

Inhaltsverzeichnis

1	Einstieg in XML und JSON	1
1.1	Basiswissen XML	2
1.1.1	Bestandteile und Aufbau eines XML-Dokuments	5
1.1.2	Validierung eines XML-Dokuments	10
1.2	XML-Verarbeitung mit JAXP	14
1.2.1	Einfaches Parsing mit SAX	15
1.2.2	Komplexere Parsing-Aufgaben mit SAX	18
1.2.3	Parsing mit DOM	21
1.2.4	Verarbeiten und Speichern mit DOM	25
1.2.5	StAX als Alternative zu SAX oder DOM?	29
1.2.6	SAX, StAX oder DOM?	36
1.2.7	XPath im Überblick	37
1.2.8	XSLT im Überblick	42
1.2.9	<code>XMLEncoder</code> und <code>XMLDecoder</code> im Überblick	46
1.3	XML-Verarbeitung mit JAXB	49
1.3.1	Schritt 1: Passende Java-Klassen erstellen	50
1.3.2	Schritt 2: Marshalling und Unmarshalling	54
1.3.3	JAXB: Stärken und Schwächen	58
1.4	JAXB und StAX in Kombination	59
1.4.1	Rekonstruktion von Objekten mit JAXB und StAX	59
1.4.2	Vergleich zu SAX, DOM und JAXB	62
1.4.3	On-the-Fly-Modifikation von Objekten	65
1.5	JSON – das bessere XML?	67
1.5.1	Crashkurs JSON	67
1.5.2	JSON mit Java verarbeiten	68
1.5.3	JSON vs. XML	73
1.6	Weiterführende Literatur	74
2	Einführung in Persistenz und relationale Datenbanken	75
2.1	Grundlagen zur Persistenz	76
2.1.1	Beschränkungen einfacher Persistenzlösungen	76
2.1.2	Modelle zur Persistierung von Objekten	78
2.1.3	Speicherung von Daten in relationalen Datenbanken	79

2.2	Abbildung zwischen Objekt- und Datenbankmodell	86
2.2.1	Abbildung von Referenzen	88
2.2.2	Abbildung von Assoziationen und Aggregationen	91
2.2.3	Abbildung von Vererbung	94
2.3	Das Datenbanksystem HSQLDB im Kurzüberblick	98
2.4	SQL-Grundlagen	100
2.4.1	DDL – Definition von Tabellen	101
2.4.2	DQL – Datenabfrage	107
2.4.3	DML – Datenmanipulation	111
2.5	Ausfallsicherheit und Replikation	114
2.6	Weiterführende Literatur	115
3	Persistenz mit JDBC	117
3.1	Datenbankzugriffe per JDBC	117
3.1.1	Schritte zur Abfrage von Datenbanken	120
3.1.2	Besonderheiten von <code>ResultSet</code>	128
3.1.3	Abfrage von Metadaten	134
3.1.4	Probleme bei der Ausführung von <code>Statements</code>	142
3.1.5	Das Interface <code>PreparedStatement</code>	145
3.1.6	Transaktionen in JDBC	148
3.2	Grundlagen zum ORM mit JDBC	151
3.2.1	Rekonstruktion von Objekten	151
3.2.2	Zugriffe mit einem Data Access Object (DAO)	156
3.3	Weiterführende Literatur	159
4	Persistenz mit JPA	161
4.1	Grundlagen zum ORM und zum JPA	162
4.2	Einführung in JPA an einem Beispiel	164
4.2.1	Definition persistenter Klassen	164
4.2.2	Die Konfigurationsdatei <code>persistence.xml</code>	167
4.2.3	Datenbankzugriffe per JPA in Java SE	168
4.2.4	Lebenszyklus von Entitäten (Entity Lifecycle)	173
4.2.5	Datenbankmodell	175
4.2.6	Vorteile der konfigurativen Persistenz	177
4.3	JPQL im Überblick	178
4.3.1	Syntax von JPQL	178
4.3.2	Besondere Arten von Queries	181
4.3.3	Abfragen mit JPQL ausführen	182
4.3.4	Typsichere Abfragen und das Criteria API	188
4.4	DAO-Funktionalität mit JPA	191
4.4.1	CRUD-Funktionalität	191
4.4.2	Einsatz des DAO	194
4.5	Fortgeschritteneres ORM mit JPA	196

