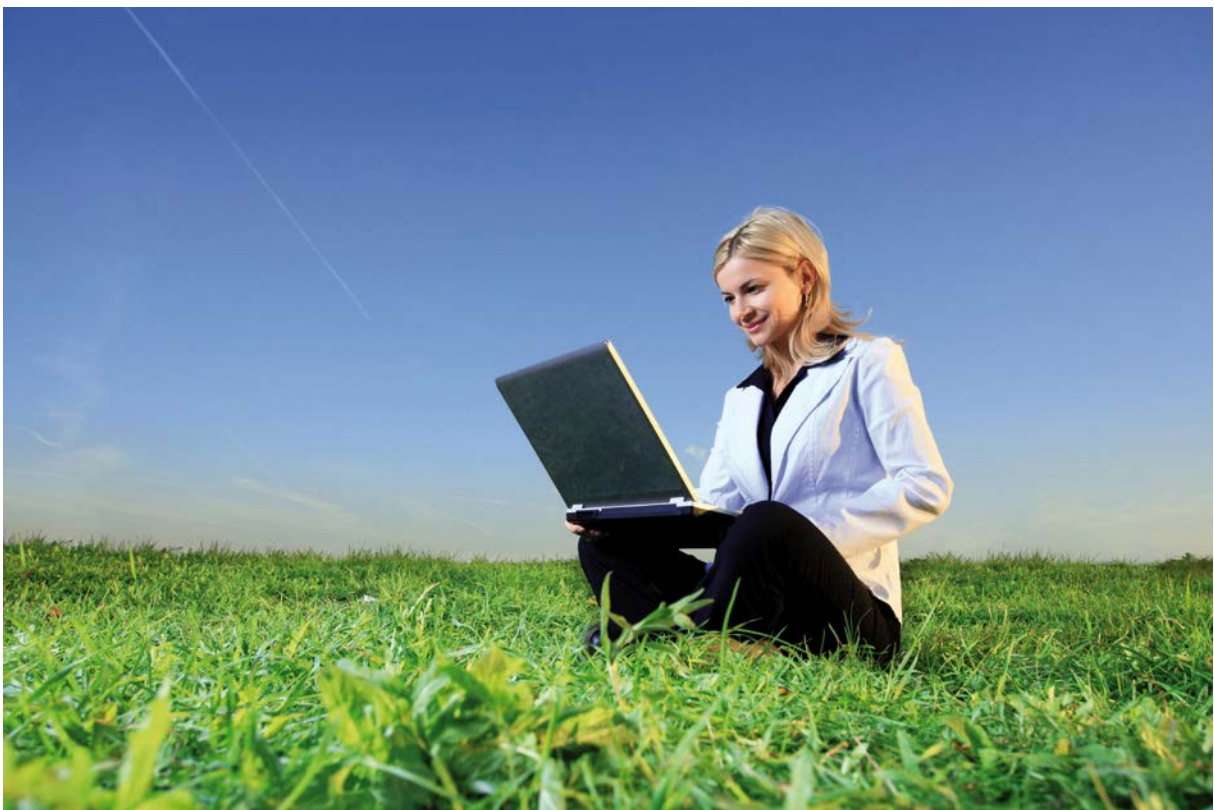


# ***Java Web Services***

***Seminarunterlage***

***Version: 3.01***



Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

## Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG  
Westernmauer 12-16  
D-33098 Paderborn  
Tel.: (+49) 0 52 51 / 10 63 - 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
An der alten Ziegelei 5  
D-48157 Münster  
Tel.: (+49) 02 51 / 9 24 35 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Marlene-Dietrich-Str. 5  
D-89231 Neu-Ulm  
Tel.: (+49) 07 31 / 9 85 88 – 550  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Kreuzberger Ring 13  
D-65205 Wiesbaden  
Tel.: (+49) 06 11 / 7 78 40 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Wikingerstraße 18-20  
D-51107 Köln  
Tel.: (+49) 02 21 / 8 70 61 – 0  
Fax.: (+49) 01 80 / 1 67 34 90

Sie können die ORDIX AG von der ganzen Welt aus durch folgende Internet Adresse kontaktieren:  
<http://www.ordix.de>

