

Аналітичний огляд фреймворків для створення додатків.

Вінницький національний технічний університет

Анотація

Проаналізовано і розглянуто переваги та недоліки Spring фреймворку для створення додатків та його функціональні можливості.

Ключові слова: фреймворк, додаток, розробка

Abstract

It was analyzed and considered the advantages and disadvantages of the Spring framework for creating an application and its functionality

Keywords: *framework, application, development.*

Вступ

У епоху стрімкого розвитку ІТ-індустрії та попиту створення програмного забезпечення інструменти, що дозволяють здійснити цю задачу набирають колосальних обертів. Одним із популярних інструментів створення програмного забезпечення є Spring framework. Spring Framework — це платформа Java, яка надає повну підтримку інфраструктури для розробки Java-додатків. Spring дозволяє розробляти програми за допомогою POJO. Перевага використання тільки POJO полягає в тому, що вам не потрібен продукт-контейнер EJB, такий як сервер додатків, є можливість використовувати лише контейнер сервлетів, такий як Tomcat. На противагу Spring Framework будуть розглядатися ще такі фреймворки, як Micronaut, Helidon SE та Ktor, що є його аналогами.

Метою роботи є проаналізувати низку переваг та недоліків фреймворку Spring відносно інших фреймворків.

Результати дослідження

У даний час немає дефіциту фреймворків для створення мікросервісів на Java. Найстарішим фреймворком є Spring. Він дуже популярний з кількох причин: по-перше, підхід до впровадження залежності Spring заохочує писати тестований, якісний код, по-друге, простий у використанні, але з потужними можливостями керування транзакціями бази даних, по-третє, Spring спрощує інтеграцію з іншими фреймворками Java, такими як JPA/Hibernate ORM, Struts/JSF/i т.д. веб-фреймворками, і є сучасним і незамінним фреймворком Web MVC для створення веб-додатків. Впровадження залежностей (DI) та інверсія управління (IoC) дозволяють писати незалежні один від одного компоненти, що дає переваги у командній розробці, переносимості модулів тощо. Spring IoC контейнер управляє життєвим циклом Spring Bean і налаштовується на зразок JNDI lookup. Проект Spring містить у собі безліч підпроектів, які зачіпають важливі частини створення софту, такі як веб-сервіси, веб-програмування, робота з базами даних, завантаження файлів, обробка помилок і багато іншого. Все це налаштовується в єдиному форматі і спрощує підтримку програм. Усі вище перераховані можливості роблять його таким популярним інструментом для створення додатків. Але існують також інші фреймворки, що є досить молодими, тому не містять багатьох функціональних можливостей, що є у Spring.

Першим розглянемо Micronaut. Micronaut — це фреймворк на JVM для побудови легких модульних додатків. Має кілька можливостей, які роблять його відмінним фреймворком для розробки хмарних додатків (cloud-native). Він підтримує безліч механізмів виявлення сервісів (відкриття сервісів), такі як

Eureka і Consul, а також працює з різними системами розширеного трасування (розподілене відстеження), такими як Zipkin і Jaeger. Додатково, він надає підтримку створення лямбда функцій AWS, що дозволяє легко створювати безсерверні додатки.

Наступним будемо розглядати Helidon. Фреймворк був створений в Oracle для внутрішнього використання, згодом став відкритим. Існують дві моделі розробки, засновані на цій структурі: Standard Edition (SE) і MicroProfile (MP). API MicroProfile, як правило, дотримуються декларативного стилю, який інтенсивно використовує анотації. Helidon SE – це наш власний реактивний API. Він використовує функціональний стиль програмування з невеликим використанням анотацій. Найкраще використовувати цей фреймворк при написанні невеликих додатків, що не працюють із базами даних.

Останнім йде Ktor. Ktor — це асинхронний фреймворк з відкритим кодом для створення мікросервісів і веб-додатків. Він був розроблений спільно з Kotlin компанією JetBrains. Його легко встановити та використовувати. Його асинхронна властивість дозволяє йому отримувати кілька запитів завдяки використанню coroutines, а також він є мультиплатформним. Має багато можливостей, але його недоліками є те, що він не підтримує Java, сумісний лише із Kotlin, та використовується лише для розробки мобільних додатків, а не десктопних додатків.

На наведеній нижче таблиці (таб.1) представлений порівняльний аналіз фреймворку Spring відносно інших фреймворків.

Таблиця 1.1 – Порівняльний аналіз фреймворку Spring відносно інших фреймворків

Фреймворк	Мови програмування, що підтримує	Час компіляції	Можливість інтеграції із third-party	Підтримка автоматизованого тестування	Впровадження залежностей	Наявність документації	Безпека
Micronaut	Groovy, Java, Kotlin	1.48 с	низька	середня	не підтримує	мінімальна	низька
Helidon	Java, Kotlin	2,2 с	низька	низька	не підтримує	мінімальна	низька
Ktor	Kotlin	1.4 с	низька	низька	не підтримує	мінімальна	низька
Spring	Groovy,Java, Kotlin	1.33 с	висока	висока	підтримує	повна	висока

На основі всього вище написаного можемо зробити висновок, що хоча Micronaut, Helidon SE та Ktor потребують менше пам'яті для своєї роботи, вони програють відносно фреймворка Spring, оскільки не мають багато функціональних можливостей, таких як можливість інтеграції із third-party, можливість побудувати стійкі сервіси із Cloud, висока безпека додатків, підтримка automation тестування та наявність детальної документації.

Висновки

Розглянуто та проаналізовано переваги та недоліки фреймворку Spring відносно інших фреймворків та його особливості, що використовуються в середовищі розробки IntelliJ IDEA для реалізації додатків. Показано, що фреймворк Spring має низку переваг та дає можливість розробляти додатки із інтеграцією із Spring Security, шаблонізатором Thymeleaf та Spring Data.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Micronaut Framework Documentation, [Електронний ресурс]. – Режим доступу: <https://micronaut.io/docs/>

2. Helidon Framework Documentation, [Електронний ресурс]. – Режим доступу: <https://helidon.io/>
3. Spring Framework Documentation, [Електронний ресурс]. – Режим доступу: <http://docs.spring.io/>
4. Koin Framework Documentation, [Електронний ресурс]. – Режим доступу: <https://insert-koin.io/>

Боднар Карина Олександрівна – студентка групи 1КІ-20м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: karishabodnar99@gmail.com.

Науковий керівник: Захарченко Сергій Михайлович — кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, м.Вінниця, e-mail: <zahar@vntu.net>

Bodnar Karina – a student of group 1CE-20m, Faculty of information technologies and computer engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: karishabodnar99@gmail.com.

Scientific supervisor: Zakharchenko, S.M. — Candidate of Technical Sciences, Associate Professor of Computer Technology, Vinnytsia National Technical University, Vinnitsa, e-mail: <zahar@vntu.net>