

Java EE 6 Support in Spring 3

Reza Rahman

Expert Group Member, Java EE 6 and EJB 3.1

Resin EJB 3.1 Lite Container Lead

Author, EJB 3 in Action

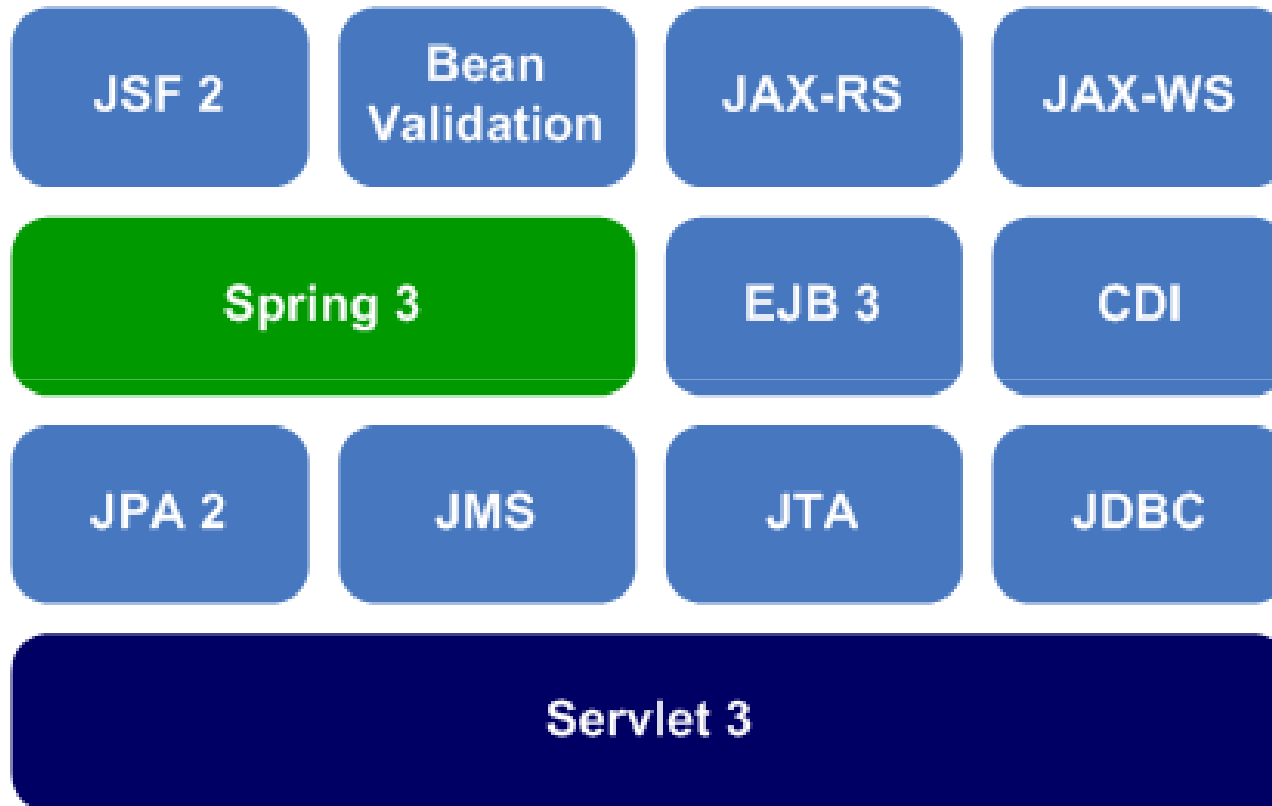
DevIgnition

December, 2010

Java EE 6/Spring 3

- **Significant release, by-and-large well-received**
- **Chopping dead wood, lightweight Web Profile**
- **CDI, JSF 2, EJB 3.1, JPA 2, Servlet 3, JAX-RS, bean validation**
- **Majority of Java EE APIs can be used with Spring**
- **Java EE 6 support major theme of Spring 3 including bean validation, JSF 2, JPA 2, managed beans, JSR 330 and even more EJB 3 annotations**
- **Create powerful hybrid solutions to get the best of both worlds**
- **Even better interoperability in the future via Seam 3**

