



mitp

Friedrich
Kiltz

Java Webservices

Friedrich Kiltz

Java Webservice



Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Bei der Herstellung des Werkes haben wir uns zukunftsbewusst für umweltverträgliche und wiederverwertbare Materialien entschieden. Der Inhalt ist auf elementar chlorfreiem Papier gedruckt.

ISBN 978-3-8266-5601-9

1. Auflage 2010

E-Mail: kundenbetreuung@hjr-verlag.de

Telefon: +49 89/2183-7928

Telefax: +49 89/2183-7620

www.mitp.de

© 2010 mitp, eine Marke der Verlagsgruppe Hüthig Jehle Rehm GmbH
Heidelberg, München, Landsberg, Frechen, Hamburg

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Lektorat: Sabine Schulz

Sprachkorrektur: Maren Feilen

Fachkorrektur: Claudia Nölker

Satz: III-satz, Husby, www.drei-satz.de

Druck: Beltz Druckpartner GmbH und Co. KG, Hemsbach

Coverbild: © Dmitry Pistrov – Fotolia.com

Inhaltsverzeichnis

	Einleitung.....	13
Teil A	Bestimmung der Mitte und Stecken der Grenzen	21
I	Überblick Webservices	23
I.1	Was sind Webservices?	23
I.1.1	Warum braucht man Webservices?	23
I.1.2	Wie werden Webservices realisiert?	24
I.1.3	Wer ist an der Konzeption und Nutzung von Webservices beteiligt?	26
I.1.4	Der gemeinsame Nenner	27
I.2	Arten von Webservices	27
I.2.1	Basis-Services	28
I.2.2	Composed-Services	28
I.2.3	Prozess-Services	29
I.3	Aufgaben rund um Webservices	29
I.4	Granularität von Webservices.....	30
I.5	Message Exchange Patterns	30
2	Kontext von Webservices	33
2.1	Überblick SOA.....	33
2.1.1	Die Bedeutung von SOA in diesem Buch	33
2.1.2	Was ist SOA?	33
2.2	Geschäftsprozesse	35
2.3	Webservices und die Container	35
2.4	Einbettung in der Gesamapplikation	36
2.4.1	Webservices als Erweiterung des Zugangs zu eigenen Services	36
2.4.2	Webservices als Unterstützung der eigenen Services.....	39

Teil B Technische Grundlagen	41
3 Technologien für Webservices	43
3.1 XML-Schema	43
3.1.1 Aufgabe des XML-Schemas	43
3.1.2 Beispiel	44
3.1.3 Daten und Aufbau	45
3.1.4 Namensräume	47
3.2 JAXP	48
3.2.1 SAX – Simple API for XML	49
3.2.2 DOM – Document Object Model	51
3.2.3 StAX – Streaming API for XML	52
3.3 JAXB	55
3.4 UDDI	56
4 SOAP	57
4.1 Was ist SOAP?	57
4.2 Aufbau einer SOAP-Nachricht	58
4.3 Beispiel einer SOAP-Message	58
4.4 Transport von Binärdaten	62
4.4.1 Attachments	62
4.4.2 Der Ansatz von MTOM	62
4.5 Vorteile und Nachteile von SOAP	63
5 WSDL – Web Services Description Language	65
5.1 Allgemeiner Aufbau einer WSDL	65
5.2 Beispiel einer WSDL	66
5.3 SOAP-Bindings	72
5.4 Unterschiede zu WSDL 2.0	74
5.5 Code-First – Contract-First	75
5.5.1 Code-First	75
5.5.2 Contract-First	76
5.5.3 Meine Empfehlung	76
6 REST – REpresentational State Transfer	77
6.1 Grundlagen von REST	77
6.1.1 Beschreibung von REST	77
6.1.2 Architekturansatz in REST	79
6.2 JAX-RS	80
6.2.1 Resource Classes	81

