



GAME OF JAVA EE

Mit Java Enterprise Edition Version 8 zeichnen sich große Veränderungen für die Zukunft des Industriestandards ab.

| von **ALEXANDER SALVANOS**

Der Philosoph Georg Wilhelm Friedrich Hegel sagte: „Es ist übrigens nicht schwer, zu sehen, dass unsere Zeit eine Zeit der Geburt und des Übergangs zu einer neuen Periode ist.“ Unter diesem Aspekt stellen wir in diesem Artikel die wichtigsten Neuerungen der lange erwarteten und am 18. September 2017 endlich erschienenen Version der Java Enterprise Edition 8 (Java EE 8) vor und geben einen kurzen Ausblick in die Zukunft von Java EE.

Es entspricht der Natur einer Behördenanwendung, dass sie – ebenso wie eine Bankanwendung – besondere Anforderungen bezüglich Sicherheit und Hochverfügbarkeit erfüllen muss. Im Fachjargon spricht man von „geschäftskritischer Unternehmensanwendung“. Mit der Java Enterprise Edition (Java EE) soll diesen besonderen Anforderungen Rechnung getragen werden. Dabei versucht der Standard stets, technologische Neuerungen auf einen gemeinsamen und stabilen Nenner zu bringen. So kommt es, dass mit jeder Version neue und wertvolle Features in Form neuer Java Specification Requests (JSR) hinzukommen.

Das Herzstück der Java-EE-Systemlandschaft ist der Java-EE-Server, der die Implementierung von Sicherheit und Hochverfügbarkeit über zwei spezielle Java-EE-Container verwirklicht: den Web-Container und den EJB-Container. Hersteller, die bei ihren Java-EE-Servern beide Container und alle Dienste der Java-EE-Spezifikation fehlerfrei anbieten, sind berechtigt, ihren Server als vollständig Java-EE-konform zertifizieren zu lassen, so wie zum Beispiel Oracle WebLogic, GlassFish oder IBM WebSphere.

BLICK ZURÜCK ...

Da die Verwendung beider Container in früheren Versionen als schwergewichtig galt, kam neben den vollständig Java-EE-konformen Servern häufig auch der Apache Tomcat zum Einsatz – obwohl er nur den Web-Container enthielt und lediglich einen Bruchteil der Java EE bereitstellen konnte. Open-Source-Frameworks, die man Java-EE-Anwendungen hinzufügte, sollten dieses Problem beheben. Manche dieser Open-Source-Technologien erwiesen sich als so genial, dass sie die Java-EE-Technologien an Finesse sogar übertrafen. Schnell erkannte die Java Community, dass man sie in den Java-EE-Standard aufnehmen musste. Die findigsten Open-Source-Erfinder, wie Gavin King (Hibernate) und Rod Johnson (SpringSource), nahmen führende Rollen in der Java-EE-Expert-Group ein und konnten dort ihren Ideenreichtum einbringen. Neue Technologien, wie beispielsweise Java Persistence API (JPA) und Contexts and Dependency Injection (CDI), entstanden; der Java-EE-Standard wurde immer weiter vereinfacht. Spätestens seit der Version 7 gilt er nun als „leichtgewichtige Alternative“. Die Empfehlung des weltberühmten Expert-Group-Members Adam Bien lautet daher, den vollständig-Java EE konformen Server vorzuziehen und auf den Einsatz von Open-Source-Frameworks zu verzichten.

... IN DIE ZUKUNFT

Diesen Weg wollte man mit Java EE 8 weiter beschreiten. Gleichzeitig sollten moderne Trends, wie beispielsweise Microservices oder cloudbasierte Anwendungen, optimal unterstützt werden. Ebenfalls sollten die bereits vorhandenen APIs noch weiter vereinfacht und mit der Möglichkeit von Java-SE-8-Funktionalität angereichert werden.

