

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>15</b>
<b>1 Einleitung</b>	<b>25</b>
1.1 Entstehung	26
1.2 Unterstützte Standards	28
1.3 Was beinhaltet Axis2?	29
1.4 Warum Axis2 einsetzen?	29
1.4.1 Bessere Performance durch StAX und AXIOM	30
1.4.2 Flexiblere Kommunikationsinfrastruktur	30
1.4.3 Einfacheres Deployment von Services	30
1.4.4 Bessere Unterstützung während der Entwicklung	30
1.4.5 Erweitertes Handler-Konzept	31
1.4.6 Bessere Unterstützung von aktuellen Standards	31
1.5 Die Zukunft von Axis2	31
<b>2 Web-Service-Grundlagen</b>	<b>33</b>
2.1 SOAP	33
2.1.1 Nachrichtenformat	34
2.1.2 Verarbeitungsmodell	37
2.1.3 SOAP-Fault	40
2.1.4 Nachrichtenaustausch	42
2.1.5 Protokoll-Binding	45
2.1.6 SOAP 1.2 vs. SOAP 1.1	47
2.2 WSDL	49
2.2.1 MEPs – Message Exchange Patterns	50
2.2.2 WSDL 1.1	52
2.2.3 WSDL 2.0	58
2.3 Code First vs. Contract First	59
2.3.1 Der Code-First-Ansatz	60
2.3.2 Der Contract-First-Ansatz	63
2.3.3 Einsatz von Contract First bei bereits bestehendem Code	66

<b>3</b>	<b>Erste Schritte</b>	<b>69</b>
3.1	Axis2-Distributionen	69
3.2	Installation von Axis2	70
3.2.1	Die Axis2-Webanwendung	71
3.2.2	Standarddistribution	73
3.3	Zentrale Konzepte von Axis2	75
3.3.1	AXIOM	75
3.3.2	Servicearchive	75
3.3.3	Message Receiver	76
3.3.4	Repository	77
3.4	Implementierung einfacher Web Services mit POJOs	78
3.5	Deployment von Services in einem Standalone-Server	85
3.6	Einsatz der Axis2-Webanwendung	87
3.6.1	Deployment von Web Services	87
3.6.2	Serviceadministration	88
3.7	Entwicklung eines Clients für den SimpleHotelService	90
3.7.1	Direkte Verwendung des Client-API von Axis2	90
3.7.2	Entwicklung von Clients mithilfe von Codegenerierung	94
3.8	Geruhssame Nächte mit Axis Hotels	95
<b>4</b>	<b>Entwicklung mit Axis2</b>	<b>97</b>
4.1	Eclipse als Entwicklungsumgebung verwenden	97
4.1.1	Projekteinrichtung	97
4.1.2	Eclipse Web Tools Platform	99
4.2	Axis2 Eclipse Plug-ins	102
4.2.1	Codegenerator-Wizard	102
4.2.2	Service Archiver Wizard	103
4.3	Debugging	105
4.4	Axis2-Quellcode erforschen	106
4.4.1	Den Axis2-Quelltext sichten und browsen	106
4.4.2	Quellcode erforschen – Beispiel	108
4.5	Werkzeuge für den Umgang mit SOAP-Nachrichten	109
4.5.1	Apache TCPMon	110
4.5.2	SOAP erforschen und lernen mit TCPMon	112
4.5.3	SOAPMonitor	113

<b>5</b>	<b>AXIOM</b>	<b>117</b>
5.1	Einführung	117
5.2	StAX	117
5.2.1	Push vs. Pull Parsing	117
5.2.2	StAX-API	119
5.2.3	XML parsen mit StAX	120
5.3	AXIOM	128
5.3.1	AXIOM-Architektur	129
5.3.2	AXIOM-API	132
5.3.3	Caching	142
5.4	Web-Service-Implementierung mit AXIOM	145
<b>6</b>	<b>Client-API</b>	<b>151</b>
6.1	ServiceClient	151
6.2	Aufrufmuster	154
6.2.1	Request-Response mit blockierendem API	156
6.2.2	Request-Response mit nichtblockierendem API über eine Verbindung	157
6.2.3	Request-Response mit nichtblockierendem API über zwei Verbindungen	159
6.2.4	Einwegaufruf	164
6.2.5	Zuverlässiger Einwegaufruf	164
6.3	Clientseitige Konfiguration	165
6.3.1	JavaBean Properties	166
6.3.2	Generische Properties	167
6.3.3	HTTP-Parameter	172
6.4	OperationClient	175
<b>7</b>	<b>Contract First mit Axis2</b>	<b>181</b>
7.1	Codegenerierung	181
7.1.1	Aufruf von WSDL2Java von der Kommandozeile	182
7.1.2	Axis2-Codegenerator-Wizard für Eclipse	192
7.1.3	Ant Task	197
7.2	Implementierung und Deployment von Services	199
7.2.1	Der Ordner „resources“	199
7.2.2	Generierter Code und Implementierung des Service	201
7.2.3	Paketierung und Deployment	205

