

Java aktuell

Praxis. Wissen. Networking. Das Magazin für Entwickler

Java trägt Früchte

Neue Technologie

Java-API für Batch-Anwendungen, Seite 8

Entwicklungsstrategien

Software modular aufbauen, Seite 19

Mehr selbst entwickeln, Seite 23

Web-Entwicklung

Das Web als Social Network, Seite 32

Neu: Wicket 6, Seite 40

Java und Oracle

ADF Essential Mobile, Seite 52

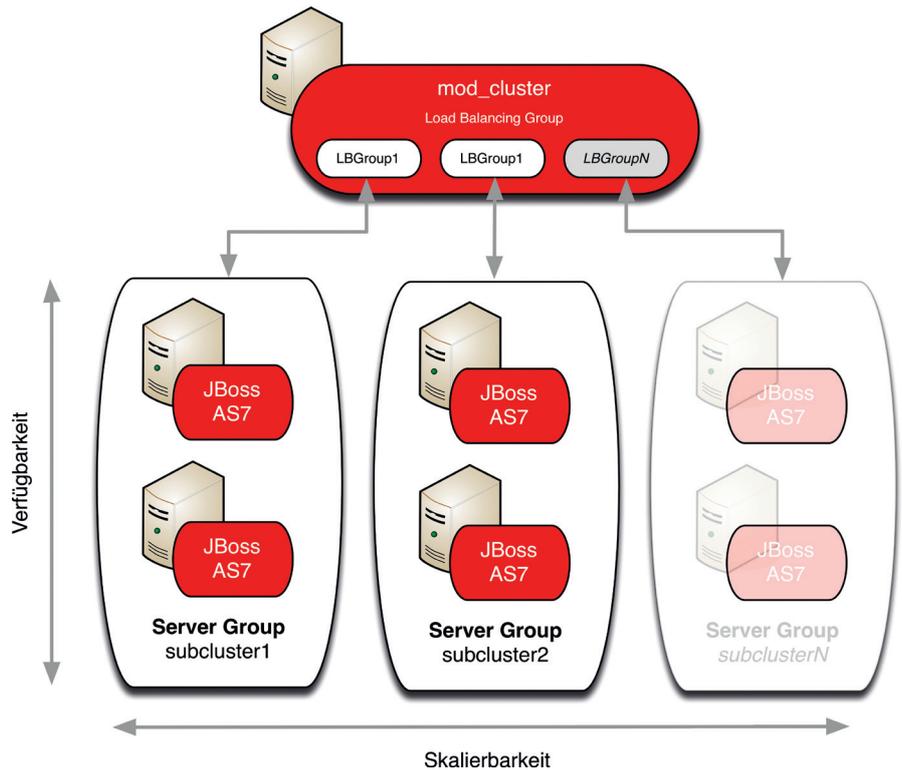


Sonderdruck

D: 4,90 EUR A: 5,60 EUR CH: 9,80 CHF Benelux: 5,80 EUR ISSN 2191-6977



- 3 Editorial
- 5 Das Java-Tagebuch
*Andreas Badelt,
Leiter der DOAG SIG Java*
- 8 Java-API für Batch-Anwendungen
Peter Doschkinow
- 11 Skalierbare Cluster-Topologien mit dem JBoss AS 7
Heinz Wilming und Immanuel Sims
- 15 Lightweight AOP mit CDI und JPA
Prof. Dr. Michael Helbig
- 19 Software modular bauen
Ulf Fildebrandt
- 23 Software – weniger kaufen, mehr bauen
Erik Dörnenburg
- 27 Diagramm-Bibliotheken für Android
Falko Krische
- 32 WebID: Das Web als Social Network
Angelo Veltens
- 37 Vermittlungsinstanzen sind gefragt
Tobias Hartwig
- 40 Anspruchsvolle Web-Anwendungen mit Wicket 6
Jochen Mader
- 44 „Java ist Grundlage für meine Software-Projekte ...“
Interview mit Dirk Dittert
- 45 Nachgefragt ...
Uwe Sauerbrei
- 49 Den Rollläden steuern mit Pi
René Jahn
- 52 ADF Essentials Mobile – Ein unlösbares Problem?
Markus Klenke
- 55 Performance-Falle beim Logging: Die Zeitstempel-Formatierung
Jürgen Lampe
- 61 Unbekannte Kostbarkeiten des SDK Heute: Web-Services
Bernd Müller
- 63 Integration von Java via JNI in traditionelle Cobol-Batch-Anwendungen
Marc Bauer und Martin Kiefer
- 43 Unsere Inserenten
- 66 Impressum



