

# **Java & Spring Boot**

## **Complete Detailed eBook (2026 Edition)**

Includes Java 25 LTS and Java 26 features; Spring Boot 3/4 updates

Author: M365 Copilot | Date: Feb 9, 2026

# Table of Contents

1. Executive Summary
2. Java Platform Today (Java 25 LTS)
3. What's New in Java 26 (Mar 17, 2026 target)
4. Performance & GC Deep Dive (G1, AOT Object Caching, Vector API)
5. Concurrency: Structured Concurrency & Virtual Threads
6. Security & Networking: HTTP/3, TLS, Final-Field Warnings
7. Spring Framework 6 & Spring Boot 3: The 2026 Baseline
8. Spring Boot 4.0 Highlights and Migration Notes
9. Building Cloud-Native Services (Config, Observability, AOT/Native)
10. Architecture Patterns (Hexagonal, DDD, Modular Monolith, Microservices)
11. Practical Guides & Recipes (Profiles, Testing, CI/CD)
12. Migration Playbook (JDK & Boot) with Checklists
13. Performance Tuning Cheat Sheet
14. Security Hardening Checklist
15. References

# 1. Executive Summary

Java continues to evolve on a 6-month cadence with Long-Term Support (LTS) anchors; Java 25 is the current LTS while Java 26 (due Mar 17, 2026) adds previews such as primitive types in patterns/switch, structured concurrency (6th preview), HTTP/3 for the client, AOT object caching, and more. On the Spring side, Spring Framework 6 and Spring Boot 3 standardized on Java 17+, Jakarta EE 9+, observability via Micrometer, and AOT support; Spring Boot 4 (Nov 2025) builds atop Framework 7 with modularization, null-safety (JSpecify), and Java 25 support while retaining Java 17 compatibility.

## **2. Java Platform Today (Java 25 LTS)**

Java 25 (LTS, Sep 2025) provides a modern baseline for enterprises: virtual threads (GA in 21), pattern matching enhancements, and stable platform APIs. Teams should standardize builds and containers on Java 25 for production, while evaluating Java 26 features in staging environments.

### 3. What's New in Java 26 (target Mar 17, 2026)

- Primitive types in patterns / instanceof / switch (4th preview).
- Ahead-of-time (AOT) object caching for faster startup/warmup across GCs.
- Vector API (11th incubator) for SIMD-style performance.
- Structured Concurrency (6th preview) for scoped lifecycles.
- Lazy Constants (2nd preview) and PEM encodings (2nd preview).
- G1 GC throughput improvements via reduced synchronization.
- HTTP/3 support in the Java HTTP Client.
- Removal of legacy Applet API; warnings for deep reflection mutating finals.

## 4. Performance & GC Deep Dive

Java 26 targets both startup and throughput: AOT object caching allows loading cached objects in a GC■agnostic format; G1 reduces coordination overhead; the Vector API keeps advancing for data■parallel workloads. Combine these with modern container flags and ergonomics for best results.

## 5. Concurrency: Structured Concurrency & Virtual Threads

Structured Concurrency simplifies starting/joining subtasks with clear cancellation rules, while virtual threads enable high concurrency servers with minimal thread management.

## 6. Security & Networking

HTTP/3 (QUIC) support lands in the Java 26 HTTP Client for lower latency, migration-friendly adoption. Final field mutation warnings guide safer reflection usage; PEM encodings improve key handling.



## 7. Spring Framework 6 & Spring Boot 3: The 2026 Baseline

Spring 6 and Boot 3 realigned the platform: Java 17+ baseline, Jakarta EE namespace, first-class AOT, Micrometer Observation, and modern web/data defaults. This release pair is a platform realignment that impacts code, build, and delivery layers; plan upgrades incrementally.

## 8. Spring Boot 4.0 Highlights and Migration Notes

- Boot codebase modularization → smaller, focused JARs.
- Null safety improvements via JSpecify annotations.
- First-class Java 25 support; retains Java 17 compatibility.
- API Versioning and HTTP Service Clients for REST.
- Follow the official migration guide; verify Spring Cloud and Azure starters compatibility.

## 9. Building Cloud■Native Services

- Configuration: externalize via Config Server or environment; use records/sealed types where helpful.
- Observability: Micrometer metrics, tracing (OpenTelemetry), health/readiness; log correlation with virtual threads.
- AOT/Native: Use Spring AOT and GraalVM native■image to cut cold starts; weigh footprint vs. throughput trade■offs.
- Containers: Distroless base images; JVM flags for container awareness; layered JARs/Buildpacks.

## 10. Architecture Patterns

Hexagonal (ports/adapters) keeps domain isolated; DDD aggregates guide transactional boundaries; modular monolith reduces premature distribution; microservices reserved for independently scaling or separately deployable domains.

## 11. Practical Guides & Recipes

- Profiles & Config: application.yml overlays, @ConfigurationProperties binding, profile-specific beans.
- Testing: @SpringBootTest slices, Testcontainers for integration, WireMock/MockMvc for HTTP.
- CI/CD: Gradle builds with reproducible images, SBOMs, and supply chain scanning; rolling updates with health checks.

## 12. Migration Playbook (JDK & Boot)

- Baseline on Java 25 LTS; enable -Xlint and run tests; containerize if not already.
- Audit deprecated APIs; replace security/config properties; update BOMs.
- Upgrade to Spring 6/Boot 3 first (Jakarta), then evaluate Boot 4; test with staging traffic.
- For Java 26 features, enable previews in staging and validate before production.

## 13. Performance Tuning Cheat Sheet

- Startup: Class data sharing (CDS), AOT object caching (Java 26), lazy init in Boot.
- Throughput: G1 ergonomics, vectorize hotspots, keep GC logs; size thread pools conservatively with virtual threads.
- I/O: Prefer HTTP/3 client where available; tune connection pooling; use reactive only when needed.

## 14. Security Hardening Checklist

- Keep to LTS JDK updates; enable TLS modern ciphers; rotate keys; adopt PEM encodings.
- Spring Security defaults; CSRF where applicable; method security; validate JSON via records.
- Supply chain: pin versions, use BOMs, enable SBOMs and scanners; restrict container capabilities.



## 15. References

- InfoWorld: JDK 26 feature overview and schedule
- HappyCoders: Java 26 features with examples
- Java Code Geeks: Future of Java (Valhalla, Panama, Amber, Vector)
- JetBrains: Java Annotated Monthly (Feb 2026)
- InfoQ: Java news roundup incl. JDK 26 schedule
- Spring.io blog: Spring Boot 4.0.0 available now
- TheLinuxCode: Spring 6 & Boot 3 practical 2026 guide