7.3	Implementierung von ServiceClients	206
7.4	Einwegkommunikation	214
<b>8</b>	<b>Weiterführende Aspekte der Entwicklung</b>	<b>219</b>
8.1	Fehlerbehandlung	219
8.1.1	Definition von Fehlern in XML Schema und WSDL	221
8.1.2	Codegenerierung	224
8.2	Lebenszyklus von Services	228
8.3	Sessionverwaltung	232
8.3.1	Request-Session-Scope	235
8.3.2	SOAP-Session-Scope	236
8.3.3	Transport-Session-Scope	238
8.3.4	Application Scope	238
8.3.5	Sessionverwaltung mit Clientanwendungen	239
8.3.6	Codebeispiel	239
<b>9</b>	<b>Architektur und Konfiguration</b>	<b>245</b>
9.1	Interne Verarbeitung von Nachrichten	246
9.1.1	Flows	246
9.1.2	Phasen	251
9.1.3	Dispatch-Mechanismus	256
9.2	Interne Datenstrukturen: Description und Context	257
9.2.1	Description-Hierarchie	257
9.2.2	Context-Hierarchie	259
9.2.3	Beziehung zwischen Context- und Description-Hierarchien	261
9.2.4	Laden von Konfigurationen	261
9.3	Globale Konfiguration	263
9.3.1	Parameter	264
9.3.2	Message Receiver	266
9.3.3	Transporte	266
9.3.4	Global aktivierte Module und Modulkonfigurationen	266
9.3.5	Standardmäßige Modulversion	266
9.3.6	Phasen	267
9.3.7	Target Resolver	268
9.3.8	Listeners/Observers	268
9.3.9	Deployer	271
9.3.10	Message Builder und Message Formatter	273
9.3.11	Data Locators	274

## Inhaltsverzeichnis

9.3.12	SOAP Role	276
9.3.13	Transaktionen	277
9.4	Konfiguration von Services	277
9.4.1	Services und Servicegruppen	278
9.4.2	Bestimmung der Namen von Services und Servicegruppen	278
9.4.3	WSDL-Dokumente und automatische WSDL-Generierung	279
9.4.4	Elemente der Datei services.xml	280
9.4.5	Serviceparameter	285
9.5	Deployment von Services	288
9.5.1	Axis2-Webanwendung	288
9.5.2	Standalone-Server	288
9.6	Zugriff eines Service auf Context und Konfiguration	289
9.7	Zugriff auf Ressourcen im Servicearchiv	290
9.8	Start von Axis2 mit entferntem Repository	290
<b>10</b>	<b>Handler und Module</b>	<b>293</b>
10.1	Handler	294
10.1.1	Die Schnittstelle Handler	295
10.1.2	Implementierung von Handlern	297
10.1.3	Konfiguration von Handlern	298
10.2	Module	300
10.2.1	Die Schnittstelle Module	301
10.2.2	Konfiguration von Modulen	302
10.2.3	Paketierung und Deployment	308
10.2.4	Engagement	310
10.2.5	Dynamisches Engagement zur Laufzeit	316
10.2.6	Mitgelieferte Module	317
<b>11</b>	<b>Data Binding</b>	<b>319</b>
11.1	Grundlagen des XML Data Binding	320
11.2	Codegenerator-Framework	322
11.3	ADB – Axis Data Binding	329
11.3.1	ADB-Schema-Compiler	329
11.3.2	ADB Integration in Axis2	333
11.3.3	Codegenerierung	334
11.3.4	Wrapped vs. Unwrapped	346

