

## **SPRING BOOT ЯК ІНСТРУМЕНТ ДЛЯ ПРИСКОРЕННЯ РОЗРОБКИ ВЕБ-ДОДАТКІВ**

*НТУ «Дніпровська політехніка»*

**Манін Богдан Максимович, Лось Єлизавета Ігорівна**  
**Наукові керівники: к.т.н., доц. Реута О.В., к.т.н., доц. Ширін А.Л.**

Для розробки сучасних веб-додатків можна використовувати багато різних технологій. Серед найчастіше використовуваних можна виділити наступні:

- Servlet API;
- Spring Framework;
- Spring Boot.

Кожна технологія має свої переваги та недоліки, а також використовується у різних галузях завдяки своєму вбудованому функціоналу. Але зважаючи на те, що ці технології призначено для розробки веб-додатків, розглянемо їх з точки зору клієнт-серверної архітектури.

Servlet API - це стандартна бібліотека, яка надає можливість розробляти веб-додатки на платформі Java EE. Servlet є Java-класом, який містить ряд методів для обробки запитів від клієнтів та відправки відповідей на них. Кожен запит, що надходить на веб-сервер, передається відповідному сервлету, який генерує відповідь у вигляді HTML-сторінки, JSON-даних, або будь-якому іншому типу відповідей.

Servlet API є нижнім рівнем інтерфейсу, тому для створення веб-додатків з його використанням необхідно здійснити більше зусиль. Розробка додатків на основі Servlet API вимагає знання HTTP-протоколу та Java-технологій, таких як Java Server Pages (JSP) та Java Server Faces (JSF).

Створення сервлету складається з кількох етапів. Спочатку необхідно створити клас, який реалізує інтерфейс `javax.servlet.Servlet` та успадковується від класу `HttpServlet`. Далі необхідно перевизначити методи життєвого циклу, такі як `init()` та `destroy()`, які викликаються при запуску та завершенні роботи Servlet.

Для обробки запитів методи використовують два параметри - об'єкт `HttpServletRequest`, що містить запит від клієнта, та об'єкт `HttpServletResponse`, що відповідає на запит.

Сервлети зберігаються в `servlet`-контейнерах, таких як `WildFly` або `Tomcat`, які потребують окремого встановлення та налаштування й є окремими елементами веб-застосунку.

Spring Framework – це програмний каркас з відкритим програмним кодом для платформи Java, який містить контейнери з підтримкою інверсії управління.

Основні особливості Spring Framework можуть бути використані будь-яким додатком Java, але є розширення для створення вебдодатків на платформі Java EE. Попри це, Spring Framework не нав'язує якоїсь конкретної моделі програмування. Spring Framework став популярним у спільноті Java-розробників як альтернатива, або доповнення моделі Enterprise Java Bean (EJB).

Spring Boot – популярний інструмент для розробки веб-додатків та мікрослужб на основі Java, створений на платформі Spring Framework. Його особливість полягає в тому, що розробнику майже не потрібно прописувати залежності між модулями проекту та створювати файли конфігурацій. Spring Boot автоматично створює залежності та конфігурує проект. Також, для запуску додатку, написаного на основі Spring Boot не потрібно завантажувати проект у сервлет-контейнер або інший локальний веб-сервер. Для цього Spring Boot поставляється із вбудованими контейнерами, такими як Tomcat, Jetty та Undertow.

Також перевагою використання Spring Boot є полегшена робота із базою даних (БД). На відміну від Servlet API, де у кожному методі класу опрацювання бази даних прописується SQL-запит відповідно до задач методу, для роботи з БД в Spring Boot достатньо створити інтерфейс repository, який успадковує інтерфейс JpaRepository. Для того, щоб звернутися до БД, достатньо лише декларувати метод в інтерфейсі, назвавши його відповідно до того, яку сутність ми хочемо повернути. За обробку запитів на основі назв сетодів відповідає вбудований препроцесор. Наприклад, щоб повернути всі сутності з БД можна оголосити метод findAll, а щоб знайти сутність в БД за її ім'ям достатньо декларувати метод findAllByName(String name). Вбудований в Spring Boot препроцесор знайде відповідний до назви методу SQL-запит у своїй бібліотеці.

Проект, який написаний на основі Spring Boot має певну структуру, яку можна розподілити на окремі компоненти. Це значно прискорює розробку додатку. Для цього використовуються анотації. Наприклад, для того, щоб позначити клас сутності, достатньо позначити його анотацією @Entity, а для того, щоб Spring Boot міг визначити контролер, його достатньо позначити анотацією @Controller.

Тож, можемо виділити наступні переваги використання інструменту Spring Boot фреймворку Spring перед використанням Servlet API:

- Швидка та легка розробка програм на основі Spring.
- Автоконфігурація всіх компонентів для Spring виробничого рівня.
- Готові вбудовані сервери (Tomcat, Jetty та Undertow), що забезпечують прискорене та більш продуктивне розгортання програм.
- Відсутність необхідності конфігурації БД та створення таблиць вручну.
- Відсутність конфігурації XML.

Spring Boot значно полегшує використання Spring Framework. Нижче наведено функції Spring Boot, які забезпечують зменшення часу розробки web-додатків та дозволяють завершувати проекти із меншими зусиллями:

- Spring Boot не вимагає розгортання war-файлів;
- Створює автономні програми та запускає їх прямо з командного рядка;
- Допомогає безпосередньо вбудувати у додаток контейнери Tomcat, Jetty або Undertow;
- Допомогає спростити відстежування слабко пов'язаних між собою блоків, не вимагаючи ніякої функції конфігурації;
- Має додаткову функціональність «з коробки», що дозволяє виконувати просте налаштування і керування проекту та зменшити обсяг вихідного коду.

Проаналізувавши використання Servlet API та Spring Boot у проектуванні веб-додатків можемо зробити висновок, що використання Spring Boot значно полегшує розробку та пришвидшує її темпи. Це відбувається завдяки відсутності необхідності конфігурувати класи проекту; можливостям роботи із сутностями БД напряму як з об'єктами класів; наявності вбудованих сервлет контейнерів та наявності анотацій, що дозволяє автоматично конфігурувати класи та робити Dependency injection (DI). Однак, використання технології Servlet API має переваги у швидкості роботи, через відсутність необхідності завантажувати у проект всіх залежностей, незалежно від необхідності в окремих з них, завдяки чому вихідний проект має меншу вагу ніж проект з використанням Spring Boot. Тому використання Spring Boot доцільне у проектах середньої тяжкості, які орієнтовані на бізнес логіку. Тоді як Servlet API доцільно використовувати у великих монолітних проектах, які потребують швидкодії та відповідності до чітких стандартів мови Java та її пакетів.

#### **Перелік посилань:**

1. What is Java Spring Boot? An intro to Spring Boot – the popular, Java-based tool for developing web apps and microservices [Електронний ресурс]. URL: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-java-spring-boot/> (дата звернення: 27.02.2023).
2. Spring Boot Tutorial [Електронний ресурс]. URL: [https://www.tutorialspoint.com/spring\\_boot/spring\\_boot\\_introduction.htm](https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm) (дата звернення: 27.02.2023).
3. Gutierrez Felipe. Pro Spring Boot 2: An Authoritative Guide to Building Microservices, Web and Enterprise Applications, and Best Practices / Apress Berkeley, CA, 2019 – 511 p.
4. What is Tomcat? The original Java servlet container, 2022 [Електронний ресурс]. URL: <https://www.infoworld.com/article/3510460/what-is-apache-tomcat-the-original-java-servlet-container.html> (дата звернення: 27.02.2023).