Doch es gab Probleme: Da der Java-Community-Process (JCP) für die Fertigstellung der aktualisierten Spezifikationen einen sehr langen Zyklus vorsieht, wurde der Java-EE-Standard in einer sich immer schneller verändernden IT-Welt den Innovationen immer weniger gerecht. Seit 2016 wurde der Java-EE-Prozess immer wieder für seine Unbeweglichkeit kritisiert. Auch der Java-EE-Expert-Group entging nicht, dass Java-EE-Clients immer mehr auf JavaScript-Technologien setzten. Die Mitglieder wollten gegensteuern. Gleichzeitig kam das Gerücht auf, die Oracle Corporation wolle sich als Schirmherr von Java EE zurückziehen. Das wäre ein empfindlicher Schlag, denn Oracle ist als Lizenzinhaber an weit über 80 Java Specification Requests (JSR) beteiligt. Außerdem werden die wichtigsten JSR von Oracle-Experten als Specification Leads gestaltet,

DIE WICHTIGSTEN NEUERUNGEN VON JAVA EE 8

In folgenden Neuerungen sieht die Java-EE-Expert-Group besondere Key-Features.

Key-Feature 1 – Servlets 4.0

Diese neue Servlet-Version verwendet erstmalig das HTTP/2-Protokoll. HTTP/2 beschleunigt die Kommunikation zwischen Client und Server durch Komprimierung und Parallelität, bekannte Hürden des altbewährten HTTP/1.1-Protokolls werden überwunden. Neben dem Einbau von HTTP/2 bietet die neue Servlet-Technologie eine spezielle Mapping-API, über die sich die Pfade zu einzelnen Elementen komfortabel ermitteln lassen.

Key-Feature 2 – JSON-P 1.1

JSON-P 1.1 unterstützt die neuen IETF-Standards JSON-Pointer, JSON-Patch und JSON-Merge-Patch. Durch hinzugefügte spezielle Hilfsklassen können Java-SE-8-Streams verwendet werden.

Key-Feature 3 – JSON-B 1.0

Über JSON-B 1.0 lassen sich einfache Java-Klassen (sogenannte POJO) zu JSON-Dokumenten wandeln, indem lediglich eine einzige Methode aufgerufen werden muss. Genauso einfach ist die Umwandlung von einem JSON-Dokument zu einem POJO.

Key-Feature 4 – JAX-RS 2.1

JAX-RS 2.1 unterstützt Server-sent-Events, womit Daten vom Server zum Client geschickt werden können. Eine neue Reactive API ermöglicht ferner, dass mit ReactiveX Frameworks wie RxJava asynchron und ereignisbasiert interagiert werden kann.

Key-Feature 5 – Security API 1.0

Die Security API 1.0 stellt eine vereinfachte Benutzerschnittstelle für die Authentifizierung und Autorisierung zur Verfügung, die sich auch für die Verwendung mit Clouds und Paas-basierten Anwendungen eignet.

Für zusätzliche Verunsicherung innerhalb der Java Community sorgte außerdem, dass die Weiterentwicklung an den neuen Technologien für längere Zeit ruhte und die Oracle-Experten nichts mehr über zukünftige Vorhaben kommunizierten. Dazu kam, dass mit Cameron Purdy, Reza Rahman, Mark Heckler und John Clingan führende Köpfe die Java-EE-Expert-Group verließen. Um die Java-EE-Entwicklung wieder anzutreiben, gründete Reza Rahman die Java-EE-Guardians.

Das Blatt wendete sich im Juli 2016 – Oracles PR-Manager Mike Moeller kündigte einen Kurswechsel an. In einer öffentlichen Wahl sollte die Java-Gemeinde über die Inhalte von Java-EE-8 mitentscheiden. Der Wahlergebnis führte dazu, dass die Java-EE-Management-API 2.0, der Java-Message-Service 2.1 und MVC 1.0 aus der Gesamtliste der Java-EE-Spezifikation entfernt wurden. Die neue Gesamtliste¹ enthielt 32 Technologien, wovon neun Technologien aktualisiert und zwei komplett neu erstellt werden sollten. Oracle verschob die Fertigstellung der Java-EE-8-Spezifikation auf die zweite Jahreshälfte 2017 und versprach, sich mit voller Kraft der Vollendung zu widmen.