Spring 3 + Java EE 6



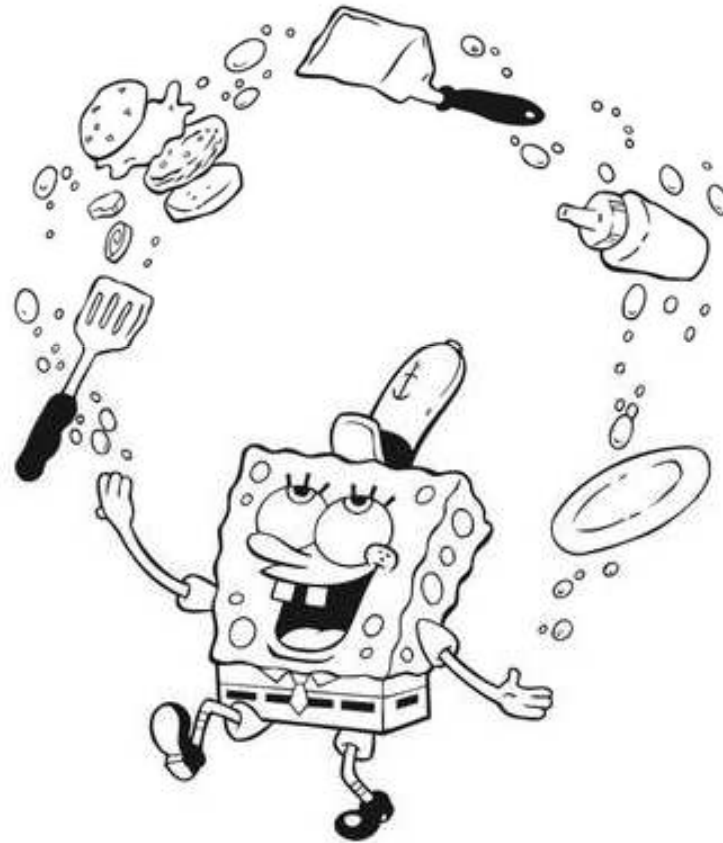
JSR 330, JSR 250, EJB 3 Annotations

- **Dependency Injection for Java (JSR 330) led by Rod Johnson and Bob Lee**
- **JSR 330 Annotations: @Named, @Inject**
- **JSR 330 Adopted by CDI**
- **JSR 250 Annotations: @ManagedBean, @PostConstruct, @PreDestroy, @Resource**
- **JSR 250 Used Across Java EE 6 – CDI, EJB, JSF**
- **EJB Annotations: @EJB, @TransactionAttribute, @Asynchronous, @Schedule**
- **Being more technology agnostic without feature loss**

Spring + JSF

- **JSF widely used in the Spring community, likely to increase adoption with JSF 2 and Facelets**
- **Spring beans can be injected into JSF Managed Beans**
- **Spring beans can be referenced in EL with no JSF Backing Beans**
- **Several resolvers available depending on usage pattern**

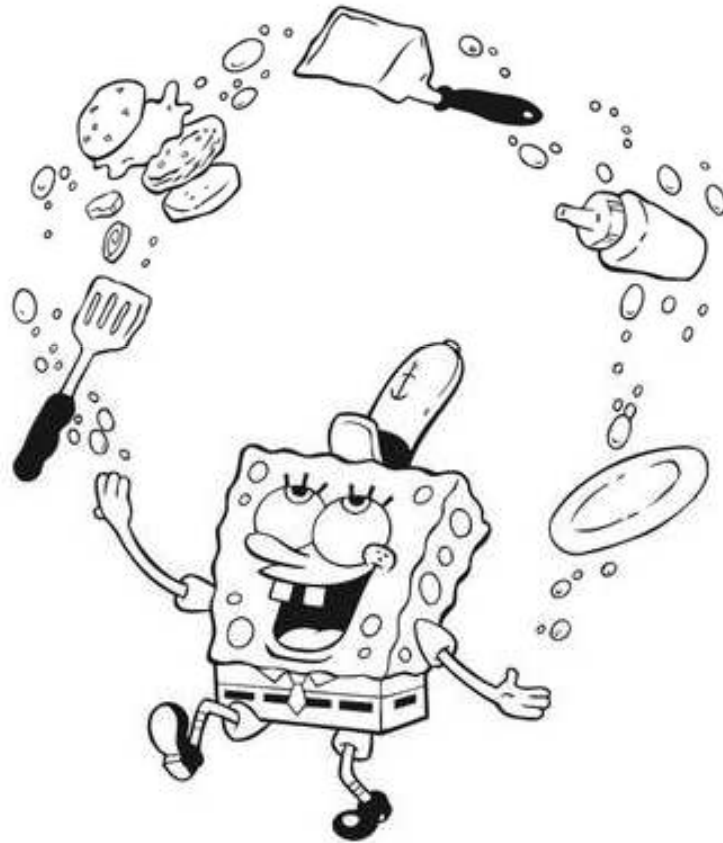
Spring + JSF Demo!



