

Richard Oates
Thomas Langer
Stefan Wille
Torsten Lueckow
Gerald Bachlmayr

Spring & Hibernate

Eine praxisbezogene Einführung

2., aktualisierte Auflage

HANSER

Inhaltsverzeichnis

1	Einleitung	1
1.1	Für wen ist dieses Buch?	1
1.2	Organisation des Buches	2
1.3	Web-Site zum Buch	3
2	Überblick und Installation	5
2.1	Management Summary	5
2.1.1	Was ist Spring?	5
2.1.2	Was ist Hibernate?	6
2.1.3	Kombination von Spring und Hibernate	6
2.2	Installation	7
2.2.1	Download von Spring	8
2.2.2	Download von Hibernate	9
2.2.3	Ein IDE-Projekt mit Hibernate einrichten	10
2.2.4	Ein Spring-Projekt	11
3	Software-Architektur der Beispielanwendung	13
3.1	Opiz – Online-Pizza-Dienst	13
3.1.1	Fachliche Use-Cases	13
3.1.2	Domain-Modell	15
3.1.3	Organisation in Eclipse	17
3.2	Architektur-Überblick	17
3.2.1	High-Level-Architektur	17
3.2.2	Data-Access-Objekt-Schicht	19
3.2.3	Service-Schicht	20
3.2.4	Präsentationsschicht bei Web-Anwendungen	21

4	Einführung in Hibernate	23
4.1	Hibernate als O/R-Mapper	23
4.2	Überblick über die Hibernate-Architektur	25
4.2.1	Persistente Klassen	25
4.2.2	Die Hibernate-Session	26
4.2.3	Hibernate-Session-Factory	26
4.2.4	Hibernate-Mapping	27
4.3	Hibernate in der Praxis	28
4.3.1	Mapping mit Annotations	29
4.3.2	Konfiguration der Session-Factory	31
4.3.3	Verzeichnisstruktur	33
4.3.4	Initialisierung der Session-Factory	33
4.3.5	Speichern eines Objekts	34
4.3.6	Laden eines Objekts	37
4.3.7	Suchen eines Objekts	38
4.3.8	Verändern eines Objekts	40
4.3.9	Löschen eines Objekts	41
4.3.10	Exception-Handling	41
4.3.11	Logging	41
4.4	Persistency-Lifecycle	43
5	O/R-Mapping mit Hibernate	45
5.1	Mapping von persistenten Klassen	46
5.2	Mapping von einfachen Properties	46
5.2.1	Persistente Properties	46
5.2.2	Spalteneigenschaften festlegen	47
5.2.3	Enumerations	48
5.2.4	Zeitbasierte Properties	49
5.2.5	Blobs und Clobs	49
5.3	Mapping von Identifier-Properties	50
5.3.1	Manuell vergebene Schlüssel	50
5.3.2	Automatisch generierte Schlüssel	51
5.3.3	Natürliche Schlüssel	53
5.4	Mapping von Beziehungen	54
5.4.1	1:N-Beziehungen	54

5.4.2	Bags, Lists, Arrays, Sets, Collections und Maps	62
5.4.3	Equals und hashCode implementieren	64
5.4.4	M:N-Beziehungen	66
5.4.5	1:1-Beziehungen	69
5.4.6	Unidirektionale Beziehungen rückwärts navigieren	74
5.4.7	Lazy Loading	75
5.5	Mapping von Vererbung	77
5.5.1	Eine Tabelle für die ganze Klassenhierarchie	78
5.5.2	Eine Tabelle pro Klasse	81
5.5.3	Eine Tabelle pro konkreter Subklasse	82
5.5.4	Polymorphe Beziehungen	82
5.6	Mapping mit XML	83
5.7	Weitere Mapping-Elemente	84
6	Einführung in Spring	85
6.1	Grundlagen	85
6.1.1	Lightweight-Container	85
6.1.2	Inversion of Control	86
6.1.3	Dependency-Injection	87
6.2	Überblick und Einstieg in Spring	89
6.2.1	Übersicht über die Spring-Module	89
6.2.2	Erstellung eines Spring-Projekts	92
6.2.3	Konfiguration des Application-Contexts	95
6.3	Spring-Beans und ihre Definitionen	96
6.3.1	Erzeugung einer Spring-Bean	96
6.3.2	Properties und Abhängigkeiten von Spring-Beans	98
6.3.3	Collections und andere Util-Klassen	104
6.3.4	Der Life-Cycle von Spring-Beans	107
6.4	Spring Inversion-of-Control-Container	109
6.4.1	Die Bean-Factory-Schnittstelle	109
6.4.2	Weitergehende Funktionalität mit dem Application-Context	110
6.5	Individualisieren des Containers	112
6.5.1	Veränderung von Bean-Definitionen	112
6.5.2	Nachbearbeitung von Beans	114
6.5.3	Definition individueller Property-Editors	116