6.2.2	REST-Provider	84
6.2.3	Übersicht der Annotations	87
6.2.4	Das Response-Objekt	88
6.2.5	REST-Beispiel	90
6.2.6	WADL	94
6.3	REST-Clients	97
6.3.1	Java-REST-Client	97
6.3.2	PHP-REST-Client	98
6.3.3	AJAX-REST-Client	98
6.4	REST und JSON	99
6.4.1	JSON in Java	99
6.4.2	Vergleich von JSON und XML	101
7	Webservice-Architektur in Java	103
7.1	Verteilte Anwendungen	103
7.1.1	Sockets	103
7.1.2	RMI – Remote Method Invocation	104
7.1.3	EJB	105
7.2	JAX-RPC	106
7.2.1	Beschreibung	106
7.2.2	Anwendungsfälle	107
7.2.3	WSDL/XML to Java Mapping	108
7.3	JAX-WS	112
7.3.1	Von JAX-RPC zu JAX-WS	112
7.3.2	Client- und Service-APIs	113
7.3.3	Annotations	115
7.3.4	SOAP-Bindings	119
Teil C Tools und Frameworks		121
8	Axis2	123
8.1	Vorstellung Axis2	123
8.1.1	Was ist Axis2?	123
8.1.2	Neu in Axis2	124
8.1.3	Die Axis2-Webapplikation	125
8.1.4	WSDL2Java und Java2WSDL	126
8.1.5	Data Binding	129
8.1.6	StAX – Streaming API for XML	130
8.1.7	AXIOM – AXIs Object Model	130

8.2	Architektur von Axis2	133
8.2.1	Flow - Phase - Handler	133
8.2.2	Handler und Module	135
8.2.3	Konfiguration und Deployment eines einfachen Services ..	140
8.2.4	Lebenszyklus eines Services	141
8.2.5	Sessions mit Axis2	142
8.3	Die Client-API	142
8.3.1	Ein Client per Hand	142
8.3.2	Der generierte Stub	144
8.3.3	Nicht blockierende Aufrufe	145
8.4	Axis2 und MTOM	147
8.5	Rampart	150
8.6	Spring-Integration in Axis2	150
8.6.1	SpringServletContextObjectSupplier	151
8.6.2	SpringAppContextAwareObjectSupplier	152
8.7	Axis2-Tools	153
8.7.1	Übersicht	153
8.7.2	Service Archive Wizard	154
8.7.3	Code Generator Wizard	154
8.7.4	TCPMon	155
8.7.5	WSDL-Editor	157
8.8	Anwendungsfälle	157
8.8.1	Installation der axis2.war	157
8.8.2	Ping-Service mit Axis2	159
8.8.3	Komplexerer Service mit Axis2	162
8.8.4	Erzeugen eines Clients mit ADB.	169
8.8.5	Erzeugen eines Dateiuploads mit MTOM	172
8.8.6	Erzeugen eines Handlers	179
8.8.7	Nutzung von Spring in Axis2 mit dem SpringServletContextObjectSupplier	184
8.8.8	Nutzung von Spring in Axis2 mit dem SpringAppContextAwareObjectSupplier	187
8.8.9	Installation und Nutzung von Rampart	191
9	Metro-Framework	199
9.1	Vorstellung	199
9.2	Die Distribution	200
9.2.1	Umfang der Distribution	200
9.2.2	Metro ohne Container	200

9.2.3	Metro mit dem Container	201
9.3	Metro-Tools	202
9.3.1	wsimport	203
9.3.2	wsgen	205
9.4	Handler	207
9.5	Client	209
9.5.1	Dynamischer Service	209
9.5.2	Generierter Stub	209
9.5.3	Asynchroner Client	211
9.6	Sonstige Features	212
9.6.1	Die Datei sun-jaxws.xml	212
9.6.2	MTOM	214
9.6.3	Deklaration von Bindings	214
9.7	Anwendungsfälle	215
9.7.1	Installation und Einstieg	215
9.7.2	Deployment eines einfachen Webservices im Tomcat	220
9.7.3	Erzeugen eines komplexeren Webservices	223
9.7.4	Erzeugen und Einbinden eines Handlers	230
9.7.5	Erzeugen eines asynchronen Clients	234
9.7.6	Übertragung von Binärdaten: MTOM	242
10	Jersey	247
10.1	Die Referenzimplementierung	247
10.1.1	Download	247
10.1.2	Dokumentation	248
10.2	Jersey-Server	248
10.3	Jersey-Client	250
10.4	Spring und Jersey	252
10.5	AJAX, JSON, Jersey und Spring	253
10.6	Anwendungsfälle	255
10.6.1	Installation in einer Webapplikation	255
10.6.2	Einstieg in Jersey mit Spring	259
10.6.3	AJAX, JSON und Jersey	265
11	Spring-WS	275
11.1	Vorstellung	275
11.2	Funktionsweise	276
11.2.1	Endpoints	276
11.2.2	EndpointMapping	277
11.2.3	WSDL-Definition	279

