відповідної якості, але не так давно з'явилися та продовжують розвиватися дві потужні спільноти Java фреймворків Quarkus та Micronaut. Ці фреймворки є досить Spring подібними, вони взяли від нього основні принципи та архітектурні рішення, тому Java розробникам буде досить легко переходити з одного фреймворку на інший для різних проектів. При цьому Quarkus та Micronaut вирішуються багато проблем з швидкістю роботи та об'ємами пам'яті, які досить часто виникають під час використання Spring для створення як монолітних так і мікросервісних проектів. Це досягнуто шляхом мінімізації використання рефлексії та проксі об'єктів для створення контексту.

Quarkus реалізує підмножину специфікації про Context and Dependency Injection, відому також як контексти та введення залежностей. Він має деякі з

добре підтримуваних функцій DI разом з обмеженнями, які можна просто конфігурувати. Micronaut зазвичай використовує просту інтеграцію залежностей під час компіляції, яка просто використовує рефлексію, без використання проксі.

Quarkus — це проект з відкритим кодом, ліцензований за ліцензією Apache версії 2.0. Продукт має досить якісну та розгорнуту документацію і є стабільним, оскільки багато функцій продукту забезпечуються екосистемами Hibernate ORM, Eclipse Vert.x, Netty, RESTEasy[4].

Від моменту створення фреймворк наростив достатньо велику спільноту користувачів та розробників і на сьогодні підтримує більшість сучасних інструментів, Spring в тому числі.

Основною перевагою Quarkus над традиційним стеком технологій є його швидкодія. На сайті спільноти розміщено порівняльні діаграми, що яскраво демонструють вигоду у використанні ресурсів пам'яті та часі роботи (Рис. 1). Максимальна оптимізація досягається шляхом використання Quarkus у поєднанні з GraalVM — віртуальної машини Java та JDK. GraalVM може працювати в контексті OpenJDK, щоб прискорити роботу Java-додатків за допомогою нової технології компіляції Just-In-Time. GraalVM бере на себе компіляцію байт-коду Java в машинний код.

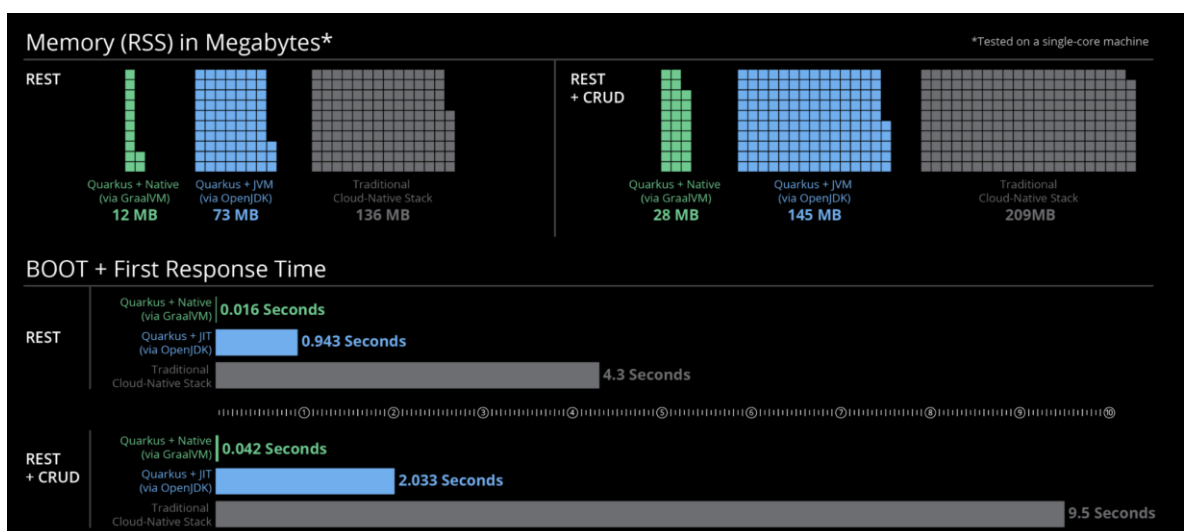


Рис. 1. Діаграми порівняння швидкодії Quarkus та традиційного стеку технологій [4]

Micronaut — це сучасний Java фреймворк з повною стековою структурою на базі JVM, призначена для створення модульних JVM-додатків з підтримкою Java, Kotlin та Groovy. Micronaut розроблений творцями фреймворка Grails і черпає натхнення з уроків, отриманих роками, створюючи реальні програми від монолітів до мікросервісів за допомогою Spring, Spring Boot та Grails.

