From Spring Boot 2 To Spring Boot 3 with the Help of Apache Tomcat and More

Ivar Grimstad
Jakarta EE Developer Advocate
Eclipse Foundation
### Jakarta EE 10 Web Profile

<table>
<thead>
<tr>
<th>Updated</th>
<th>Not Updated</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication 3.0</td>
<td>Persistence 3.1</td>
<td>RESTful Web Services 3.1</td>
</tr>
<tr>
<td>Concurrency 3.0</td>
<td>Server Pages 3.1</td>
<td>JSON Processing 2.1</td>
</tr>
<tr>
<td>CDI 4.0</td>
<td>WebSocket 2.1</td>
<td>JSON Binding 3.0</td>
</tr>
<tr>
<td>Expression Language 5.0</td>
<td>Bean Validation 3.0</td>
<td>Annotations 2.1</td>
</tr>
<tr>
<td>Faces 4.0</td>
<td>Debugging Support 2.0</td>
<td>Interceptors 2.1</td>
</tr>
<tr>
<td>Security 3.0</td>
<td>Enterprise Beans Lite 4.0</td>
<td>Dependency Injection 2.0</td>
</tr>
<tr>
<td>Servlet 6.0</td>
<td>Managed Beans 2.0</td>
<td>CDI Lite 4.0</td>
</tr>
<tr>
<td>Standard Tag Libraries 3.0</td>
<td>Transactions 2.0</td>
<td></td>
</tr>
</tbody>
</table>
Jakarta EE 10 Core Profile

- RESTful Web Services 3.1
- JSON Processing 2.1
- JSON Binding 3.0
- Annotations 2.1
- Interceptors 2.1
- Dependency Injection 2.0
- CDI Lite 4.0

Updated
Not Updated
New
Jakarta EE 10

API source level: Java SE 11
API binary level: Java SE 11
TCK run with: Java SE 11+17

App Developers (YOU) can use Java SE 17 features if you like!!
How is this Relevant to spring?
javax.* -> jakarta.*
javax.* -> jakarta.*
Persistence 3.1
Hibernate ORM
UUID as Basic Java Type

@Entity
public class Item {

    @Id
    @GeneratedValue(strategy=GenerationType.UUID)
    private java.util.UUID id;
    private String description;

    ...

}
Hibernate Validator
Scenario 1

New Spring App
Spring Boot 3
+ Tomcat 10

Spring Boot 3
- Tomcat 9
Scenario 2

Upgrade Spring App
Spring Boot 2  →  Spring Boot 3
javax.* -> jakarta.*
IDE
Package and Class Migration

Select migration map: **Java EE to Jakarta EE**

Replaces javax.* imports with jakarta.* imports in Java code.
Does not affect usages of: javax.sql, javax.swing, javax.xml, javax.naming and javax.transaction.xa.

You need to add Jakarta EE libraries before the migration.
1. Update **Spring** version in pom.xml
2. Update **Jakarta EE** version in pom.xml
3. Fix the imports

https://github.com/ivargrimstad/jakartaeese-spring
Scenario 3

Upgrade Spring App with Dependencies
Spring Boot 2  →  Spring Boot 3
1. Update Spring version in pom.xml
2. Update Jakarta EE version in pom.xml
3. Fix the imports
4. Repeat for Dependencies

https://github.com/ivargrimstad/jakartaee-spring
Scenario 4

Upgrade Spring App with External Dependencies
Transformation
Spring Boot 2

Spring Boot 3
1. Update Spring version in pom.xml
2. Update Jakarta EE version in pom.xml
3. Fix the imports
4. Repeat for Dependencies
5. Transform Libraries if necessary

https://github.com/ivargrimstad/jakartaeese-spring
Additional Migration Steps
1. Update Spring version in pom.xml
2. Update Jakarta EE version in pom.xml
3. Fix the imports
4. Repeat for Dependencies
5. Transform Libraries if necessary
6. Rename properties prefixed with javax
7. Verify data and dynamic content

https://github.com/ivargrimstad/jakartaee-spring
Jakarta EE 11

H1, 2024
Release Cadence

~6-9 Months after the latest LTS of Java
Jakarta Data

Standardizes the repository pattern for data access

https://jakarta.ee/specifications/
Jakarta NoSQL

Standardizes integration with NoSQL databases

https://jakarta.ee/specifications/nosql/
Jakarta RPC

Standardizes gRPC within Jakarta EE

https://jakarta.ee/specifications/rpc/
Jakarta EE 11

API source level: Java SE 21
API binary level: Java SE 21
TCK run with: Java SE 21
Summary
<table>
<thead>
<tr>
<th>Jakarta EE 11 Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization 3.0</td>
</tr>
<tr>
<td>Activation 2.1</td>
</tr>
<tr>
<td>Batch 2.1</td>
</tr>
<tr>
<td>Connectors 2.1</td>
</tr>
<tr>
<td>Mail 2.1</td>
</tr>
<tr>
<td>Messaging 3.1</td>
</tr>
<tr>
<td>Enterprise Beans 4.0</td>
</tr>
<tr>
<td>Authentication 3.1</td>
</tr>
<tr>
<td>Concurrency 3.1</td>
</tr>
<tr>
<td>CDI 4.1</td>
</tr>
<tr>
<td>Expression Language 6.0</td>
</tr>
<tr>
<td>Faces 5.0</td>
</tr>
<tr>
<td>Security 4.0</td>
</tr>
<tr>
<td>Servlet 6.1</td>
</tr>
<tr>
<td>Standard Tag Libraries 3.0</td>
</tr>
<tr>
<td>Persistence 3.2</td>
</tr>
<tr>
<td>Pages 4.0</td>
</tr>
<tr>
<td>WebSocket 2.2</td>
</tr>
<tr>
<td>Validation 3.1</td>
</tr>
<tr>
<td>Debugging Support 2.0</td>
</tr>
<tr>
<td>Validation 3.1</td>
</tr>
<tr>
<td>Dependency Injection 2.0</td>
</tr>
<tr>
<td>CDI Lite 4.1</td>
</tr>
</tbody>
</table>

**Under Development**:
- Data 1.0
- Expression Language 6.0
- Servlet 6.1

**Updated**: Authentication 3.1, Concurrency 3.1, CDI 4.1, Faces 5.0, Security 4.0, Servlet 6.1, Persistence 3.2, Pages 4.0, WebSocket 2.2, Validation 3.1, Debugging Support 2.0, Validation 3.1, Dependency Injection 2.0

**Not Updated**: Authorization 3.0, Activation 2.1, Batch 2.1, Connectors 2.1, Mail 2.1, Messaging 3.1, Enterprise Beans 4.0, Standard Tag Libraries 3.0, Transactions 2.0

**New**: RESTful Web Services 4.0, JSON Processing 2.1, JSON Binding 3.0, Annotations 3.0, Interceptors 2.2, Dependency Injection 2.0
How JAKARTA EE is Relevant to spring
builds on
javax.* -> jakarta.*
1. Update **Spring** version in pom.xml
2. Update **Jakarta EE** version in pom.xml
3. Fix the imports
4. Repeat for Dependencies
5. Transform Libraries if necessary
6. Rename properties prefixed with javax
7. Verify data and dynamic content

https://github.com/ivargrimstad/jakartaeem-spring
Jakarta EE
https://jakarta.ee

Jakartablogs
https://jakartablogs.ee/

Ivar's Hashtag Jakarta EE
https://www.agilejava.eu/category/jakarta-ee/

Demo Code
https://github.com/ivargrimstad/jakartaeex-spring
Jakarta EE Overview

Course available on LinkedIn Learning