Spring + Bean Validation

- **The Bean Validation API (JSR 303) allows validation constraints to be declared only once and used across application**
- **Usually used through JSF 2 and JPA 2 without any Spring configuration**
- **Spring can be used to bootstrap/inject Validator or use Bean Validation with Spring MVC instead of JSF 2**

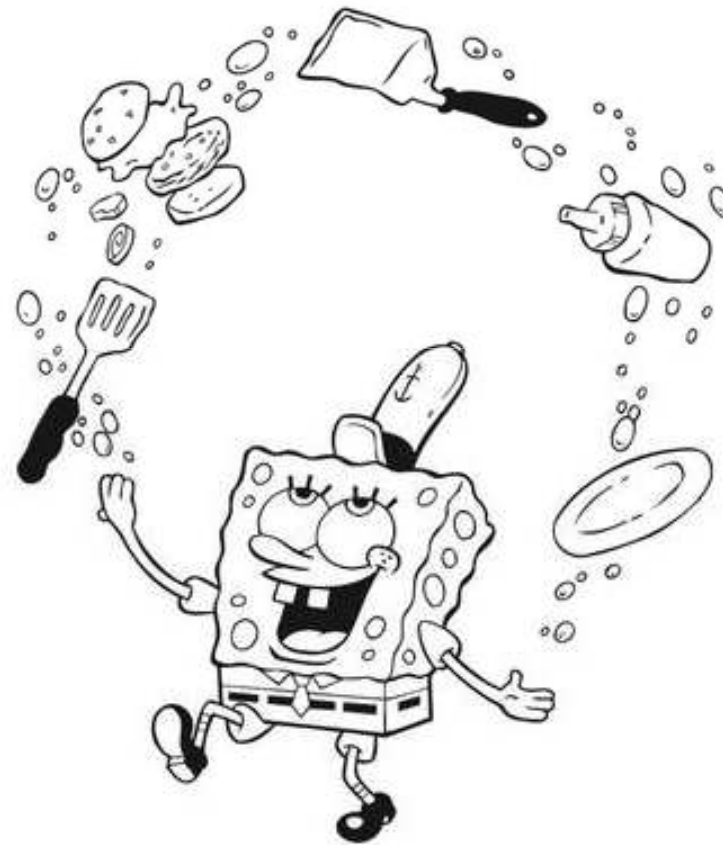
Spring + JSF + JPA + Bean Validation Demo!



Spring + JPA 2

- **JPA is popular with newer Spring projects, JPA 2 is likely to increase adoption further**
- **Several JPA support configurations possible – Java SE bootstrap, Java EE container managed and Spring managed**
- **Most JTA implementations use local transactions by default and transparently upgrade to XA when actually needed**
- **Application server data sources use connection pooling, prepared statement caching, debugging, monitoring and security**

