Spring Framework

---

1. Overview

Spring Framework is a powerful, lightweight **Java framework** primarily used to build enterprise-grade applications.

- First released in 2003.
- Provides comprehensive infrastructure support for developing Java applications.
- Promotes **loose coupling** through **Dependency Injection (DI)**.
- Modular architecture — you use only what you need.
- Widely used for building web applications, microservices, batch processing, and more.
- Forms the basis for the popular **Spring Boot** project, which simplifies Spring app development.

---

2. Core Concepts

Dependency Injection (DI) / Inversion of Control (IoC)

- Objects don't instantiate their dependencies themselves.
- Instead, dependencies are **injected** by the Spring container.
- Reduces coupling, improves testability and modularity.
- Configured via XML, annotations (@Autowired), or Java config.

Aspect-Oriented Programming (AOP)

- Separates cross-cutting concerns (logging, security, transactions).
- Implements aspects (modular units) that can be applied across the application.
- Declarative approach to add behaviors to methods without changing code.

Beans and ApplicationContext

- **Beans** are objects managed by the Spring container.

- **ApplicationContext** is the IoC container that instantiates, configures, and manages beans lifecycle.
- Bean scopes: singleton (default), prototype, request, session.

---

3. Spring Modules

Module	Description
Core Container	DI and IoC support
Spring AOP	Aspect-oriented programming
Data Access/Integration	JDBC, ORM (Hibernate, JPA), transaction management
Web	Web MVC framework, REST APIs
Security	Authentication and authorization
Test	Testing support with JUnit and TestNG

---

4. Spring MVC

- Model-View-Controller framework to build web applications.
- **DispatcherServlet** routes requests to controllers.
- Controllers handle requests, returning views or data.
- Supports RESTful web services with JSON/XML responses.
- Flexible view resolution (JSP, Thymeleaf, FreeMarker).

---

5. Spring Boot

- Opinionated framework to **simplify Spring app development**.
- Provides auto-configuration, embedded servers (Tomcat, Jetty).
- Starter dependencies reduce manual configuration.
- Enables standalone apps with minimal setup.
- Supports production-ready features: metrics, health checks, externalized config.

---

6. Data Access

- Supports relational databases with **Spring Data JPA**, Hibernate.
- Simplifies database interaction with repository interfaces.
- Supports NoSQL databases (MongoDB, Redis) via Spring Data projects.
- Declarative transaction management with @Transactional.

---

7. Security

- **Spring Security** offers authentication, authorization, and protection against common attacks.
- Supports OAuth2, LDAP, JWT, and custom security configurations.
- Integrates easily with web and REST APIs.

---

8. Microservices

- Spring Cloud builds on Spring Boot to support microservices patterns:
    - Service discovery (Eureka)
    - Circuit breakers (Hystrix)
    - Config management (Config Server)
    - API Gateway (Zuul, Spring Cloud Gateway)
    - Distributed tracing (Sleuth)

---

9. Testing

- Supports unit, integration, and mock testing.
- Provides **Spring TestContext Framework**.
- Easy to bootstrap Spring contexts during tests.

---

10. Advantages

- Comprehensive ecosystem.
- Loose coupling with DI improves maintainability.

- Large community and excellent documentation.
- Integration with modern tools and cloud-native support.
- Strong support for both monolithic and microservice architectures.