

# ВИКОРИСТАННЯ SPRING BOOT ФРЕЙМВОРКУ У РОЗРОБЦІ СЕРВЕРНИХ ЗАСТОСУНКІВ

Вінницький національний технічний університет

## *Анотація*

*Робота присвячена розгляду переваг та актуальності використання Spring Boot фреймворку. Проаналізовано його основні технології, засоби та інструменти. Досліджено характерні особливості.*

**Ключові слова:** розробка програмного забезпечення, фреймворк, оптимізація, функціональність.

## *Abstract*

*The paper is dedicated to exploring the advantages and relevance of using the Spring Boot framework. Its main technologies, tools, and instruments are analyzed. The characteristic features are investigated.*

**Keywords:** software development, framework, optimization, functionality.

## **Вступ**

У сучасному світі розробки програмного забезпечення швидкість та ефективність вирішально важливі. Для розробників виникає завдання створити надійне та функціональне програмне забезпечення, яке швидко відповідає на потреби користувачів. У цьому контексті фреймворки стають незамінними інструментами. Одним із таких потужних фреймворків є Spring Boot, який здобуває все більшу популярність у розробників програмного забезпечення. Spring Boot надає зручність у створенні різноманітних застосунків, починаючи від невеликих мікросервісів та закінчуючи складними корпоративними системами. Його широкий спектр функцій та можливостей робить його привабливим для розробників будь-якого рівня. У цій статті було розглянуто ключові аспекти використання Spring Boot [1] у розробці програмного забезпечення, зокрема його взаємодію з базами даних, підтримку RESTful API, інтеграцію з іншими сервісами та можливості розгортання на віддалених серверах.

Підтримка баз даних, зокрема PostgreSQL, у Spring Boot фреймворку є ключовим аспектом, який дозволяє розробникам створювати потужні та надійні додатки.

## **Основні засоби та технології**

Інтеграція з реляційними базами даних у Spring Boot фреймворку базується на принципах об'єктно-реляційного відображення (ORM). ORM використовується для відображення об'єктів додатку на записи в базі даних та автоматичного виконання операцій з ними. Цей підхід не лише спрощує роботу з базами даних, але й робить код більш модульним та легким у розумінні. Використання Spring Data JPA разом з Spring Boot дозволяє автоматизувати створення репозиторіїв та забезпечити підтримку реляційних моделей даних за допомогою простих анотацій та стандартів JPA. Для використання доступні власноруч написані SQL-запити або згенеровані автоматично за ключовими словами, що робить розробку надзвичайно оптимізованою і гнучкою. Такі принципи забезпечують чистоту коду та полегшують розширення програмного забезпечення. Вони не лише спрощують процес розробки і є зручними у використанні, але й є функціонально обґрунтованими, оскільки сприяють підвищенню продуктивності та якості програмного забезпечення.

Багатий вибір інструментів у Spring Boot дозволяє розробникам зосередитися на розробці функціональності, замість того, щоб турбуватися про рутинні завдання. Один із найпопулярніших інструментів - Lombok, спрощує процес створення POJO-класів. Він автоматизує генерацію методів, таких як геттери, сеттери, equals та hashCode, що дозволяє зменшити кількість написаного коду та підвищити читабельність. Використання Lombok у Spring Boot дозволяє розробникам зосередитися на бізнес-логіці та швидше впроваджувати нові функції, зменшуючи час розробки та ризик виникнення помилок. Крім того, інші інструменти, такі як Actuator [3], DevTools, Security, доповнюють функціонал Spring Boot, надаючи розробникам широкий набір інструментів для різних вимог проекту.

Використання цих інструментів у Spring Boot значно оптимізує процес розробки, та покращує ефективність робочого часу.

Взаємодія з іншими API у Spring Boot є важливою складовою для створення розширених та інтегрованих сервісів. Фреймворк надає зручні інструменти для спрощення цього процесу, включаючи вбудовану підтримку для викликів RESTful API. Для цього можна використовувати бібліотеку Spring Web, яка дозволяє легко взаємодіяти з іншими веб-службами через HTTP-запити. Крім того, Spring Boot надає можливість інтеграції з іншими API за допомогою бібліотек, таких як Feign або Retrofit, що спрощують взаємодію з віддаленими сервісами та дозволяють швидко створювати клієнтські компоненти для взаємодії з іншими API. Це дозволяє розробникам легко інтегрувати свої застосунки з різними сервісами та джерелами даних, що розширює можливості їх функціональності та забезпечує більш гнучке та розширене використання. Таким чином, взаємодія з іншими API у Spring Boot сприяє створенню більш інтегрованих та функціональних застосунків.

