

HANSER

Christian Bauer, Gavin King

Java Persistence mit Hibernate

ISBN-10: 3-446-40941-6

ISBN-13: 978-3-446-40941-5

Inhaltsverzeichnis

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-40941-5>
sowie im Buchhandel



Inhalt

Geleitwort zur zweiten Auflage	XIII
Geleitwort zur ersten Auflage	XV
Vorwort zur zweiten Auflage	XVII
Vorwort zur ersten Auflage	XIX
Danksagung	XXI
Über dieses Buch.....	XXIII
Teil 1: Erste Schritte mit Hibernate und EJB 3.0	1
1 Objekt-relationale Persistenz.....	3
1.1 Was ist Persistenz?	5
1.1.1 Relationale Datenbanken	5
1.1.2 Die Grundlagen von SQL	6
1.1.3 SQL in Java nutzen	7
1.1.4 Persistenz in objektorientierten Applikationen	7
1.2 Die Unvereinbarkeit der Paradigmen	9
1.2.1 Das Problem der Granularität	11
1.2.2 Das Problem der Subtypen.....	12
1.2.3 Das Problem der Identität	14
1.2.4 Mit Assoziationen verbundene Probleme	15
1.2.5 Das Problem der Datennavigation	17
1.2.6 Die Kosten der Unvereinbarkeit der Paradigmen.....	18
1.3 Persistenzschichten und Alternativen	19
1.3.1 Schichtarchitektur	19
1.3.2 Eine Persistenzschicht mit SQL/JDBC handcodieren.....	20
1.3.3 Serialisierung	21
1.3.4 Objektorientierte Datenbanksysteme	22
1.3.5 Andere Optionen.....	23
1.4 Objekt-relationales Mapping	23
1.4.1 Was ist ORM?	24

1.4.2	Generische ORM-Probleme.....	26
1.4.3	Warum ORM?	27
1.4.4	Hibernate, EJB3 und JPA	29
1.5	Zusammenfassung.....	34
2	Ein neues Projekt beginnen.....	35
2.1	Ein Hibernate-Projekt beginnen	36
2.1.1	Auswahl eines Entwicklungsprozesses.....	36
2.1.2	Das Projekt aufsetzen	39
2.1.3	Konfiguration und Start von Hibernate.....	45
2.1.4	Starten und Testen der Applikation	55
2.2	Ein neues Projekt mit Java Persistence.....	62
2.2.1	Die Arbeit mit Hibernate Annotations.....	62
2.2.2	Die Arbeit mit Hibernate EntityManager	65
2.2.3	Die Komponenten von EJB	71
2.2.4	Wechsel zu Hibernate-Interfaces	77
2.3	Reverse Engineering einer Legacy-Datenbank.....	79
2.3.1	Erstellen einer Datenbankkonfiguration	80
2.3.2	Reverse Engineering anpassen.....	81
2.3.3	Generieren von Java-Quellcode.....	82
2.4	Integration mit Java EE-Diensten.....	86
2.4.1	Integration mit JTA	86
2.4.2	JNDI-gebundene SessionFactory.....	90
2.4.3	Bereitstellung von JMX-Diensten	92
2.5	Zusammenfassung.....	93
3	Domain-Modelle und Metadaten.....	95
3.1	Die Applikation CaveatEmptor	96
3.1.1	Analyse der Business-Domain.....	96
3.1.2	Das Domain-Modell für CaveatEmptor.....	97
3.2	Implementierung des Domain-Modells	100
3.2.1	Das Vermischen von Aufgabenbereichen.....	100
3.2.2	Transparente und automatische Persistenz	101
3.2.3	POJOs und persistente Entity-Klassen.....	103
3.2.4	Implementierung von POJO-Assoziationen.....	105
3.2.5	Logik in Zugriffs-Methoden einfügen	109
3.3	Objekt-relationale Mapping-Metadaten.....	111
3.3.1	Metadaten in XML	111
3.3.2	Auf Annotationen basierende Metadaten.....	113
3.3.3	Die Arbeit mit XDoclet	119
3.3.4	Umgang mit globalen Metadaten.....	120
3.3.5	Die Manipulation von Metadaten zur Laufzeit	125
3.4	Alternative Entity-Repräsentation	127
3.4.1	Erstellung von dynamischen Applikationen	128
3.4.2	Daten in XML repräsentieren	134
3.5	Zusammenfassung.....	137

