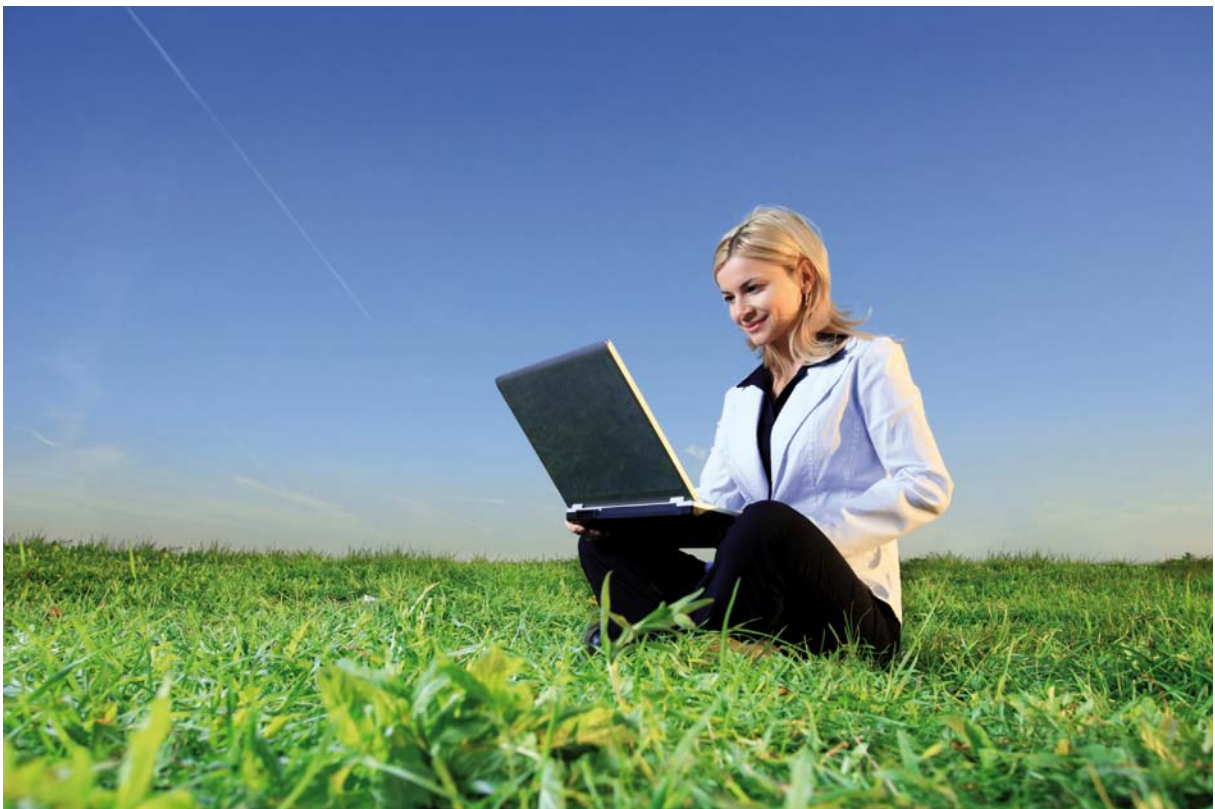


XML unter Java mit DOM und SAX

Seminarunterlage

Version: 1.04



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 oder info@ordix.de kontaktieren