Ein Lastverteiler sorgt dafür, dass Lastverteilung und Failover nur innerhalb eines Subclusters stattfinden. Skalierbare Cluster-Topologien mit dem JBoss AS 7, Seite 11



Wer auf keine Funktion verzichten möchte und schon immer von einer eigenen Rollladensteuerung träumte, findet hier eine Anleitung für den Raspberry Pi, Seite 49

Unbekannte Kostbarkeiten des SDK

Heute: Web-Services

Bernd Müller, Ostfalia

Das Java SDK enthält eine Reihe von Features, die wenig bekannt sind. Wären sie bekannt und würden sie verwendet, könnten Entwickler viel Arbeit und manchmal sogar zusätzliche Frameworks einsparen. Wir stellen in dieser Reihe derartige Features des SDK vor: die unbekanntesten Kostbarkeiten.

Nachdem Gartner und andere Analysten einen Web-Service-Hype für die Dekadenmitte vorhergesagt hatten und dieser auch tatsächlich eintrat, sah sich Sun veranlasst, eine Web-Service-Implementierung in das Java SDK zu integrieren, die wir hier kurz vorstellen wollen.

Web-Services und das SDK

Gartner veröffentlichte im Jahr 2001 eine Analyse, die für die Jahre 2005 und folgende eine starke Verbreitung von Web-Services und die Etablierung serviceorientierter Architekturen prognostizierte [1]. Der entsprechende JSR 224 (Java API for XML-Based Web Services (JAX-WS) 2.0) [2] wurde 2006 verabschiedet. Die Referenz-Implementierung [3] stammte von Sun. Diese wurde in das SDK 6 integriert und kann damit ohne weitere Bibliotheken oder Server wie etwa GlassFish oder JBoss-AS als einfache Web-Service-Implementierung ausschließlich mit dem SDK betrieben werden. Wir gehen auf WSDL, SOAP und weitere Themen nicht ein und beschränken uns auf das unbedingt Nötige.

Das obligatorische „Hello World“

Wir entwickeln im Folgenden die typische Hello-World-Anwendung, um sie als Web-Service zu publizieren. Danach bauen wir einen Client für diesen Service. Die Klasse „HelloWorld“, die nachfolgend wiedergegeben ist, wird mit „@WebService“ annotiert. Diese Annotation ist im JSR 181 [4] definiert und veröffentlicht die Klasse als Web-Service-Endpoint, sodass alle öffentlichen Methoden als Web-Service verfügbar sind. Die Annotation einer Methode mit „@WebMethod“ ist damit nicht notwendig (siehe Listing 1).

```
package de.pdbm;
import javax.ws.WebService;
@WebService
public class HelloWorld {
    public String helloWorld(String whom) {
        return "\"hello world\" from " + whom;
    }
}
```

Listing 1

Um den Web-Service veröffentlichen zu können, ist ein Web-Service-Endpoint erforderlich. Dieser ist durch die Klasse „Endpoint“ realisiert, sodass zur Vervollständigung unseres Beispiels nur ein einziger Aufruf der „publish()“-Methode dieser Klasse genügt, wie Listing 2 zeigt. Damit ist offensichtlich, dass das SDK auch einen einfachen HTTP-Server enthalten muss, um das Binding für das verwendete URI-Schema bereitzustellen.

