# Migration Guide: JDK 11 + Spring Boot 2.7.9 + JBoss 7 → JDK 17 + Spring Boot 3.3.4 + JBoss 8

## 1. JDK Upgrade (11 → 17)

- Set <java.version> in pom.xml to 17  
- Upgrade maven-compiler-plugin to support Java 17  
- Ensure code compiles with JDK 17; fix deprecated or removed APIs

Example pom.xml snippet / code:

<properties>  
 <java.version>17</java.version>  
</properties>  
  
<build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>${java.version}</source>  
 <target>${java.version}</target>  
 </configuration>  
 </plugin>  
 </plugins>  
</build>

* What to change:
* Update <java.version> property in pom.xml to 17.
* Upgrade maven-compiler-plugin to version 3.11.0 or later.
* Set source and target compatibility to 17 in maven-compiler-plugin.
* Rebuild and ensure compilation passes with Java 17.
* What to validate:
* Project compiles without errors.
* No use of deprecated or removed Java APIs.
* Run ‘java -version’ in your build environment confirms JDK 17.

## 2. Spring Boot Upgrade (2.7.9 → 3.3.4)

Major changes:  
- Spring Boot 3.x requires Java 17+  
- Switch all javax.\* dependencies and imports to jakarta.\*  
- Spring Framework 6.x underneath Spring Boot 3.x uses Jakarta EE 9 namespaces  
- Some APIs deprecated or removed—review your code for compatibility

* What to change:
* Change Spring Boot parent version to 3.3.4 or update spring-boot-dependencies BOM.
* Update all Spring Boot starter dependencies to version 3.3.4.
* Remove any incompatible dependencies with Spring Boot 3.x.
* What to validate:
* Application starts up without Spring Boot version conflicts.
* All Spring Boot starter dependencies resolved correctly.
* Run integration tests to confirm Spring context loads.

## 3. Jakarta Namespace Migration: javax → jakarta

Key dependency changes (pom.xml):

|  |  |  |
| --- | --- | --- |
| Old Dependency | New Dependency | Version Example |
| javax.servlet:javax.servlet-api | jakarta.servlet:jakarta.servlet-api | 6.0.0 |
| javax.persistence:javax.persistence-api | jakarta.persistence:jakarta.persistence-api | 3.1.0 |
| javax.validation:validation-api | jakarta.validation:jakarta.validation-api | 3.0.2 |
| javax.ws.rs:javax.ws.rs-api | jakarta.ws.rs:jakarta.ws.rs-api | 3.1.0 |
| javax.xml.bind:jaxb-api | jakarta.xml.bind:jakarta.xml.bind-api | 4.0.0 |
| javax.annotation:javax.annotation-api | jakarta.annotation:jakarta.annotation-api | 2.1.1 |
| javax.el:javax.el-api | jakarta.el:jakarta.el-api | 5.0.0 |

Example pom.xml snippet / code:

<dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-dependencies</artifactId>  
 <version>3.3.4</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 </dependencies>  
</dependencyManagement>  
  
<dependencies>  
 <!-- Servlet API -->  
 <dependency>  
 <groupId>jakarta.servlet</groupId>  
 <artifactId>jakarta.servlet-api</artifactId>  
 <version>6.0.0</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!-- JPA API -->  
 <dependency>  
 <groupId>jakarta.persistence</groupId>  
 <artifactId>jakarta.persistence-api</artifactId>  
 <version>3.1.0</version>  
 </dependency>  
  
 <!-- Validation API -->  
 <dependency>  
 <groupId>jakarta.validation</groupId>  
 <artifactId>jakarta.validation-api</artifactId>  
 <version>3.0.2</version>  
 </dependency>  
  
 <!-- Spring Boot Starter Web -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Spring Boot Starter Data JPA -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
</dependencies>

* What to change:
* Replace all javax.\* dependencies in pom.xml with jakarta.\* equivalents.
* Update imports in all source code from javax.\* to jakarta.\* (can use IDE refactoring tools).
* Update all XML configuration files (web.xml, persistence.xml) namespaces from javax to jakarta.
* What to validate:
* No javax.\* imports remain in compiled classes.
* Application deploys successfully to JBoss 8 without namespace errors.
* Run smoke tests on JPA, Servlet, Validation, and REST layers.

## 4. JAXB and SOAP Dependencies

Dependencies to update for JAXB and SOAP:

|  |  |  |
| --- | --- | --- |
| Old Dependency | New Dependency | Version Example |
| javax.xml.bind:jaxb-api | jakarta.xml.bind:jakarta.xml.bind-api | 4.0.0 |
| com.sun.xml.bind:jaxb-impl (runtime) | org.glassfish.jaxb:jaxb-runtime | 4.0.2 |
| javax.xml.ws:jaxws-api | jakarta.xml.ws:jakarta.xml.ws-api | 3.0.0 |
| com.sun.xml.ws:jaxws-rt (runtime) | com.sun.xml.ws:jaxws-rt | 3.0.3 |
| javax.activation:activation | com.sun.activation:jakarta.activation | 2.0.1 |
| javax.xml.soap:javax.xml.soap-api | jakarta.xml.soap:jakarta.xml.soap-api | 2.0.0 |
| javax.annotation:javax.annotation-api | jakarta.annotation:jakarta.annotation-api | 2.1.1 |

* What to change:
* Replace javax.xml.bind:jaxb-api with jakarta.xml.bind:jakarta.xml.bind-api.
* Add or update JAXB runtime implementation to org.glassfish.jaxb:jaxb-runtime 4.x.
* Replace javax.xml.ws:jaxws-api with jakarta.xml.ws:jakarta.xml.ws-api.
* Add or update JAX-WS runtime dependencies as needed.
* What to validate:
* SOAP and JAXB serialization/deserialization features work correctly.
* No JAXB or SOAP related runtime errors due to missing classes.
* Run any web service integration tests.