Inhaltsverzeichnis

1	Simple API for XML (SAX)	5
1.1	Spezifikationen zu XML	6
1.1.1	XML (Extensible Markup Language).....	6
1.1.2	XSL (Extensible Stylesheet Language), XSLT (XSL Transformations).....	6
1.1.3	XML Schema.....	7
1.1.4	DTD (Document Type Definition).....	7
1.1.5	DOM (Document Object Model).....	7
1.2	Verarbeitung von XML (mit Java)	8
1.2.1	SAX (Simple API for XML)	8
1.2.2	JAXP (Java API for XML Processing).....	9
1.2.3	JAXB (Java Architecture for XML Binding)	9
1.3	Was ist SAX?	10
1.4	Event-Model des Sax-Parsers	12
1.5	Beispiel Parsen mit JAX-P	13
1.6	Beispiel Definition eines eigenen DefaultHandlers	15
1.7	SaxParser / DefaultHandler	17
1.8	Auswerten eines Dokuments	19
1.9	Start und Ende des XML-Dokuments	20
1.10	Auswerten des Start-Tags und des Ende-Tags.....	21
1.11	Auswerten von character data (PCDATA)	23
1.12	Namensräume berücksichtigen	24
1.13	Namensräume de-/aktivieren.....	25
1.14	Das Interface ErrorHandler	26
1.15	Validierung	27
1.16	Beispiel mit DTD-Validierung.....	28
1.17	Beispiel mit Schema-Validierung	30
1.18	Weitere Handler	32
1.19	Filter - Pipelines - Verarbeitungsketten.....	33
1.19.1	Einsatz eines Filters	35
1.19.2	Beispiel für einen Filter	36
1.19.3	Einsatz von zwei Filtern	38
1.19.4	Beispiel zum Einsatz von zwei Filtern.....	39
2	Document Object Model (DOM)	40
2.1	Was ist DOM?.....	41
2.2	Warum nicht SAX?.....	42
2.3	Baumstruktur.....	43
2.4	Das Interface org.w3c.dom.Document	44
2.5	Composite Pattern	46
2.6	Das Interface org.w3c.dom.Node	48
2.7	Namespaces	52
2.8	Parsen mittels DOM.....	54
2.9	Ausgeben eines Dokuments.....	56
2.10	DOM-Baum durchlaufen und ausgeben	58
2.11	DTD-Validierung	61
2.12	Fehler-Behandlung	63
2.13	XML-Schema-Validierung.....	65
2.14	XML-Dokumente erzeugen.....	67
2.15	Verwendung von XPath	69
3	Data Binding	71
3.1	Was ist Data Binding?.....	72
3.2	Java Architecture for XML Binding 2.0 (JAXB).....	74
3.3	Klassengenerierung beim Data Binding	76
3.4	Un-/Marshalling bei Data Binding	77
3.5	Zyklus des Data Binding	79

3.6	unmarshal: XML -> Java.....	80
3.7	marshal: Java -> XML.....	82
3.8	marshal und unmarshal: Erweitertes Beispiel	85
3.9	Annotations	88
3.10	@XmlRootElement	90
3.11	@XmlElement und @XmlAttribute	91
3.12	@XmlAccessorType	93
3.13	XmlAccessType.PUBLIC_MEMBER.....	95
3.14	XmlAccessType.PROPRTY.....	96
3.15	XmlAccessType.FIELD.....	97
3.16	XmlAccessType.NONE.....	98
3.17	@XmlType	99
3.18	@XmlTransient.....	100
3.19	Beispiel: Annotierte Klasse - Mensch	101
3.20	Java Anwendung und XML Data Binding	103
3.21	Klassengenerierung.....	104
3.22	Das Tool „schemagen“.....	105
3.23	Das Tool „xjc“.....	106
3.24	Die Tools „schemagen“ und „xjc“.....	107
4	Streaming API for XML (StAX).....	109
4.1	Was ist StAX?	110
4.2	Warum StAX?	111
4.3	Pull-Parsing vs. Push-Parsing	112
4.4	Anwendungsfälle der StAX – Spezifikation	113
4.5	Die Kern-API von StAX.....	114
4.6	Die StAX Cursor-API: XMLStreamReader.....	115
4.7	Die StAX Cursor-API: XMLStreamWriter.....	116
4.8	Die StAX Cursor-API: StreamFilter.....	117
4.9	Die StAX Event-Iterator-API	118
	4.9.1 XMLEventReader.....	120
	4.9.2 XMLEventWriter.....	121
4.10	Die StAX Cursor-API: EventFilter	122
4.11	Event-Iterator-API oder Cursor-API.....	123
4.12	Die Verwendung von StAX	124
4.13	Die XMLInputFactory	125
4.14	Die XMLOutputFactory	126
4.15	Die XMLEventFactory.....	127
4.16	Lesen von XML-Streams	128
4.17	Schreiben von XML-Streams.....	129
4.18	Lesen von XML-Events.....	130
4.19	Schreiben von XML-Events	131