

HANSER

Spring & Hibernate

Thomas Langer, Gerald Bachlmayr, Torsten
Lueckow, Stefan Wille, Richard Oates

Eine praxisbezogene Einführung

ISBN 3-446-40457-0

Inhaltsverzeichnis

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/3-446-40457-0> sowie im Buchhandel

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	14
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	Lazy Loading	74
5.5	Mapping von Vererbung	76
5.5.1	Eine Tabelle für die ganze Klassenhierarchie	77
5.5.2	Eine Tabelle pro Klasse	80
5.5.3	Eine Tabelle pro konkreter Subklasse	81
5.5.4	Polymorphe Beziehungen	81
5.6	Mapping mit XML	82
5.7	Weitere Mapping-Elemente	83
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	94
6.3	Spring-Beans und ihre Definitionen	95
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	103
6.3.4	Der Life-Cycle von Spring-Beans	106
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	111
6.5.1	Veränderung von Bean-Definitionen	112
6.5.2	Nachbearbeitung von Beans	114
7	Die Datenzugriffsschicht	117
7.1	Vorteile von Spring im Kontext eines O/R-Mappers	118

7.2	Hibernate-Konfiguration mit Spring	119
7.2.1	Definition der JDBC-DataSource	119
7.2.2	Definition der Hibernate-SessionFactory	120
7.3	Abfragen mit einem Data-Access-Objekt	121
7.3.1	Objekte mit dem Primärschlüssel laden	122
7.3.2	Konfiguration der DAO-Klasse als Spring-Bean	123
7.3.3	Eine DAO-Finder-Methode	123
7.4	Test der DAO-Klassen	124
7.4.1	Das Spring-Test-Framework	125
7.4.2	Test einer DAO-Abfrage-Methode	126
7.4.3	Weitere Methoden in der Spring-Test-Klassenhierarchie	130
7.4.4	Initialisierung der Test-Klasse	130
7.4.5	Autowire by Name	131
7.5	Weitere Features der Hibernate Query Language	132
7.5.1	Die select-Klausel	132
7.5.2	Funktionen	133
7.5.3	Die where-Klausel	133
7.5.4	Subquery	134
7.6	Zusätzliche Hibernate-Zugriffsmöglichkeiten	134
7.6.1	Weitere Methoden in HibernateTemplate	135
7.6.2	Die HibernateCallback-Klasse	137
7.6.3	Eine Paged-Abfrage	138
7.6.4	DAO-Implementierung ohne HibernateDaoSupport	139
7.6.5	DAO-Implementierung ohne Spring	140
7.7	Die Criteria-API	141
7.7.1	Ein Criteria-basiertes Beispiel	141
7.7.2	Test einer Criteria-basierten Methode	142
7.7.3	Query-by-Example	142
7.7.4	Aggregat-Funktionen	144
7.8	DAO-Schreibmethoden	145
7.8.1	Neue Objekte speichern	145
7.8.2	Aktualisieren eines persistenten Objekts	146
7.8.3	Aktualisieren eines detachten Objekts	147
7.8.4	Das Flush-Problem in Tests	147
7.9	Arbeiten mit Stored Procedures	149
7.9.1	Aufruf über JDBC	150
7.9.2	Aufruf über Spring	150

7.9.3	Aufruf über Hibernate	151
7.9.4	Empfehlung	152
8	Services mit Spring	153
8.1	Allgemeine Themen bei der Implementierung der Service-Schicht	153
8.2	Der Use-Case Pizza-Bestellung	156
8.2.1	Implementierung einer Service-Methode	156
8.2.2	Test der Service-Methode	158
8.3	Aufnehmen einer weiteren Pizza in einer Bestellung	159
8.3.1	Testen mit Mock-Objekten	160
8.4	Spring AOP – einfach und mächtig	163
8.4.1	Konzept des Aspect-Oriented Programming	163
8.4.2	Spring-AOP: Annotations und XML	165
8.4.3	Pointcuts in Spring	166
8.4.4	Advices in Spring	167
8.4.5	Binärcode erzeugen – JDK versus CGLIB	171
8.4.6	Testen des Aspekts	172
8.4.7	Das Spring-Aspekt-Toolset	173
8.4.8	AOP mit Spring 1.2	175
8.5	Transaktionen in Spring	178
8.5.1	Grundlagen und das ACID-Prinzip	179
8.5.2	Der Transaktionsmanager	180
8.5.3	Beeinflussung des Transaktionsverhaltens	182
8.5.4	Transaktionsverläufe	183
8.5.5	Deklarative Transaktionsdurchführung	186
8.5.6	Programmatische Transaktionsdurchführung	192
8.5.7	Durchführung von Transaktionstests	195
8.5.8	Transaktionen mit Spring 1.2	196
9	Webanwendungen mit Spring und Hibernate	199
9.1	Die Beispielanwendung	199
9.2	Spring und Webanwendungen	202
9.2.1	Laden des Application-Contexts	202
9.2.2	Integration mit JavaServer Faces	204
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	221
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	233
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
10	Integration in die JEE-Welt	245
10.1	Eine RMI-Anbindung	247
10.1.1	Exportieren der Service-Schnittstelle	247
10.1.2	Konfiguration der Client-Anbindung	249
10.1.3	Implementierung der Client-Klasse	250
10.2	Die EJB-Schicht der Opiz-Anwendung	251
10.3	Implementierung einer Stateless-Session-Bean	252
10.3.1	Die Remote- und Home-Schnittstellen	252
10.3.2	Die Implementierung der Session-Bean	253
10.3.3	Die BeanFactory finden	254
10.3.4	Deployment	256
10.4	Stateful-Session-Beans	257
10.4.1	Remote- und Home-Schnittstellen	257
10.4.2	Die Implementierung der Stateful-Session-Bean	258
10.4.3	Deployment	259
10.5	Message-Driven-Beans	259
10.6	Client-Anbindung an den EJB-Container	259
10.6.1	Zugriff auf eine Stateless-Session-Bean	260
10.6.2	Zugriff auf eine Stateful-Session-Bean	263
10.7	Test-Möglichkeiten	265
10.8	Spring Unterstützung für JMX	269

10.8.1	Registrierung der Spring-Beans beim JMX-Server	269
10.8.2	Konfiguration mit einem Interface	270
10.8.3	Konfiguration mit Annotations	271
10.8.4	JMX ohne einen Applikations-Server?	273
10.9	Spring-Integration von JCA	274
10.9.1	Konfiguration des Resource-Adapters	275
10.9.2	Nutzung des Resource-Adapters	277
 Stichwortverzeichnis		 281