## Inhaltsverzeichnis

11.4	XMLBeans	350
11.5	JiBX	357
11.6	JAXB RI	363
11.7	Zusammenfassung	367
<b>12</b>	<b>Message Receiver und ServiceObjectSupplier</b>	<b>369</b>
12.1	Einführung	369
12.1.1	Blick zurück: Provider in Axis 1.x	369
12.1.2	Blick in die Gegenwart: Reise durch die Axis2 Engine	370
12.2	Nachrichtenempfänger	371
12.2.1	Contract First	372
12.2.2	Message Receiver von innen	372
12.3	Axis2 und Groovy	375
12.4	Message Receiver und WSDL	381
12.5	Enterprise JavaBeans und Axis2	384
12.5.1	Einführung	384
12.5.2	Möglichkeiten zur Integration von EJBs	386
12.5.3	Der Bankleitzahlenservice als EJB	386
12.5.4	Die Realisierung von EJBMessageReceiver	391
12.5.5	EJB als Web Service bereitstellen	393
12.6	ServiceObjectSupplier	395
12.7	Spring Framework	397
12.7.1	Einführung	397
12.7.2	Axis2 und das Spring Framework	398
12.7.3	Der Bankleitzahlenservice als Spring Bean	400
12.7.4	SpringServletContextObjectSupplier	405
12.7.5	Erforderliche Spring-Bibliotheken	408
12.7.6	SpringApplicationContextAwareObjectSupplier	408
12.8	Die EJBUtil-Implementierung	412
<b>13</b>	<b>MTOM &amp; SwA</b>	<b>421</b>
13.1	Base64 & SwA	421
13.1.1	Base64	422
13.1.2	SwA	424

## Inhaltsverzeichnis

13.2	XOP & MTOM	426
13.2.1	XOP	427
13.2.2	MTOM	430
13.2.3	SwA vs. MTOM	431
13.3	MTOM in Axis2	432
13.3.1	OMText	432
13.3.2	MTOM Web Service mit AXIOM-API	433
13.3.3	MTOM mit Data Binding	445
13.4	SwA in Axis2	454
13.5	Attachment Caching	458
<b>14</b>	<b>Transportprotokolle</b>	<b>461</b>
14.1	Transportmechanismus	462
14.1.1	TransportListener	462
14.1.2	TransportSender	464
14.2	Aktivierung von Transportprotokollen auf Serviceebene	465
14.3	HTTP	466
14.3.1	Transport Receiver für Standalone-Modus	466
14.3.2	SimpleHttpServer in eigene Applikationen einbetten	468
14.3.3	CommonsHTTPTransportSender	469
14.4	JMS	473
14.4.1	Installation von ActiveMQ	473
14.4.2	Services mit JMS-Kommunikation	473
14.4.3	Clientanwendungen mit JMS-Kommunikation	479
14.5	TCP	480
<b>15</b>	<b>JAX-WS</b>	<b>483</b>
15.1	JAX-WS-Programmiermodell	483
15.1.1	Code-First-Entwicklung	484
15.1.2	Contract-First-Entwicklung	490
15.2	Servicecliententwicklung mit JAX-WS	494
15.2.1	Synchroner Serviceclient	494
15.2.2	Asynchroner Serviceclient	499

## Inhaltsverzeichnis

15.3	Weiterführende Themen	504
15.3.1	Low-Level-API	504
15.3.2	MTOM	508
15.3.3	Handler	512
15.4	JAX-WS-Implementierung	516
<b>16</b>	<b>Alternative Nachrichtenformate</b>	<b>523</b>
16.1	Message Formatter und Message Builder	523
16.2	JSON	526
16.2.1	Badgerfish- und Mapped-Konvention	526
16.2.2	JSON in Axis2	528
16.2.3	Fast Infoset	532
16.2.4	Konzept	532
16.2.5	Performance	536
16.2.6	Fast Infoset in Axis2	538
<b>17</b>	<b>WS-*-Standards mit Axis2</b>	<b>541</b>
17.1	WS-Addressing	542
17.1.1	Grundlagen	542
17.1.2	WS-Addressing mit Axis2	544
17.2	WS-Policy	549
17.2.1	Grundlagen	549
17.2.2	Apache Neethi: WS-Policy mit Axis2	551
17.3	WS-Security	554
17.3.1	Grundlagen	554
17.3.2	Rampart: WS-Security mit Axis2	558
17.3.3	Konfiguration mit WS-Policy	599
17.4	WS-ReliableMessaging	603
17.4.1	Grundlagen	603
17.4.2	Sandesha2: WS-ReliableMessaging mit Axis2	608
<b>A</b>	<b>XML Schema und WSDL von Axis Hotels</b>	<b>623</b>
<b>B</b>	<b>WSDL2Java</b>	<b>631</b>
B.1	Kommandozeile	631
B.2	Ant-Task	631
B.3	Maven-Plug-in	631



## Inhaltsverzeichnis

<b>C</b>	<b>Java2WSDL</b>	<b>635</b>
C.1	Kommandozeile	635
C.2	Maven-Plug-in	636
<b>D</b>	<b>Maven 2 AAR Plug-in</b>	<b>639</b>
<b>E</b>	<b>Maven 2 MAR Plug-in</b>	<b>641</b>
	<b>Stichwortverzeichnis</b>	<b>643</b>
	<b>Über die Autoren</b>	<b>649</b>