Sie können uns weiterhin über die Email-Adressen [training@ordix.de](mailto:training@ordix.de) oder [info@ordix.de](mailto:info@ordix.de) kontaktieren

**Inhaltsverzeichnis**

<b>1</b>	<b>Einführung .....</b>	<b>7</b>
1.1	Was sind Web Services? .....	8
1.2	Ziele .....	9
1.3	Architektur .....	10
1.4	Eigenschaften .....	11
1.5	Standards.....	14
1.6	Vorteile.....	17
1.7	Nachteile .....	19
1.8	Frühere Ansätze .....	20
<b>2</b>	<b>XML.....</b>	<b>22</b>
2.1	eXtensible Markup Language .....	23
2.2	Unterschiede HTML und XML .....	24
2.3	Verwendung von XML.....	25
2.4	Verwandte Technologien .....	26
2.5	Wohlgeformtheit.....	27
2.6	Gültigkeit.....	28
2.7	Grundlegender Aufbau.....	29
2.8	DTD (Document Type Definiton) .....	31
2.9	Was ist ein XML-Schema .....	33
2.10	Aufbau einer XML-Schema-Datei .....	34
2.11	Syntax XML Schema .....	36
2.12	Datentypen XML Schema.....	39
2.13	Namensräume (Namespaces).....	40
	2.13.1 Verwendung von Namensräumen .....	41
	2.13.2 Default-Namensraum.....	42
2.14	XML-Standards .....	43
<b>3</b>	<b>SOA, Web Services und Realisierungsstrategien .....</b>	<b>44</b>
3.1	SOA.....	45
3.2	Services .....	46
3.3	Ziele von SOA.....	47
3.4	SOA Themen .....	48
3.5	Umsetzung einer SOA mit Web Services .....	50
3.6	Contract .....	51
3.7	„Code first“-Ansatz .....	52
3.8	„Contract first“ Ansatz .....	53
3.9	„Code first“ vs. „Contract first“.....	54
3.10	“Code first” vs. “Contract first” – Fazit.....	55
<b>4</b>	<b>Web Services mit JAX-WS .....</b>	<b>56</b>
4.1	JAX-WS .....	57
	4.1.1 Details JAX-WS.....	58
	4.1.2 JAX-WS Eigenschaften.....	59
	4.1.3 Web Services Komponenten und JAX-WS.....	60
	4.1.4 JAX-WS Server.....	61
	4.1.5 JAX-WS Architektur .....	62
	4.1.6 JAX-WS Web Service erstellen .....	63
4.2	Beispiel-Projekt.....	64
	4.2.1 Service erstellen.....	65
	4.2.2 Exception-Klasse erstellen.....	66
	4.2.3 Server erstellen und starten.....	67
	4.2.4 Schnittstellenbeschreibung .....	68
	4.2.5 Client-Klassen erstellen .....	72
	4.2.6 Client implementieren .....	73
	4.2.7 Client/Service testen .....	74