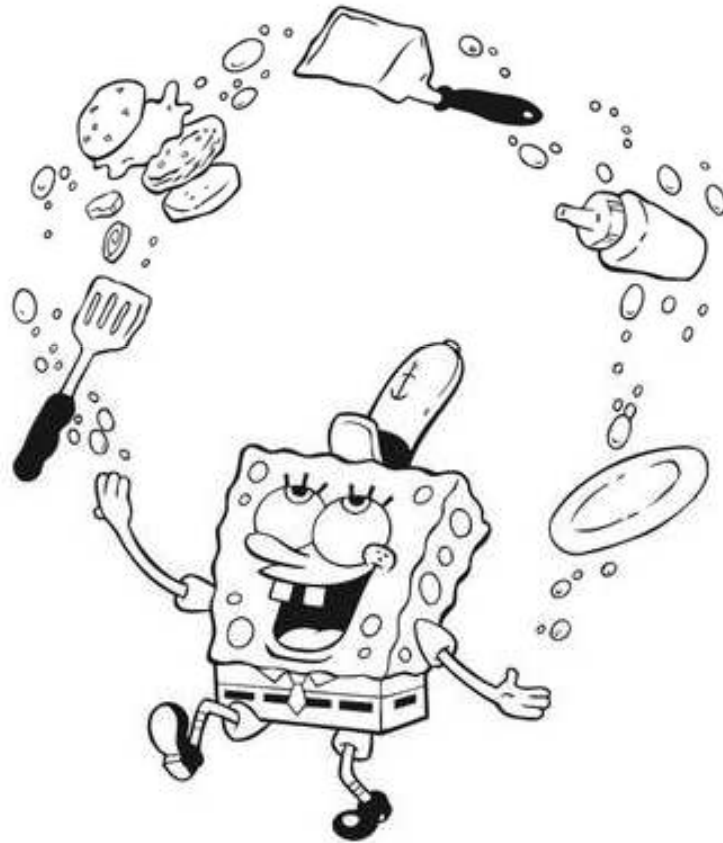
Spring + JPA Demo!



Spring + JAX-RS/JAX-WS

- **Spring has Spring MVC based non-standard REST API**
- **JAX-RS has better adoption in the Spring community**
- **Jersey, RESTEasy and Apache CXF include integration support for Spring**
- **JAX-RS via EJB 3/Spring integration minimizes needless configuration**
- **Spring/JAX-WS integration very similar**
- **Spring has JAX-WS exporters**

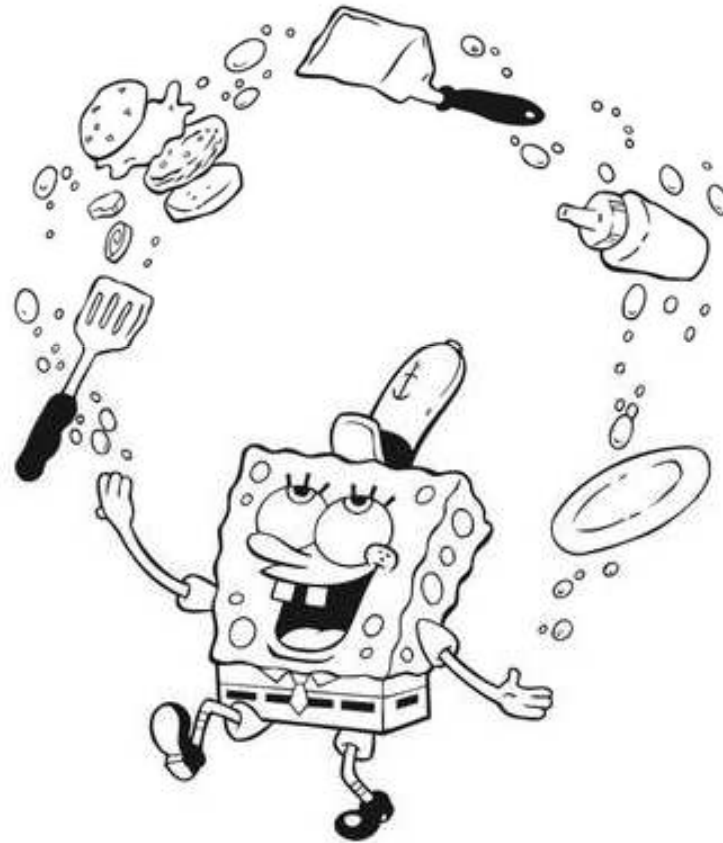
Spring + JAX-RS Demo!



Spring + JMS

- **Spring JmsTemplate can be used on top of raw JMS API for ease-of-use (with some careful considerations)**
- **Spring listeners roughly equivalent to EJB 3 Message Driven Beans, especially JCA rather than JMS listeners**
- **Transactions/XA, acknowledgement, pooling, parallel processing not provided by default!**
- **EJB 3 can be used for better reliability and scalability**