Підтримка RESTful API у Spring Boot є однією з його ключових можливостей, що дозволяє розробникам створювати потужні та гнучкі веб-сервіси. Фреймворк надає простий та зручний спосіб створення API, зокрема для виконання операцій CRUD (створення, читання, оновлення, видалення). Завдяки анотаціям Spring та підтримці об'єктно-реляційного відображення (ORM), розробники можуть швидко створювати контролери, які відповідають за обробку HTTP-запитів та взаємодію з базою даних.

Крім базових операцій CRUD, Spring Boot також підтримує різноманітні технологічні можливості для реалізації розширених функціональностей у веб-сервісах. Наприклад, вбудовані анотації Spring Security дозволяють забезпечити аутентифікацію та авторизацію користувачів, забезпечуючи безпеку API [3].

Spring Boot надає можливість використання інструментів валідації даних, таких як анотації Bean Validation, що дозволяють перевіряти вхідні дані та валідувати їх на предмет відповідності певним правилам. Це робить API надійнішим та зменшує ризик введення некоректних даних.

Загалом, підтримка RESTful API в Spring Boot дозволяє розробникам створювати сучасні, ефективні та безпечні веб-сервіси з широким спектром функціональності.

Розгортання на віддалених серверах у Spring Boot включає у себе не лише сам процес розміщення застосунка на віддаленому сервері, але й автоматизацію цього процесу, зокрема автоматичне підняття серверу.

Spring Boot надає зручні інструменти для розгортання програмних застосунків на віддалених серверах, таких як AWS EC2 або Google Cloud Platform Instance. Розробники можуть легко створювати виконувані файли JAR або WAR, які містять усі необхідні залежності, включаючи вбудований сервер застосунків, такий як Tomcat або Jetty. Після цього програма може бути запущена на віддаленому сервері за допомогою стандартних команд запуску Java.

Більше того, Spring Boot надає можливість автоматизації підняття серверу через такі інструменти, як Docker або Kubernetes. Це дозволяє автоматично створювати та розгортати контейнеризовані версії сервісів на віддалених серверах з мінімальними зусиллями з можливістю організації мікросервісності архітектури.

Такий підхід забезпечує швидке та ефективне розгортання програмного забезпечення, зменшуючи час і витрати на управління серверами та забезпечуючи надійність та масштабованість виконання.

### **Висновки**

Spring Boot - це потужний фреймворк для розробки програмного забезпечення, який надає розробникам зручність та ефективність у створенні різноманітних застосунків. Використання Spring Boot в сучасній розробці дозволяє ефективно використовувати інструменти та технології, такі як підтримка PostgreSQL та інших баз даних, багатий вибір інструментів, взаємодія з іншими API, підтримка RESTful API та можливість розгортання на віддалених серверах типу AWS EC2.

Цей фреймворк дозволяє розробникам зосередитися на розробці бізнес-логіки та функціональності програмного забезпечення, мінімізуючи рутинні завдання та спрощуючи процес розробки. Завдяки його гнучкості та розширюваності, Spring Boot відповідає на потреби як початківців, так і досвідчених розробників, надаючи їм інструменти для створення якісного та ефективного програмного забезпечення. Таким чином, Spring Boot є незамінним інструментом у сучасній розробці програмного забезпечення, який допомагає прискорити процес розробки та забезпечити високу якість готового продукту.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Spring Boot. URL: <https://spring.io/projects/spring-boot> (дата звернення: 15.03.2024).
2. Long J., Bastani K. Cloud Native Java: Designing Resilient Systems with Spring Boot, Spring Cloud, and Cloud Foundry. O'Reilly Media, 2017. 648 p.
3. Walls C. Spring Boot in Action. Manning Publications, 2016. 264 p.

**Кіпоре́нко Ілля Євге́нович** – студент групи 4ПІ-20Б, факультет інформаційних технологій та комп'ютерної інженерії, група 4ПІ-20Б, Вінницький національний технічний університет, Вінниця, e-mail: [kiporenkoillia@outlook.com](mailto:kiporenkoillia@outlook.com)

**Кателі́нко́в Дени́с Іва́нович** - кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [fuzzy2dik@gmail.com](mailto:fuzzy2dik@gmail.com).

**Кіпоре́нко Ілля Євге́нович** - student of Software Engineering Department, Vinnytsia National Technical University, Faculty of Information Technology and Computer Engineering, Vinnytsia.

**Katielnikov Denys Ivanovich** - PhD, Associate Professor of Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, E-mail: [fuzzy2dik@gmail.com](mailto:fuzzy2dik@gmail.com).