Wird die Klasse als Main-Klasse gestartet, so liefern HTTP-Get-Aufrufe für „http://localhost:8080/HelloWorld“ und „http://localhost:8080/HelloWorld?wsdl“ allgemeine Informationen zum Endpunkt beziehungsweise das entsprechende WSDL-Dokument zurück. Um den Web-Service tatsächlich deployen und nutzen zu können, müssen noch weitere Klassen generiert werden. Hierzu verwendet man das Werkzeug „ws-gen“, das sich im bin-Verzeichnis des SDK

```
import javax.xml.ws.Endpoint;
public class HelloWorldServer {
    public static void main(String[] args) {
        Endpoint.publish("http://localhost:8080/HelloWorld", new HelloWorld());
    }
}
```

Listing 2

befindet. Wir zitieren dazu die Dokumentation des Werkzeugs: „The ws-gen tool generates JAX-WS portable artifacts used in JAX-WS web services. The tool reads a web service endpoint implementation class (SEI) and generates all the required artifacts for web service deployment, and invocation.“

Zurück zum Beispiel: Der Aufruf von „ws-gen -cp bin -d bin -s generate de.pdbm.HelloWorld“ erzeugt für die Web-Service-Klasse „de.pdbm.HelloWorld“ zwei neue Klassen, deren Quell-Code im Verzeichnis „generate“ und deren Compiledate im Verzeichnis „bin“ abgelegt werden.

Auch für das Erzeugen der benötigten Client-Artefakte enthält das SDK die entsprechende Unterstützung. Das Werkzeug „wsimport“ des SDK erzeugt die für einen Client benötigten Artefakte. Wir zitieren auch hier die Dokumentation: „The wsimport tool generates JAX-WS portable artifacts, such as:

- Service Endpoint Interface (SEI)
- Service
- Exception class mapped from wsdl:fault (if any)
- Async Response Bean derived from response wsdl:message (if any)
- JAXB generated value types (mapped java classes from schema types)“

Der Aufruf erfolgt durch „wsimport -keep -p de.pdbm.client -d bin -s generate http://localhost:8080/HelloWorld?wsd“. Auch hier wird eine Reihe von Klassen erzeugt, die im Verzeichnis „generate“ und hier wiederum im Package „de.pdbm.client“ abgelegt ist. Als Grundlage der Generierung wird die Service-Beschreibung verwendet, die über den HTTP-Server verfügbar ist. Als letzter Schritt kann nun ein Client entwickelt werden, der die generierten Artefakte verwendet. Die Klasse HelloWorldClient leistet genau dies (siehe Listing 3).

Die verwendeten Klassen „HelloWorldService“ und „HelloWorld“ wurden zuvor von „wsimport“ erzeugt. Wir sind damit am Ende unseres einfachen Beispiels angelangt.

Zusammenfassung

Mit Version 6 des SDK wurde eine Implementierung von Web-Services in das Java SE SDK integriert. Damit ist es möglich, einfache Web-Services ohne weitere Bibliotheken und Frameworks zu realisieren. Komplexe Anwendungen auf Basis von

```
package de.pdbm;
import de.pdbm.client.HelloWorldService;
public class HelloWorldClient {
    public static void main(String[] args) {
        HelloWorldService service = new HelloWorldService();
        de.pdbm.client.HelloWorld proxy = service.getHelloWorldPort();
        System.out.println(proxy.helloWorld(„Duke“));
    }
}
```

Listing 3

Web-Services bleiben natürlich nach wie vor umfassenderen Frameworks wie AXIS2 oder Application-Servern vorbehalten.

Literatur

- [1] Gartner: The Hype Is Right: Web Services Will Deliver Immediate Benefits, <http://www.gartner.com/id=344028>
- [2] JSR 224: Java API for XML-Based Web Services (JAX-WS) 2.0, <http://jcp.org/en/jsr/detail?id=224>.
- [3] JAX-WS Reference Implementation: <http://jax-ws.java.net>
- [4] JSR 181: Web Services Metadata for the Java Platform 2.0, <http://jcp.org/en/jsr/detail?id=181>.

