# Порівняння Spring MVC, Spring Webflux, Quarkus, Micronaut

Керівник:
асистент кафедри, Андрощук Максим Віталійович
Виконав:
Загривий Олег Сергійович

# Мета роботи

Порівняти та дослідити відмінності Spring MVC, Webflux, Quarkus та Micronaut з метою визначення оптимального фреймворку у певних ситуаціях розробки. Результати дослідження допоможуть встановити рекомендації щодо найкращої сфери використання кожного з фреймворків.

# Етапи роботи

1. Огляд фреймворків:

Spring, Spring Boot, Spring MVC, Spring Webflux

Quarkus

Micronaut

2. Порівняння можливостей фреймворків:

Обробка запитів, безпека, автоматичне перезавантаження, наявна документація та розмір спільноти

Загальні переваги та недоліки кожного фреймворку

3. Побудова проєктів та вимірювання характеристик:

Реалізація серверної частини

Тестування швидкості роботи

Тестування продуктивності за допомогою ApacheBench

Тестування використання пам’яті після навантаження

# Загальні поняття

- Фронтенд (англ. Front-end) – клієнтська частина.
- Бекенд (англ. Back-end) – серверна частина.
- Фреймворк (англ. Framework, каркас) – набір інструментів, бібліотек та шаблонів, що спрощують процес розробки шляхом надання загальної структури та стандартів, розширюючи функціональність що надає мова програмування або платформа.

# Огляд Spring та Spring Boot

- Spring – це фреймворк для розробки ПЗ на платформах Java, Kotlin та Groovy. Він надає широкий спектр інструментів та компонентів для побудови різних типів додатків.
- Spring Boot – це підпроєкт Spring, який спрощує розробку додатків на основі Spring, надаючи конфігурації за замовчуванням та автоматичне налаштування, що дозволяє швидше розгортати додатки.

# Spring MVC

Spring MVC (Model-View-Controller) – один з найпопулярніших фреймворків, частина Spring, використовує підхід MVC для розділення бізнес-логіки додатка, представлення даних та управління інтерфейсом.

# Spring Webflux

Spring Webflux – реактивний фреймворк, що також входить до складу Spring. Він надає можливості для розробки реактивних додатків, що працюють на основі асинхронних та неблокуючих операцій.

# Quarkus

Quarkus – супершвидкий фреймворк для розробки додатків, включаючи хмарні, та мікросервісів. Він створений з метою забезпечити максимальну продуктивність та ефективне використання ресурсів.

# Micronaut

Micronaut – альтернатива Quarkus для розробки мікросервісів та хмарних додатків. Він також розроблений з метою підвищення продуктивності, зменшення затримки, та ефективнішого використання ресурсів.

# Порівняння можливостей

Багато можливостей за якими можна порівняти ці фреймворки, сконцентруємося на таких:

- Маршрутизація та обробка запитів
- Безпека
- Автоматичне перезавантаження
- Наявна документація та спільнота

# Маршрутизація

- У Spring MVC, Quarkus та Micronaut маршрутизація заснована на використанні анотацій, таких як ‘@RequestMapping’ або ‘@GetMapping’ для визначення шляхів та обробників запитів. Це надає зручну модель для визначення маршрутів та їх обробки.
- У випадку Spring Webflux, маршрутизація використовує функціональний підхід, де маршрути визначаються за допомогою функцій, що приймають HTTP-запит та повертають відповідь, що дозволяє розробникам більшу гнучкість та контроль над маршрутизацією. Функції можуть бути складнішими та включати більше логіки, що дозволяє складні сценарії.

# Обробка запитів

- Через те, що Spring MVC, Quarkus та Micronaut використовують анотації, обробка запитів відбувається послідовно, один за одним, тобто кожен запит блокує виконання потоку до завершення обробки запиту.
- У випадку Spring Webflux, обробка запитів відбувається асинхронно та неблокуючим способом. Це дозволяє фреймворку обробляти багато запитів одночасно та мати більшу пропускну здатність

# Безпека

- Quarkus та Micronaut мають вбудовану підтримку безпеки з різними конфігураційними можливостями.
- Spring Webflux та Spring MVC не мають вбудовану безпеку, а надають можливості безпеки через бібліотеку Spring Security.