„CONSIDER IT DONE!“

Am 18. September 2017 war es dann so weit – fristgerecht wurden die festgelegten Funktionalitäten sowohl in Papierform als auch in Gestalt des Referenz-Servers GlassFish 5 publiziert.

Doch noch vor der Veröffentlichung von Java-EE-8 gab Oracle bekannt, dass Java EE zukünftig durch die Eclipse Foundation unter dem neuen Namen Eclipse Enterprise for Java (EE4J) weiter-

entwickelt würde. Nachdem die Java-Gemeinde Oracle bereits in den letzten Jahren dafür kritisiert hatte, sie nicht genügend in die Weiterentwicklung eingebunden zu haben, reagierte der IT-Gigant nun, indem er die Rolle des Specification Leads abgab.

Bedeutet das, dass Oracle hiermit jegliche zukünftige Verantwortung ablehnt? Die Antwort lautet: Jein. Oracle wird nach eigener Aussage Java EE innerhalb seines eigenen Java-EE-Servers WebLogic weiterhin unterstützen. Ferner sollen sich Oracles Java-EE-Spezialisten bei der Reifung der Java-EE-Technologien einbringen. Kurzum: Oracle will nicht mehr und stellt diese Entscheidung als Fortschritt dar. Viele Java-EE-Entwickler haben nun die Sorge, dass die über Spenden geförderte Eclipse Foundation überfordert sein wird. Mittlerweile haben sich die Mitglieder der Java-EE-Expert-Group unter dem neuen Schirmherrn zusammengefunden.

Technisch gesehen wird die erste Version von EE4J der jetzigen Java-EE-8-Version entsprechen. Für Java-EE-Entwickler ist es allerdings schon jetzt kritisch, dass sich der Java-EE-8-Referenzserver GlassFish 5 nicht mit der Entwicklungsumgebung Eclipse steuern lässt. Denn die Weiterentwicklung des unterstützenden Plug-Ins (GlassFish-Tools) liegt noch im Hause Oracle. Und die Arbeiten an diesem Plug-in können bei der Eclipse Foundation erst beginnen, nachdem die hierzu erforderlichen Verträge mit Oracle unterzeichnet sind. Bei den verantwortlichen Software-Architekten in den Projekten dürften solche Verhältnisse zur Missstimmung führen.

Der Leiter der Eclipse Foundation Mike Milinkovich steht der Entwicklung des Plug-ins allerdings positiv gegenüber und kommentiert sie mit „Consider it done!“. Ferner ist zu erwarten, dass die sich neu zusammengefundenen Expert-Group-Mitglieder nun beweglicher und freier handeln können. Denn Entscheidungen können jetzt frei von politischen Strategien eines übermächtigen IT-Konzerns fallen. Außerdem entspricht es dem ursprünglichen Grundgedanken von Sun Microsystems und der Java-Community, dass JavaEE beziehungsweise EE4J von der Open-Source-Gemeinde vorangetrieben wird. ●

LESETIPP

Professionell entwickeln mit Java-EE-8: Das umfassende Handbuch von Alexander Salvanos

Mit diesem praxisorientierten Lehrbuch zu Java-EE-8 gibt der langjährige und erfolgreiche Autor – selbst Mitglied der Java-EE-Expert Group (JSON-B 1.0) – seinen Lesern ein Standardwerk an die Hand, das sie auf dem Weg zum Enterprise-Entwickler begleitet.

Neben einem Überblick über Java-EE-8 und der Software-Architektur geht der Autor unter anderem auf Konfiguration und Monitoring eines Java-EE-8-Servers, Planung und Entwurf mit Scrum und AMDD/XP, die Oracle-Datenbank, Java Persistence API, Web Services mit REST und SOAP, JavaMail und vieles mehr ein. Best Practices für leichtgewichtige Enterprise Applications sind ebenso enthalten wie Standards zu servicebasierten Anwendungen, Persistence API, Dependency Injection, HTML5 und Security.

Erscheinungstermin: 23. März 2018



ANSPRECHPARTNER – ALEXANDER SALVANOS

Senior IT Consultant
Public Sector Solutions Consulting



¹ <http://www.oracle.com/technetwork/java/javaee/tech/index.html>