Bernd Müller

bernd.mueller@ostfalia.de



Bernd Müller ist Professor für Software-Technik an der Ostfalia. Er ist Autor des Buches „JavaServer Faces 2.0“ und Mitglied in der Expertengruppe des JSR 344 (JSF 2.2).

Die iJUG-Mitglieder auf einen Blick



Java User Group Deutschland e.V.
<http://www.java.de>

Java User Group München (JUGM)
<http://www.jugm.de>

Sun User Group Deutschland e.V.
<http://www.sugd.de>

DOAG Deutsche ORACLE Anwendergruppe e.V.
<http://www.doag.org>

Java User Group Metropolregion Nürnberg
<http://www.source-knights.com>

Swiss Oracle User Group (SOUG)
<http://www.soug.ch>

Java User Group Stuttgart e.V. (JUGS)
<http://www.jugs.de>

Java User Group Ostfalen
<http://www.jug-ostfalen.de>

Berlin Expert Days e.V.
<http://www.bed-con.org>

Java User Group Köln
<http://www.jugcologne.eu>

Java User Group Saxony
<http://www.jugsaxony.org>

Java Student User Group Wien
www.jsug.at

Java User Group Darmstadt
<http://jugda.wordpress.com>

Der iJUG möchte alle Java-Usergroups unter einem Dach vereinen. So können sich alle Java-Usergroups in Deutschland, Österreich und der Schweiz, die sich für den Verbund interessieren und ihm beitreten möchten, gerne unter office@ijug.eu melden.



www.ijug.eu

**JETZT
ABO
BESTELLEN**

Sichern Sie sich 4 Ausgaben für 18 EUR

Für Oracle-Anwender und Interessierte gibt es das Java aktuell Abonnement auch mit zusätzlich sechs Ausgaben im Jahr der Fachzeitschrift DOAG News und vier Ausgaben im Jahr Business News zusammen für 70 EUR. Weitere Informationen unter www.doag.org/shop/

FAXEN SIE DAS AUSGEFÜLLTE FORMULAR AN

0700 11 36 24 39

ODER BESTELLEN SIE ONLINE

go.ijug.eu/go/abo



Interessenverband der Java User Groups e.V.
Tempelhofer Weg 64
12347 Berlin

Java aktuell

+++ AUSFÜLLEN +++ AUSSCHNEIDEN +++ ABSCHICKEN +++ AUSFÜLLEN +++ AUSSCHNEIDEN +++ ABSCHICKEN +++ AUSFÜLLEN

Ja, ich bestelle das Abo Java aktuell – das IJUG-Magazin: 4 Ausgaben zu 18 EUR/Jahr

Ja, ich bestelle den kostenfreien Newsletter: Java aktuell – der IJUG-Newsletter

ANSCHRIFT

Name, Vorname

Firma

Abteilung

Straße, Hausnummer

PLZ, Ort

GGF. ABWEICHENDE RECHNUNGSANSCHRIFT

Straße, Hausnummer

PLZ, Ort

E-Mail

Telefonnummer

X _____

Die allgemeinen Geschäftsbedingungen* erkenne ich an, Datum, Unterschrift

*Allgemeine Geschäftsbedingungen:

Zum Preis von 18 Euro (inkl. MwSt.) pro Kalenderjahr erhalten Sie vier Ausgaben der Zeitschrift "Java aktuell - das IJUG-Magazin" direkt nach Erscheinen per Post zugeschickt. Die Abonnementgebühr wird jeweils im Januar für ein Jahr fällig. Sie erhalten eine entsprechende Rechnung. Abonnementverträge, die während eines Jahres beginnen, werden mit 4,90 Euro (inkl. MwSt.) je volles Quartal berechnet. Das Abonnement verlängert sich automatisch um ein weiteres Jahr, wenn es nicht bis zum 31. Oktober eines Jahres schriftlich gekündigt wird. Die Wiederrufsfrist beträgt 14 Tage ab Vertragserklärung in Textform ohne Angabe von Gründen.