Micronaut прагне надати всі інструменти, необхідні для створення додатків JVM, включаючи:

- Інтеграція залежностей та інверсія управління;
- Аспектно-орієнтоване програмування;
- Значення за промовчанням та автоматична конфігурація.

Micronaut прагне уникнути недоліків таких фреймворків, як Spring, Spring Boot та Grails, забезпечуючи:

- Швидкий час запуску;
- Зменшений обсяг необхідної пам'яті;
- Мінімальне використання рефлексії;
- Мінімальне використання проксі;
- Відсутність генерації байт-коду під час виконання;
- Просте модульне тестування[2].

Важко і не потрібно визначати, який з фреймворків краще інших, і повністю відмовитися від інших. Це неправильний підхід до вибору технологій. Для правильного порівняння потрібно подивитись на фреймворки з різних сторін, залежно від призначення та стеку технологій, що потрібні для реалізації проекту. Для прикладу Micronaut підтримує Java, Groovy та Kotlin мови. Quarkus підтримує Java, Kotlin та Scala, але не підтримує Groovy. Тоді як Spring підтримує ці всі чотири мови програмування. За кількістю підтримуваних мов Spring однозначно виграє. Якщо брати у порівняння інструменти та фреймворки, що підтримуються Java фреймворками, то Quarkus тут виграє навіть у Spring, тоді як Micronaut підтримує найменшу кількість сторонніх інструментів.

Якщо порівнювати підтримку командного рядка та його розширень, то тут Spring програє і Micronaut, і Quarkus. При цьому Quarkus має велику перевагу через підтримку hot reloading розширення, що дозволяє не перезавантажувати проект під час зміни програмного коду. Для розробників це розширення є дуже зручним та корисним.

Порівнюючи підтримку фреймворками cloud технологій, то тут однозначно Micronaut та Quarkus виграють у Spring. Вони підтримують не тільки Cloud, а також Kubernetes — розширювану платформу з відкритим кодом для управління навантаженням та сервісами за допомогою контейнеризації[3].

Висновки. Аналізуючи характеристики та особливості цих фреймворків очевидно, що в деяких аспектах нові фреймворки перевершують звичний для всіх Spring. Зрозуміло, що вони не можуть замінити всі інструменти, що пропонує Spring, але з часом та збільшенням спільноти Quarkus та Micronaut ці фреймворки будуть ставати більш надійними та підтримувати ширший ряд технологій на вимогу спільноти. На напрямки та темпи розвитку цих фреймворків напряму впливає зацікавленість та інтерес зі сторони розробників та корпорацій. Тому потрібно знайомитися з новими та незвичними фреймворками, знаходити їх переваги та сфери застосування, створювати нові навчальні та практичні курси для розробників на базі цих фреймворків, збільшувати стек можливих та доступних технологій на ринку.

Список літератури

1. Diffblue Survey Finds 86 Percent of Java Developers Rely on Spring Framework [Електронний ресурс] // GlobeNewswire. – 2021. – Режим доступу до ресурсу: <https://www.globenewswire.com/news-release/2021/04/19/2212157/0/en/Diffblue-Survey-Finds-86-Percent-of-Java-Developers-Rely-on-Spring-Framework.html>
2. Introduction [Електронний ресурс] // Micronaut. – 2021. – Режим доступу до ресурсу: <https://docs.micronaut.io/latest/guide/#introduction>
3. Micronaut vs Quarkus vs Spring Boot Reactive Framework Deep Comparison [Електронний ресурс] // Regupathi Thankaraj. – 2021. – Режим доступу до ресурсу:

<https://regupathit.medium.com/quarkus-vs-micronaut-a-deep-comparison-84cae4fea966> Quarkus

[Электронный ресурс] // Quarkus. – 2021. – Режим доступа до ресурсу: <https://quarkus.io/faq/>