7 Die Datenzugriffsschicht	119
7.1 Vorteile von Spring im Kontext eines O/R-Mappers	120
7.2 Hibernate-Konfiguration mit Spring	121
7.2.1 Definition der JDBC-DataSource	121
7.2.2 Definition der Hibernate-SessionFactory	122
7.3 Abfragen mit einem Data-Access-Objekt	124
7.3.1 Objekte mit dem Primärschlüssel laden	124
7.3.2 Konfiguration der DAO-Klasse als Spring-Bean	125
7.3.3 Eine DAO-Finder-Methode	125
7.4 Test der DAO-Klassen	127
7.4.1 Das Spring-Test-Framework	127
7.4.2 Test einer DAO-Abfrage-Methode	129
7.4.3 Weitere Methoden in der Spring-Test-Klassenhierarchie	132
7.4.4 Initialisierung der Test-Klasse	133
7.4.5 Autowire by Name	133
7.5 Weitere Features der Hibernate Query Language	134
7.5.1 Die select-Klausel	134
7.5.2 Funktionen	135
7.5.3 Die from-Klausel	136
7.5.4 Die where-Klausel	137
7.5.5 Subquery	137
7.5.6 Gruppierung	138
7.6 Zusätzliche Hibernate-Zugriffsmöglichkeiten	138
7.6.1 Weitere Methoden in HibernateTemplate	138
7.6.2 Die HibernateCallback-Klasse	141
7.6.3 Eine Paged-Abfrage	141
7.6.4 DAO-Implementierung ohne HibernateDaoSupport	143
7.6.5 DAO-Implementierung ohne Spring	143
7.7 Die Criteria-API	145
7.7.1 Ein Criteria-basiertes Beispiel	145
7.7.2 Test einer Criteria-basierten Methode	146
7.7.3 Query-by-Example	146
7.7.4 Aggregat-Funktionen	148
7.8 DAO-Schreibmethoden	149
7.8.1 Neue Objekte speichern	149
7.8.2 Aktualisieren eines persistenten Objekts	150
7.8.3 Aktualisieren eines detachten Objekts	150

7.8.4	Das Flush-Problem in Tests	151
7.9	Arbeiten mit Stored Procedures	153
7.9.1	Aufruf über JDBC	153
7.9.2	Aufruf über Spring	154
7.9.3	Aufruf über Hibernate	155
7.9.4	Empfehlung	156
8	Services mit Spring	157
8.1	Allgemeine Themen bei der Implementierung der Service-Schicht	157
8.2	Der Use-Case „Pizza-Bestellung“	160
8.2.1	Implementierung einer Service-Methode	160
8.2.2	Test der Service-Methode	162
8.3	Aufnehmen einer weiteren Pizza in einer Bestellung	163
8.3.1	Testen mit Mock-Objekten	164
8.4	Spring AOP – einfach und mächtig	167
8.4.1	Konzept des Aspect-Oriented Programming	167
8.4.2	Spring-AOP: Annotations und XML	169
8.4.3	Pointcuts in Spring	170
8.4.4	Advices in Spring	171
8.4.5	Binärcode erzeugen – JDK versus CGLIB	175
8.4.6	Testen des Aspekts	175
8.4.7	Das Spring-Aspekt-Toolset	177
8.5	Transaktionen in Spring	179
8.5.1	Grundlagen und das ACID-Prinzip	180
8.5.2	Der Transaktionsmanager	181
8.5.3	Beeinflussung des Transaktionsverhaltens	183
8.5.4	Transaktionsverläufe	185
8.5.5	Deklarative Transaktionsdurchführung	187
8.5.6	Programmatische Transaktionsdurchführung	193
8.5.7	Durchführung von Transaktionstests	195
9	Webanwendungen mit Spring und Hibernate	199
9.1	Die Beispielanwendungen	199
9.2	Spring und Webanwendungen	202
9.2.1	Laden des Application-Contexts	203
9.2.2	Integration mit JavaServer Faces	205

9.2.3	Gültigkeitsbereich von Spring-Beans	207
9.3	Hibernate und Webanwendungen	209
9.3.1	OpenSessionInViewFilter	209
9.3.2	Detached Objekte	212
9.3.3	NonUniqueObjectException	217
9.3.4	Behandlung von Exceptions	219
9.3.5	Konkurrierender Zugriff	220
9.3.6	Nebenläufigkeit	222
9.3.7	Connection-Pooling	224
9.3.8	Caching	226
9.4	Zugriffsschutz mit Acegi	231
9.4.1	Authentifizierung	232
9.4.2	Autorisierung	237
9.4.3	Filter in Reihenschaltung	239
9.4.4	Acegi-Tags	242
9.4.5	Acegi und JSF	242
9.5	Alles im Fluss mit Spring Web Flow	244
9.5.1	Endlich automatisch	246
9.5.2	Spring Web Flow mit JavaServer Faces	248
9.5.3	Flow-Persistence-Context	251
9.6	Testmöglichkeiten	257
9.6.1	Test von JSF-Komponenten und Spring	257
9.6.2	Testbeispiel Bestellvorgang	260
10	Integration in die JEE-Welt	265
10.1	Eine RMI-Anbindung	267
10.1.1	Exportieren der Service-Schnittstelle	268
10.1.2	Konfiguration der Client-Anbindung	269
10.1.3	Implementierung der Client-Klasse	270
10.2	Die EJB-Schicht der Opiz-Anwendung	271
10.3	Implementierung einer Stateless-Session-Bean	272
10.3.1	Die Remote- und Home-Schnittstellen	272
10.3.2	Die Implementierung der Session-Bean	273
10.3.3	Die BeanFactory finden	274
10.3.4	Deployment	276
10.4	Stateful-Session-Beans	277

10.4.1 Remote- und Home-Schnittstellen	277
10.4.2 Die Implementierung der Stateful-Session-Bean	278
10.4.3 Deployment	279
10.5 Message-Driven-Beans	279
10.6 Client-Anbindung an den EJB-Container	279
10.6.1 Zugriff auf eine Stateless-Session-Bean	280
10.6.2 Zugriff auf eine Stateful-Session-Bean	282
10.7 Testmöglichkeiten	284
10.8 Springs Unterstützung für JMX	286
10.8.1 Registrierung der Spring-Beans beim JMX-Server	287
10.8.2 Konfiguration mit einem Interface	288
10.8.3 Konfiguration mit Annotations	289
10.8.4 JMX ohne einen Applikations-Server?	291
10.9 Spring-Integration von JCA	293
10.9.1 Konfiguration des Resource-Adapters	294
10.9.2 Nutzung des Resource-Adapters	295
Stichwortverzeichnis	299