4.5.1	Abbildung von Assoziationen	197
4.5.2	Abbildung von Vererbungshierarchien	200
4.5.3	Verarbeitung der Typen aus JSR-310: Date and Time	204
4.5.4	Bean Validation im Einsatz	207
4.6	Transaktionen und Locking	211
4.6.1	Isolationslevel und Effekte	211
4.6.2	Problemkontext	212
4.6.3	Optimistic Locking	213
4.7	Caching in JPA	216
4.8	Fazit	218
4.9	Weiterführende Literatur	218
5	NoSQL-Datenbanken am Beispiel von MongoDB	219
5.1	Einführung und Überblick	219
5.2	Einführung MongoDB	225
5.2.1	Analogie von CRUD (RDBMS) zu IFUR (MongoDB)	227
5.2.2	Komplexere Abfragen	232
5.2.3	MongoDB und Transaktionen	235
5.3	Ausfallsicherheit und Skalierbarkeit	236
5.3.1	Hintergrundwissen: Formen der Skalierung	236
5.3.2	Ausfallsicherheit und Replica Sets	238
5.3.3	Skalierung und Sharding	239
5.3.4	Anmerkungen zu Replica Sets und Sharding	241
5.4	MongoDB aus Java ansprechen	241
5.4.1	Einführendes Beispiel	242
5.4.2	Daten einfügen und auslesen	243
5.4.3	Verarbeitung komplexerer Daten	247
5.4.4	Einfaches JSON-basiertes Object/Document Mapping	252
5.4.5	Object/Document Mapping mit Spring Data MongoDB	255
5.5	Fazit	267
5.6	Weiterführende Literatur	268
6	REST-Services mit JAX-RS und Jersey	269
6.1	REST im Kurzüberblick	270
6.1.1	Einführendes Beispiel eines REST-Service	272
6.1.2	Zugriffe auf REST-Services	276
6.1.3	Unterstützung verschiedener Formate	278
6.1.4	Zugriffe auf REST-Services am Beispiel von MongoDB	280
6.2	Ein REST-Service mit CRUD-Funktionalität	283
6.2.1	MIME-Types und unterschiedliche Datenformate	283
6.2.2	HTTP-Kommandos und CRUD-Funktionalität	285
6.3	Tipps zum Design von REST-Interfaces	290
6.3.1	Varianten der Rückgabe und Error Handling bei REST	290

6.3.2	Wertübergabe als @QueryParam oder @PathParam	293
6.3.3	Paging bei GET	294
6.4	Fortgeschrittene Themen	295
6.4.1	Einsatz von Request- und Response-Filtern	295
6.4.2	Security im Kontext von REST	299
6.4.3	Testen mit restfuse	302
6.5	Fazit	304
6.6	Weiterführende Literatur	305
7	Entwurf einer Beispielapplikation	307
7.1	Iteration 0: Ausgangsbasis	307
7.2	Iteration 1: Zentrale Verwaltung von Highscores	311
7.3	Iteration 2: Verwaltung von XML	316
7.4	Iteration 3: Bereitstellen als REST-Service	322
7.5	Iteration 4: Web-GUI mit HTML und JavaScript	328
7.6	Iteration 5: Protokollierung von Aktionen mit MongoDB	336
7.7	Fazit zum Abschluss der Iterationen	341
A	Einführung Gradle	345
A.1	Projektstruktur für Maven und Gradle	345
A.2	Builds mit Gradle	347
B	Client-Server-Kommunikation und HTTP im Überblick	355
B.1	Client-Server-Kommunikation	355
B.2	Basiswissen HTTP	357
C	Grundlagenwissen HTML	363
C.1	Basiswissen HTML	363
C.1.1	HTML am Beispiel	364
C.1.2	Interaktivität und Formulare	366
D	Wissenswertes zu JavaScript	371
D.1	Grundlagen zur Sprache	371
D.2	Modifikation von HTML	373
D.3	JSON-Verarbeitung	377
D.4	REST-Services ansprechen	378
	Literaturverzeichnis	379
	Index	381