Spring Framework

Spring is a comprehensive open-source Java framework for developing enterprise-grade and web-based applications. It promotes loosely coupled, testable, and maintainable code using core features like Dependency Injection (DI) and Aspect-Oriented Programming (AOP).

Overview

- Created By: Rod Johnson
- First Release: October 2002 (based on ideas from his book "Expert One-on-One J2EE Design and Development")
- Current Maintainers: Spring.io, a project under VMware (formerly Pivotal)
- Latest Stable Version (as of 2025): Spring Framework 6.x

Purpose

Spring was created to simplify Java Enterprise Edition (J2EE) development by:

- Reducing boilerplate code
- Managing dependencies automatically (IoC & DI)
- Making the application loosely coupled and testable

Key Concepts in Spring Framework

1. Inversion of Control (IoC):

- The control of object creation and dependency management is handed over to the Spring container.
- Achieved through Dependency Injection.

2. Dependency Injection (DI):

- Instead of creating objects manually, Spring injects dependencies into classes automatically.
- Types:
    - Constructor-based Injection
    - Setter-based Injection
    - Field Injection (Not recommended)

3. Aspect-Oriented Programming (AOP):

- Allows you to separate cross-cutting concerns (e.g., logging, security).

- Implemented via concepts like Aspect, Join Point, Advice, Pointcut.

4. Spring Beans:

- A bean is an object that is managed by the Spring container.
- Defined in configuration (@Component, XML, or Java Config).

5. ApplicationContext:

- It is the Spring container that instantiates, configures, and wires the beans.

Spring Boot

Spring Boot is a rapid application development framework built on top of Spring. It removes the boilerplate configuration and helps in creating stand-alone production-ready apps quickly.

Overview

- Created By: Spring Team at Pivotal Software (now part of VMware)
- First Release: April 2014
- Primary Goal: Rapid development with minimal configuration using prebuilt setups and defaults
- Current Stable Version (as of 2025): Spring Boot 3.x

Spring Boot was designed to remove complexity from traditional Spring applications. It allows developers to:

- Skip XML configurations
- Use embedded web servers (Tomcat, Jetty)
- Auto-configure common patterns
- Quickly build and test production-grade applications

Key Features of Spring Boot:

Feature	Description
Auto-Configuration	Automatically configures beans based on dependencies
Standalone	No need for external servers – runs with embedded Tomcat/Jetty
Spring Initializr	Web tool to generate Spring Boot skeleton projects
Starter Dependencies	Predefined POMs like `spring-boot-starter-web`, `spring-boot-starter-data-jpa`
Actuator	Ready-to-use production monitoring endpoints
No XML Needed	Pure Java/Annotation-based configuration