4.2.8	Annotationen .....	75
4.2.8.1	@WebService .....	76
4.2.8.2	@WebMethod .....	77
4.2.8.3	@WebParam .....	78
4.2.8.4	Sonstige wichtige Annotationen .....	79
<b>5</b>	<b>SOAP .....</b>	<b>80</b>
5.1	Was ist SOAP? .....	81
5.2	Ziele .....	82
5.3	Historie .....	83
5.4	Aufbau einer SOAP-Nachricht .....	84
5.5	Aufbau eines SOAP-Requests .....	85
5.6	Aufbau eines SOAP-Response .....	86
5.7	SOAP-Envelope .....	87
5.8	SOAP-Header .....	88
5.9	Attribute „must understand“ .....	89
5.10	SOAP-Body .....	90
5.11	SOAP-Faults Version 1.1 .....	91
5.12	SOAP-Faults Version 1.2 .....	93
5.13	Vordefinierte SOAP-Fehlercodes .....	94
5.14	Unterschiede SOAP 1.1 / SOAP 1.2 .....	96
<b>6</b>	<b>Web Services Description Language (WSDL) .....</b>	<b>98</b>
6.1	Was ist WSDL .....	99
6.2	WSDL in der WebServices Architektur .....	100
6.3	Aufgaben der Dienstbeschreibung .....	101
6.4	Prinzipieller Aufbau von WSDL 1.1 .....	102
6.5	WSDL 1.1 Aufbau: abstrakter und konkreter Teil .....	103
6.6	WSDL 1.1 XML Grundgerüst .....	104
6.7	WSDL 1.1: Grundstruktur Calculator WebService (fiktiv) .....	105
6.8	WSDL 1.1: abstrakter Teil <types> .....	106
6.9	WSDL 1.1: abstrakter Teil <message> .....	107
6.10	WSDL 1.1: abstrakter Teil <portType> .....	108
6.11	WSDL 1.1: Message Exchange Pattern .....	109
6.12	Bindung: verschiedene Arten .....	111
6.13	Bindung: Beispiel .....	112
6.14	Bindung: RPC/encoded .....	113
6.15	Bindung: RPC/encoded – Eigenschaften .....	114
6.16	Bindung: RPC/literal .....	115
6.17	Bindung: RPC/literal – Eigenschaften .....	116
6.18	Bindung: Document/encoded .....	117
6.19	Bindung: Document/literal .....	118
6.20	Bindung: Document/literal – Eigenschaften .....	119
6.21	Bindung: Suche nach besserer Lösung .....	120
6.22	Bindung: Document/literal wrapped .....	121
6.23	Bindung: Document/literal wrapped – Fazit .....	123
6.24	WSDL 1.1: konkreter Teil <binding> .....	124
6.25	WSDL 1.1: konkreter Teil <service> .....	125
6.26	WSDL 1.1 Dokument im Überblick .....	126
6.27	WSDL 1.1 vs. WSDL 2.0 .....	127
6.28	Grundgerüst WSDL: 1.1 vs. 2.0 .....	128
6.29	Vergleich WSDL 1.1 und 2.0 .....	129
6.30	WSDL 2.0: <interface> .....	130
6.31	WSDL 2.0: <service> .....	131
6.32	Best Practices WSDL und SOAP .....	132
<b>7</b>	<b>Apache Axis2 .....</b>	<b>133</b>
7.1	Was ist Axis2? .....	134
7.2	Features von Axis2 .....	135

7.3	Warum Axis2?.....	136
7.4	Installation.....	137
7.5	Installation Tomcat 6.0 Servlet-Container.....	138
7.6	Installation Axis2 WAR-Datei.....	139
7.7	Installation Axis2 Standard Binary Distribution.....	140
7.8	Installation Axis2 Code Generator (Eclipse-PlugIn).....	141
7.9	Entwicklung von Web Services nach „Contract first“ Ansatz.....	142
7.10	Der „vereinfachte“ Contract.....	143
7.11	Vorbereitung (Eclipse) 1.....	144
7.12	Vorbereitung (Eclipse) 2.....	145
7.13	Schritt 1: Definition der Datentypen.....	146
7.14	Schritt 1a: XML-Schema mit Eclipse.....	147
7.15	Schritt 2: Definition der Schnittstellen.....	148
7.16	Schritt 3: Definition der Bindungen.....	149
7.17	Schritt 3a: WSDL-Datei mit Eclipse.....	150
7.18	Schritt 4: Codegenerierung.....	151
7.19	Schritt 4a: Java Code from WSDL.....	152
7.20	Schritt 4b: WSDL-Datei.....	153
7.21	Schritt 4c: Server-Code.....	154
7.22	Schritt 4d: Ziel festlegen.....	155
7.23	Schritt 4e: Erzeugter Code.....	156
7.24	Schritt 5: Detailimplementierung.....	157
7.25	Schritt 5a: Konstruktion der Geschäftsobjekte.....	158
7.26	Schritt 5b: Implementierung der Service-Methoden.....	159
7.27	Schritt 5c: Implementierung der Service-Methoden.....	160
7.28	Anpassen der services.xml.....	161
7.29	Erzeugen des Web Service Archivs.....	162
7.30	Deploy des Web Service Archivs.....	163
7.31	Test des Web Services.....	164
7.32	Entwicklung eines Clients.....	165
7.33	Java Code from WSDL.....	166
7.34	WSDL-Datei.....	167
7.35	Ziel festlegen.....	169
7.36	Erzeugter Client-Code.....	170
7.37	Implementierung.....	171
7.38	Test.....	172
<b>8</b>	<b>Tools.....</b>	<b>173</b>
8.1	Fehlersuche.....	174
8.2	Maßnahmen zur Fehlersuche.....	175
8.3	Logging.....	176
8.4	Debugging: Client.....	177
8.5	Remote Debugging: Server.....	178
8.6	Remote Debugging: Eclipse.....	179
8.7	TCPMon.....	180
8.8	TCPMON: Konfiguration.....	181
8.9	TCPMon: Datenansicht.....	182
8.10	SOAPMonitor.....	183
8.11	SOAPMonitor: Aktivierung.....	184
8.12	SOAPMonitor: Servlet.....	185
8.13	SOAPMonitor: Aufruf.....	186
<b>9</b>	<b>RESTful WebServices.....</b>	<b>187</b>
9.1	Historie.....	188
9.2	Motivation für REST.....	189
9.3	Interoperabilität.....	190
9.4	Entkoppelung.....	191
9.5	Skalierbarkeit und Perfomance.....	192
9.6	Wiederverwendung.....	193

