

МЕТОДИ ВИКОРИСТАННЯ SPRING BOOT У ХМАРНИХ ОБЧИСЛЕННЯХ

Дмитренко В., Федорченко В.М.

Харківський національний університет радіоелектроніки, Харків, Україна

У сучасну епоху хмарних обчислень широко використовується повностековий застосунок – розробка як фронтенда (клієнтської частини), так і бекенда (серверної частини) веб-додатка. У таких додатках бекенд являє собою серце і суть програми. Без масштабованого рішення додаток не може бути розширено для задоволення сучасних потреб середовища хмарних обчислень.

Тому реалізація та системний дизайн бекенд-додатку мають бути ретельно продумані. Spring Boot - це надбудова над Spring, яка спрощує конфігурацію та розгортання додатків.

Основним його завданням є автоматизація процесу конфігурації, що дає змогу мінімізувати ручні налаштування та прискорити процес створення застосунку.

Тобто він надає готові шаблони для створення мікросервісів і веб-додатків, що робить його чудовим вибором для старту нових проєктів [1, 2].

Коли ми маємо справу з постійними змінами та проблемами, використання корисних інструментів є життєво важливим для створення безпечних та масштабованих серверних систем. Це дослідження зосереджується на використанні Spring Boot.

Spring Boot – це механізм для багатьох швидких розробок додатків, що складаються з різноманітних мікросервісних архітектур та різноманітних корпоративних систем.

Великими перевагами Spring Boot є мінімізація готової конфігурації та інтеграція з усім стеком Spring. Іншою користю є сильна підтримка сучасних способів упакування (наприклад, Docker) разом з управлінням (наприклад, Kubernetes), а також певний відомий набір інструментів (як-от Spring Security та Spring Data JPA).

Основною проблемою в розробці хмарних застосунків є велика складність коду, а саме великий обсяг стандартного коду, який з'являється при автоматизації багатьох процесів.

Звичайно ж використання декількох засобів таких як JPA (це Java Persistence API) разом із Spring Data для налагодження взаємодії з різними базами даних (зокрема PostgreSQL), значно покращує процес розробки.

Але це може привести до великого, майже безмежного росту думок на рівні доступу до даних, у такому вигляді науковий аналіз однозначно вказує на те, що коли автоматизація дещо зменшується, зокрема, обмежується використання створення запитів у Spring Data, створюються умови для помітного підвищення адаптивності та безпеки кожної спеціальної системи [3].

Ще однією важливою перевагою Spring Boot для розробки хмарних застосунків є управління залежностями.

У чистому Spring розробникам доводиться вручну додавати та керувати версіями залежностей, що може призвести до конфліктів.

Spring Boot вирішує цю проблему, надаючи попередньо налаштовані стартові залежності, які гарантують сумісність усіх бібліотек.

Наприклад, якщо взяти до уваги Spring Security, можна просто додати Spring Security Starter Dependency до вашого проекту та забезпечити захист від таких атак, як фіксація сеансу, клікджекінг, підробка міжсайтових запитів тощо, а також розширювану підтримку як для автентифікації, так і для авторизації та керування сеансами [4].

Spring Boot у поєднанні з Spring Security, Docker та Kubernetes стає сильним інструментом для створення великих та безпечних додатків.

Завдяки автоматичній налаштуванні розробники можуть швидко запускати моделі і відмінно впроваджувати структуру мікросервісів.

Проте надмірна залежність від шаблонного коду може ускладнити проект, так що збалансований підхід до автоматизації – наприклад обмеження використання авто-сгенерованих запитів в Spring Data – допомагає зберегти контроль над кодовою базою.

Використання наскрізного шифрування E2EE й інтегрованих механізмів Spring Security таких як OAuth2 та JWT забезпечує надійний захист даних і API, зберігаючи при цьому гнучкість в розробці.

Ще одна перевага Spring Boot є його здатність управляти залежностями що робить процес створення і підтримки додатків більш стабільними.

Список літератури

1. Спілке Лауренціу, “Spring Security in Action” | <https://surl.li/uoyhsk> . – 2024. – С. 480.
2. Spring і Spring Boot: основні відмінності та особливості використання // [Електронний ресурс] <https://foxminded.ua/ru/razlichiya-mezhdu-spring-i-spring-boot/> Режим доступу 30.10.24
3. Greg L. Turnquist, “Learning Spring Boot 3.0: Simplify the development of production-grade applications using Java and Spring, 3rd Edition” | <https://github.com/PacktPublishing/Learning-Spring-Boot-3.0-Third-Edition> , 2022.
4. Moises Macero Garcia, Tarun Telang, “Learn Microservices with Spring Boot 3: A Practical Approach Using Event-Driven” | <https://github.com/Book-Microservices-v3>