II.2.4	JAX-WS und JAXB	280
II.3	Interceptor	282
II.4	Client mit Spring-WS	283
II.5	Security mit Spring-WS	285
II.6	Marshalling XML	286
II.7	Charakteristik	288
II.8	Anwendungsfälle	288
II.8.1	Kommunikationstest	288
II.8.2	Einbau eines Interceptors	297
II.8.3	Erzeugung eines Clients	300
II.8.4	Komplexeres Beispiel mit Marshalling	304
I2	Apache CXF	313
I2.1	Vorstellung	313
I2.2	Funktionsweise	314
I2.2.1	Der CXF-Bus	314
I2.2.2	CXFServlet und Spring-Konfiguration	314
I2.2.3	Frontends	314
I2.2.4	Contract-First – Code-First	316
I2.3	Interceptor	319
I2.4	Client	321
I2.5	JAX-RS mit CXF	322
I2.6	Charakteristik	322
I2.7	Anwendungsfälle	323
I2.7.1	Installation und Einstieg	323
I2.7.2	Integration eines Interceptors	329
I2.7.3	Erstellung eines komplexeren Beispiels mit Codegenerierung	331
I3	Vergleich der Frameworks	339
I3.1	Technische Daten	339
I3.1.1	Eckpunkte	340
I3.1.2	Standards	340
I3.1.3	WS-*	341
I3.1.4	Data Bindings	341
I3.1.5	Transportprotokolle	342
I3.2	Charakteristik	342
I3.2.1	Axis2	342
I3.2.2	Metro	343
I3.2.3	Spring-WS	344

13.2.4	Apache CXF	344
13.2.5	Jersey	345
13.3	Vergleich der Vorgehensweisen	346
13.3.1	Axis2	346
13.3.2	Metro	347
13.3.3	Spring-WS	348
13.3.4	Apache CXF	348
13.3.5	Jersey	349
13.4	Gleiche Aufgabe für alle	350
13.4.1	Aufgabenbeschreibung	350
13.4.2	Lösung Axis2	352
13.4.3	Lösung Metro	355
13.4.4	Lösung Spring-WS	356
13.4.5	Lösung Apache CXF	358
13.4.6	Lösung Jersey	360
13.4.7	Vergleich der Lösungen	362
13.5	Welches Framework sollten Sie nehmen?	363
13.5.1	Techniken	363
13.5.2	Entscheidungsfindung	363
13.5.3	Fazit	364
Teil D Weitere wichtige Informationen		365

14	Weiterführende Themen	367
14.1	Sicherheit	367
14.1.1	Anforderungen an die Sicherheit	367
14.1.2	Möglichkeiten für sichere Webservices	368
14.2	Transaktionen	371
14.2.1	WS-Transaction	371
14.3	WS-Addressing	371
14.3.1	Überblick	371
14.3.2	Beispiel	372
14.4	WS Reliable Messaging	375
14.4.1	WS-RM	375
14.4.2	WS-RM Policy	376
14.5	Interoperabilität	377
14.5.1	Ursachen für mangelnde Interoperabilität	377

14.5.2	Ansätze, um die unterschiedlichen SOAP-Engines kompatibel zu machen.....	379
14.6	Performance.....	380
14.6.1	Checkliste für Webservice-Performance	380
14.6.2	Performancevergleich	381
Teil E Anhänge		383
A	Abkürzungsverzeichnis.....	385
B	Glossar	387
C	Literaturverzeichnis.....	389
D	Weitere Infos	393
D.1	XML	393
D.1.1	Ursprung und Wesen von XML	393
D.1.2	Aufbau und Syntax einer XML-Datei	393
D.1.3	Wohlgeformtes XML	397
D.2	Maven.....	397
D.2.1	Installation	398
D.2.2	Grundsätzliche Aufgaben	398
D.2.3	Einbinden des Tomcat-Plugins in eine pom.xml.....	399
D.2.4	Eclipse-Plugin für Maven.....	401
D.2.5	Weiterführende Informationen.....	401
D.3	Ajax.....	401
D.3.1	Beschreibung	401
D.3.2	Eventhandler.....	402
D.3.3	DHTML.....	403
D.3.4	Daten vom Server	404
D.3.5	Framework SACK.....	406
E	Annotations	409
E.1	JSR 181 Web Services Metadata for the Java Platform	409
E.2	JSR 222 Java Architecture for XML Binding (JAXB) 2.0	414
E.3	JSR 224 Java API for XML-Based Web Services (JAX-WS) 2.0	416
E.4	JSR 250 Common Annotations for the Java Platform	425
Stichwortverzeichnis		427