9.7	Grundprinzipien.....	194
9.8	Prinzip: Eindeutige Identifikation von Ressourcen .....	195
9.9	Prinzip: Repräsentationen von Ressourcen .....	196
9.10	Prinzip: Hypermedia.....	197
9.11	Prinzip: uniforme Schnittstelle .....	198
9.12	Prinzip: uniforme Schnittstelle (Verhalten).....	199
9.13	Prinzip: uniforme Schnittstelle (Garantien/Verben) .....	200
9.14	HATEOAS – REST Anwendungsarchitektur .....	201
9.15	RESTful Webanwendungen .....	202
9.16	Vorteile von RESTful Anwendungen .....	203
9.17	Vergleich von Webservices SOAP vs. REST .....	204
9.18	JAX-RS .....	205
9.19	Beispiel mit Jersey.....	206
9.20	Projekt anlegen.....	207
9.21	Fachklasse.....	208
9.22	Ressource-Klasse.....	209
9.23	Annotationen.....	210
9.24	Test.....	211
<b>10</b>	<b>Webservices Sicherheit.....</b>	<b>212</b>
10.1	Anforderungen an sichere WebServices .....	213
10.2	Möglichkeiten für sichere WebServices.....	214
10.3	Transportsicherheit: SSL/TLS → HTTPS .....	215
10.4	SSL-Handshake.....	218
10.5	Tomcat-Konfiguration für SSL/HTTPS.....	219
10.6	HTTP-Authentifizierung mit Tomcat und JAX-WS.....	223
10.7	WS-Security .....	230
10.8	WS-Security Spezifikationen und Standards.....	232
10.9	Public-Key-Verfahren: Verschlüsselung und Signierung.....	233
10.10	Digitale Signaturen.....	234
10.11	XML-Security-Dokumente.....	235
10.12	W3C – XML Signature .....	236
10.13	W3C – XML Signature – XML-Struktur.....	237
10.14	W3C – XML Signature: Signaturformen .....	238
10.15	W3C – XML Signature (Beispiel) .....	239
10.16	W3C – XML Signature: Referenzen, Transformation, Kanonisierung.....	240
10.17	W3C – XML Signature: Algorithmen.....	241
10.18	W3C – XML Encryption.....	243
10.19	W3C – XML Encryption: XML-Struktur .....	244
10.20	W3C – XML Encryption: Beispiel.....	245
10.21	W3C – XML Encryption: Schlüsseltransport.....	246
10.22	OASIS – Web Services Security (WS-Security) .....	247
10.23	OASIS – Token Profile.....	248
10.24	OASIS – WS-Security: Bsp. Username Token in SOAP.....	249
10.25	OASIS – WS-Security: Bsp. BinarySecurity Token in SOAP .....	250
	<b>Anhang A: UDDI.....</b>	<b>251</b>
	<b>Anhang B: StAX.....</b>	<b>278</b>
	<b>Anhang C: AXIOM .....</b>	<b>288</b>