

Adam Bien „JavaEE 7 – Complete, Simple, Powerful“

Vgl. <http://www.adam-bien.com/roller/abien/>

Relevante Hinweise aus Adam Biens Präsentation:

JSP wird unterschätzt

JSPs sind einfach zu implementieren und haben gute Performance-Eigenschaften. Vor allem in Kombination mit Expression-Language (EL) sind JSPs sehr mächtig. EL ist mit Java7 ausgebaut worden.

Stellenwert von REST

REST ist die Zukunft von Web-Applikationen und –Schnittstellen. JavaEE 7 bietet mit JAX-RS und Json sehr gute Unterstützung für REST an. SOAP spielt keine Rolle mehr, ist in der Handhabung viel zu schwerfällig.

Vergleich REST – SOAP:

REST -> URI beschreibt Objekte: nachvollziehbare Umsetzung des Domänenmodells

SOAP -> WSDL beschreibt Methoden

Vergleich Java – JavaEE

Bei Enterprise Java geht es um Applikationen innerhalb eines Unternehmens. Grosser Stellenwert wird auf hohe Verfügbarkeit und Monitoring-Infrastruktur gesetzt.

Java ist für Applikationen, welche sich mit einem coolen UI an die Aussenwelt wenden.

Adam Bien sieht eine Tendenz, dass sich die Ansprüche dieser zwei Welten annähern.

Gute JavaEE-Applikationsserver

Biens Präferenzen: Glassfish (JavaEE Referenz-Implementierung), TomEE (Tomcat + JavaEE), JBoss

Funktion von EJBs

EJBs werden vom Applikations-Server erzeugt und gepoolt. Dies ergibt einen Performance-Vorteil, weil die EJB (mit dem ganzen Objektbau, welche sie referenziert) schon existiert.

JMS (Java Message Service)

JMS ist ein API. Der Vorteil von JMS ist, dass „deliver once and only once“ sichergestellt ist (weil JMS eine *persistent queue* aufweist). Eine Alternative zu JMS sind Webservices.

Unit-Test mit Mockito

Unit-Tests setzen mit Vorteil Mockito ein, weil auf diese Weise alle Abhängigkeiten zu einem externen Kontext gemockt werden kann. Auf diese Weise testet der Unit-Test ausschliesslich den Code der unterliegenden Klasse.

Bien macht Unit-Tests nur für Klassen, welche eine Cyclomatic Complexity (CCN) > 7 haben.

Integration-Test mit Jenkins

Integrations-Tests können am besten mit Hilfe von Jenkins gemacht werden. Wichtig ist, dass auch bei Integrations-Test rasch eine Rückmeldung kommt, ob der Test (die Integration) korrekt ist. Muss lange auf eine solche Rückmeldung gewartet werden, so zögern die Entwickler immer länger, ihren Code in die Versionsverwaltung einzubuchen und einen Integrations-Tests auszulösen. Das Test-Framework muss also rasch starten können, damit die Testsuite mit den Integrations-Tests zu schnellen Rückmeldungen führen kann.