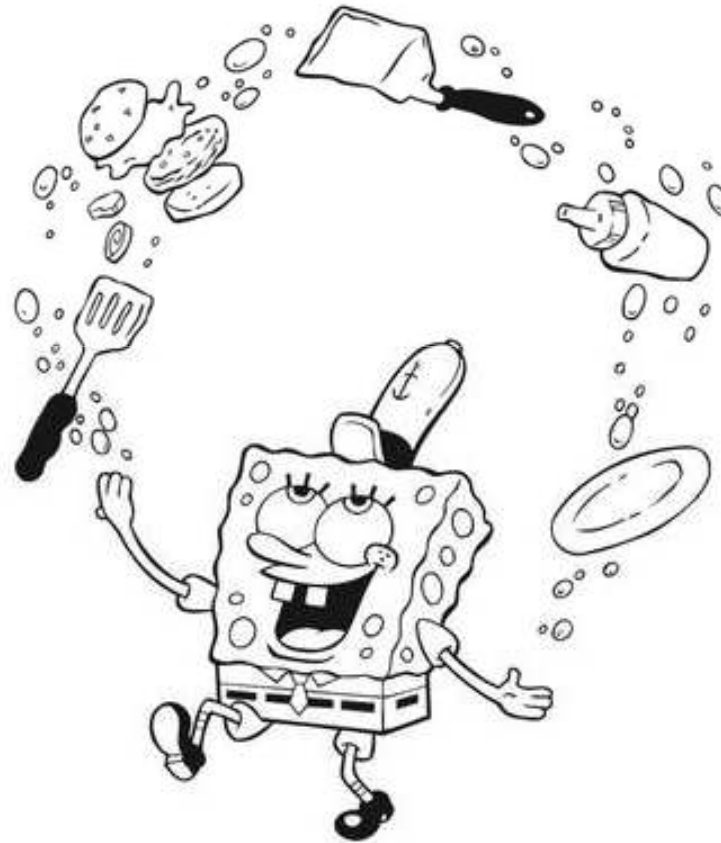
Spring + JMS Demo!



Spring + EJB 3

- **EJB 3 beans transactional, secure, pooled, thread-safe, bandwidth-throttled and closely managed by the application server**
- **EJB 3 can utilize passivation, caching, clustering, load-balancing and remoting for greater scalability**
- **EJB 3 is well-equipped for the service tier of non-trivial applications but is overkill for most other components**
- **Streamlined EJB 3 component model approaches zero-configuration systems with intelligent defaults**
- **Spring 3 supports excellent bi-directional integration with EJB 3**

Spring + EJB 3 Demo!



Spring on Tomcat with Java EE APIs

- **JSF 2, Bean Validation, JPA 2, JAX-RS and JAX-WS can easily be embedded into Tomcat**
- **Pluggable JMS implementations like ActiveMQ and HornetQ can be embedded into Tomcat**
- **CDI/Weld/Seam can be used in Tomcat**
- **Generally requires additional configuration either via Spring and/or Servlet**
- **Spring provides specialized configuration for complex integration cases like JPA**

Spring on Tomcat with Java EE APIs

- **Spring provides some replacement services for transactions, database connectivity, thread-pooling, etc**
- **Embedded EJB 3 containers like OpenEJB can even be integrated into Tomcat to provide EJB 3, JTA/XA, connection/thread/session pooling, caching, monitoring, bandwidth-throttling, etc**
- **Useful not just in Tomcat, but also for testing**

Summary

- **Create a powerful super-stack that utilizes the strengths of both Spring and Java EE**
- **Use Java EE as the runtime for standardization, ease-of-use, administration and scalability while Spring is used for flexibility and third-party integration as well as preserving existing development investment/knowledge-base**
- **Can also use Java EE technologies on Tomcat with Spring**
- **Seam 3 allows for even better Spring integration to utilize CDI features and portable modular extensions**

References

- **Java EE 6 Tutorial:**

<http://www.oracle.com/technetwork/java/javaeec/documentation/tutorials-137605.html>

- **Spring 3 Reference:**

<http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/>

- **Jersey/Spring:**

http://blogs.sun.com/enterprisetechtips/entry/jersey_and_spring

- **RESTEasy/Spring:**

http://docs.jboss.org/resteasy/docs/1.0.1.GA/userguide/html/RESTEasy_Spring_Integration.html

References

- JAX-WS RI/Spring: <http://jax-ws-commons.java.net/spring/>
- Apache CXF: <http://cxf.apache.org>
- Seam 3: <http://seamframework.org/Seam3>
- OpenEJB: <http://openejb.apache.org>
- GlassFish: <http://glassfish.java.net/>
- JBoss: <http://jboss.org/jbossas>
- Resin: <http://www.caucho.com/download/>