Teil 2: Konzepte und Strategien für das Mapping	139
4 Mapping von Persistenzklassen	141
4.1 Entities- und Wert-Typen	141
4.1.1 Feingranulierte Domain-Modelle	142
4.1.2 Konzeptdefinition	142
4.1.3 Identifizierung von Entities und Wert-Typen	143
4.2 Entities mit Identität mappen	145
4.2.1 Identität und Gleichheit bei Java	145
4.2.2 Umgang mit Datenbankidentität	146
4.2.3 Primärschlüssel für Datenbanken	148
4.3 Optionen für das Mapping von Klassen	153
4.3.1 Dynamische SQL-Generierung	153
4.3.2 Eine Entity unveränderlich machen	154
4.3.3 Bezeichnung von Entities für Abfragen	155
4.3.4 Deklaration eines Paketnamens	156
4.3.5 Quoting von SQL-Identifikatoren	156
4.3.6 Implementierung von Namenskonventionen	157
4.4 Feingranulierte Modelle und Mappings	159
4.4.1 Mapping von grundlegenden Eigenschaften	159
4.4.2 Mapping von Komponenten	165
4.5 Zusammenfassung	170
5 Vererbung und selbst erstellte Typen	171
5.1 Mapping von Klassenvererbung	171
5.1.1 Tabelle pro konkrete Klasse mit implizitem Polymorphismus	172
5.1.2 Tabelle pro konkrete Klasse mit Unions	175
5.1.3 Tabelle pro Klassenhierarchie	177
5.1.4 Tabelle pro Subklasse	181
5.1.5 Mischen von Vererbungsstrategien	184
5.1.6 Wahl einer Strategie	186
5.2 Das Typsystem von Hibernate	188
5.2.1 Wiederholung von Entity- und Wert-Typen	188
5.2.2 Eingebaute Mapping-Typen	190
5.2.3 Die Arbeit mit Mapping-Typen	194
5.3 Erstellen eigener Mapping-Typen	196
5.3.1 Überlegungen zu eigenen Mapping-Typen	196
5.3.2 Die Extension Points	197
5.3.3 Über eigene Mapping-Typen	198
5.3.4 Erstellen eines UserType	199
5.3.5 Erstellen eines CompositeUserType	202
5.3.6 Parametrisierung eigener Typen	205
5.3.7 Mapping von Aufzählungen	207
5.4 Zusammenfassung	211

6	Mapping von Collections und Entity-Assoziationen	213
6.1	Sets, Multimengen, Listen und Maps mit Wert-Typen.....	213
6.1.1	Wahl eines Collection-Interfaces.....	214
6.1.2	Mapping eines Set	216
6.1.3	Mapping einer Identifikator-Multimenge	217
6.1.4	Mapping einer Liste.....	218
6.1.5	Mapping einer Map	219
6.1.6	Sortierte und geordnete Collections.....	220
6.2	Collections von Komponenten	222
6.2.1	Schreiben der Komponentenklasse.....	223
6.2.2	Mapping der Collection	223
6.2.3	Aktivieren der bidirektionalen Navigation	224
6.2.4	Vermeiden von not-null-Spalten.....	224
6.3	Mapping von Collections mit Annotationen.....	226
6.3.1	Grundlegendes Mapping von Collections.....	226
6.3.2	Sortierte und geordnete Collections.....	228
6.3.3	Eine Collection von eingebetteten Objekten mappen	228
6.4	Mapping einer Parent/Children-Beziehung	230
6.4.1	Kardinalität.....	231
6.4.2	Die einfachste mögliche Assoziation.....	231
6.4.3	Die Assoziation bidirektional machen	233
6.4.4	Kaskadierung des Objektzustands	236
6.5	Zusammenfassung.....	243
7	Fortgeschrittene Mappings für Entity-Assoziationen.....	245
7.1	Entity-Assoziationen mit einem Wert	246
7.1.1	Gemeinsame Primärschlüssel-Assoziationen	247
7.1.2	one-to-one-Fremdschlüssel-Assoziationen	250
7.1.3	Mapping mit einer Join-Tabelle.....	252
7.2	Mehrwertige Entity-Assoziationen.....	256
7.2.1	one-to-many-Assoziationen	257
7.2.2	many-to-many-Assoziationen.....	263
7.2.3	Zusätzliche Spalten bei Join-Tabellen	268
7.2.4	Mapping von Maps.....	273
7.3	Polymorphe Assoziationen.....	276
7.3.1	Polymorphe many-to-one-Assoziationen.....	276
7.3.2	Polymorphe Collections	278
7.3.3	Polymorphe Assoziationen mit Unions	279
7.3.4	Polymorphe Tabelle pro konkrete Klasse	281
7.4	Zusammenfassung.....	283
8	Legacy-Datenbanken und eigenes SQL	285
8.1	Integration von Datenbanken aus Altsystemen	286
8.1.1	Umgang mit Primärschlüsseln	287
8.1.2	Beliebige Join-Bedingungen mit Formeln	297
8.1.3	Zusammenführen beliebiger Tabellen	302
8.1.4	Die Arbeit mit Triggern.....	305