## 5. JBoss Upgrade (7 → 8)

- JBoss EAP 8 supports Jakarta EE 9+ APIs (jakarta.\* namespace)  
- Update server modules and deployment descriptors to use jakarta namespaces  
- Ensure Java 17 runtime environment on JBoss 8  
- Review JBoss-specific configs, security domains, and modules for compatibility

* What to change:
* Deploy application to JBoss 8 server configured with Java 17.
* Update JBoss deployment descriptors and modules for Jakarta namespaces.
* Adjust any JBoss-specific configurations for security and datasources.
* What to validate:
* Application deploys and starts successfully on JBoss 8.
* No classloading or namespace conflicts during runtime.
* Run full system integration and load tests.

## 6. Codebase Changes

- Replace all imports:  
 - javax.\* → jakarta.\*  
- Update XML files (web.xml, persistence.xml) to use jakarta namespaces  
- Refactor Spring Security or other frameworks code using javax imports  
- Verify custom filters, listeners, and servlets using Jakarta APIs

* What to change:
* Refactor all Java imports from javax.\* to jakarta.\*
* Update XML config namespaces from javax to jakarta
* Review and update Spring Security and custom servlet code
* What to validate:
* No javax imports remain anywhere in the source
* Application context loads and runs without errors
* Web and persistence layers tested thoroughly

## 7. Testing Dependencies Update

Recommended versions compatible with Java 17 and Spring Boot 3.x:

|  |  |  |
| --- | --- | --- |
| Library | Recommended Version | Notes |
| JUnit Jupiter (JUnit 5) | 5.10.x | Use JUnit 5; Spring Boot 3 uses it |
| Mockito Core | 5.5.0+ | Latest version for Java 17 support |
| Mockito Inline | 5.5.0+ | For mocking final classes/methods |
| Spring Boot Starter Test | 3.3.4 | Brings in updated testing dependencies |
| AssertJ | 3.24.2+ | Fluent assertions |
| Hamcrest | 2.2 | For matcher libraries |

Example pom.xml snippet / code:

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <version>3.3.4</version>  
 <scope>test</scope>  
 <exclusions>  
 <exclusion>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 </exclusion>  
 </exclusions>  
</dependency>  
  
<!-- Explicit Mockito Core -->  
<dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>5.5.0</version>  
 <scope>test</scope>  
</dependency>  
  
<!-- Mockito Inline for final classes support -->  
<dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-inline</artifactId>  
 <version>5.5.0</version>  
 <scope>test</scope>  
</dependency>  
  
<!-- JUnit Jupiter -->  
<dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
</dependency>  
  
<!-- AssertJ -->  
<dependency>  
 <groupId>org.assertj</groupId>  
 <artifactId>assertj-core</artifactId>  
 <version>3.24.2</version>  
 <scope>test</scope>  
</dependency>  
  
<!-- Hamcrest -->  
<dependency>  
 <groupId>org.hamcrest</groupId>  
 <artifactId>hamcrest</artifactId>  
 <version>2.2</version>  
 <scope>test</scope>  
</dependency>

* What to change:
* Update Spring Boot Starter Test dependency to 3.3.4.
* Add explicit Mockito core and Mockito inline dependencies version 5.5.0+.
* Update JUnit Jupiter version to 5.10.x.
* Update AssertJ and Hamcrest to latest compatible versions.
* What to validate:
* All unit and integration tests run successfully under JUnit 5.
* Mockito works correctly including mocking of final classes.
* CI pipelines pass without test failures.

## 8. Build Plugins Update

- Upgrade maven-surefire-plugin to a version supporting JUnit 5 and Java 17 (e.g., 3.1.2+)  
- Upgrade maven-failsafe-plugin similarly if you use integration tests

Example pom.xml snippet / code:

<plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.1.2</version>  
 <configuration>  
 <useModulePath>false</useModulePath>  
 </configuration>  
</plugin>

* What to change:
* Upgrade maven-surefire-plugin to 3.1.2 or newer.
* Upgrade maven-failsafe-plugin if integration tests are used.
* Configure plugins to support Java 17 and JUnit 5.
* What to validate:
* Build completes successfully with all tests executed.
* No surefire/failsafe plugin errors related to Java version.

## 9. Summary Table

Quick reference of changes needed:

|  |  |  |
| --- | --- | --- |
| Area | Change | Notes |
| Java Version | Update to 17 | maven-compiler-plugin and JDK |
| Spring Boot Version | Upgrade to 3.3.4 | Use BOM for dependency management |
| Jakarta Namespace | Migrate javax.\* → jakarta.\* | Update all dependencies & imports |
| JAXB & SOAP | Update JAXB and SOAP dependencies & imports | Use jakarta.xml.bind and jakarta.xml.ws |
| JBoss | Upgrade to JBoss 8 | Update server modules & configs |
| Test Dependencies | Use JUnit 5 (Jupiter), Mockito 5.5+ (core & inline) | Compatible with Java 17 & Spring Boot 3 |
| Build Plugins | Upgrade surefire and failsafe plugins | Support JUnit 5 and Java 17 |