# Автоматичне перезавантаження та перекомпіляція

- Quarkus та Micronaut підтримують автоматичне перезавантаження коду та гарячу перекомпіляцію (hot reloading), що дозволяє розробникам швидше вносити зміни до своїх додатків та бачити результати без повного перезапуску.

# Наявна документація та активна спільнота

Spring Framework має найбільшу спільноту, кількість ресурсів та проєктів, що може викликати схильність до використання вже знайомого та перевіреного досвідом фреймворку.

Quarkus та Micronaut новіші, отже вони більш привабливі для розробників що бажають експериментувати з новими технологіями.

# Переваги та недоліки

- Spring MVC:
  - +: широке використання, велика спільнота, стабільність, простота
  - -: традиційний підхід, синхронне програмування
- Spring Webflux:
  - +: реактивний підхід, ефективна обробка багатьох запитів, підтримка нестандартних протоколів
  - -: потребує додаткових знань, обмежена підтримка інтегрованих інструментів
- Quarkus:
  - +: швидкість запуску та відгуку, низьке споживання пам’яті, гарячий перезапуск
  - -: менша популярність, обмежена підтримка бібліотек, можливі проблеми зі сумісністю
- Micronaut:
  - +: аналогічна до Quarkus швидкість та ефективність, вбудована підтримка обробки залежностей
  - -: аналогічні до Quarkus, також деякі бібліотеки вимагають додаткового налаштування

# Тестування та вимірювання характеристик

Кожен проєкт має однакову структуру з одним маршрутом:

“/hello/{target}” (GET)

, де target – параметр. Маршрут повертає відповідь з привітанням та ціллю, наприклад, “Hello World!”.

Цей маршрут не впливає на порівняння фреймворків, а служить для вимірювання характеристик.

Micronaut використовує MicronautTest, Quarkus – QuarkusTest, Spring MVC та WebFlux – Spring Test. Всі тести однакової структури.

# Результати тестування (1)

- Час компіляції та виконання тестів
  - Micronaut та Quarkus використовують оптимізацію на етапі компіляції, тому вони повільніше.
- Час запуску (dev mode, release mode)
  - Quarkus повільніший у dev mode, через використання JVM замість свого GraalVM.
  - Spring Webflux і Spring MVC в загалому повільніші за Quarkus та Micronaut (за винятком попереднього випадку).
- Час до першої відповіді
  - Micronaut та Quarkus вдвічі швидше за Spring MVC та Webflux.

# Результати тестування (2)

- Продуктивність
  - Micronaut найкращий, Quarkus трохи повільніший, WebMVC найповільнійший, Webflux виключено з тесту (через відмову keep-alive).
- Використання пам’яті
  - Micronaut і Quakus мають менше споживання пам’яті, приблизно на 20 відсотків.
- Вплив обмеження розміру пам’яті на продуктивність
  - Quarkus: найбільша стабільність, втрати приблизно 9%.
  - Micronaut: втрати 25%.
  - Spring WebMVC: найбільші втрати, до 42%.

# Технічні висновки

Micronaut і Quarkus перевершують Spring Webflux і Spring MVC у всіх параметрах крім часу компіляції та тестування, що пов’язано з відсутністю аналізу коду на етапі компіляції у Spring фреймворках.

# Висновки

Було досліджено чотири фреймворки, побудовано ідентичні проєкти та протестовано їх ефективність.

Результати - кожен з фреймворків має свою нішу і може бути підходящим для конкретних вимог і сценаріїв розробки.

Spring MVC - найпопулярніший і найпростіший фреймворк.

Spring Webflux пропонує реактивний та асинхронний підхід але вимагає додаткових знань.

Quarkus та Micronaut мають незначні відмінності, але вони обидва дуже швидкі, ефективні та мають додаткові вбудовані можливості. Маючи низьке споживання пам’яті, вони обидва краще підходять для мікросервісних додатків.