8.2	Anpassung von SQL.....	309
8.2.1	Eigene CRUD-Anweisungen schreiben.....	310
8.2.2	Integration von Stored Procedures und Functions	314
8.3	Verbesserung der Schema-DDL.....	321
8.3.1	Eigene Namen und Datentypen in SQL.....	321
8.3.2	Gewährleistung von Datenkonsistenz	323
8.3.3	Einfügen von Domain- und Spalten-Constraints.....	325
8.3.4	Constraints auf Tabellenebene.....	326
8.3.5	Datenbank-Constraints.....	329
8.3.6	Erstellung von Indizes	330
8.3.7	Einfügen zusätzlicher DDL.....	331
8.4	Zusammenfassung	333
Teil 3: Dialogorientierte Objektverarbeitung		335
9	Die Arbeit mit Objekten	337
9.1	Der Persistenz-Lebenszyklus.....	338
9.1.1	Objekt-Zustände	338
9.1.2	Der Persistenzkontext	341
9.2	Objektidentität und Objektgleichheit.....	344
9.2.1	Die Konversationen	345
9.2.2	Der Geltungsbereich der Objektidentität.....	346
9.2.3	Die Identität von detached Objekten.....	347
9.2.4	Erweiterung eines Persistenzkontexts	353
9.3	Die Hibernate-Interfaces.....	353
9.3.1	Speichern und Laden von Objekten	354
9.3.2	Die Arbeit mit detached Objekten	360
9.3.3	Management des Persistenzkontexts.....	365
9.4	Die Java Persistence API.....	368
9.4.1	Speichern und Laden von Objekten	368
9.4.2	Die Arbeit mit detached Entity-Instanzen.....	373
9.5	Java Persistence in EJB-Komponenten.....	376
9.5.1	EntityManager injizieren	377
9.5.2	Lookup eines EntityManagers	379
9.5.3	Zugriff auf eine EntityManagerFactory	379
9.6	Zusammenfassung	381
10	Transaktionen und gleichzeitiger Datenzugriff	383
10.1	Einführung in Transaktionen	383
10.1.1	Datenbank und Systemtransaktionen	385
10.1.2	Transaktionen in einer Hibernate-Applikation.....	387
10.1.3	Transaktionen mit Java Persistence	397
10.2	Steuerung des zeitgleichen Zugriffs	400
10.2.1	Zeitgleicher Zugriff auf Datenbanklevel.....	401
10.2.2	Optimistische Steuerung des zeitgleichen Zugriffs.....	406
10.2.3	Zusätzliche Isolationsgarantien.....	412

10.3	Nicht-transaktionaler Datenzugriff.....	416
10.3.1	Entlarvte Mythen über Autocommit	416
10.3.2	Die nicht-transaktionale Arbeit mit Hibernate.....	418
10.3.3	Optionale Transaktionen mit JTA.....	419
10.4	Zusammenfassung.....	421
11	Konversationen implementieren	423
11.1	Propagation der Hibernate-Session	424
11.1.1	Der Anwendungsfall für die Session-Propagation	424
11.1.2	Thread-local-Propagation	426
11.1.3	Propagation mit JTA.....	427
11.1.4	Propagation mit EJBs	429
11.2	Konversationen mit Hibernate.....	430
11.2.1	Die Garantien einer Konversation	430
11.2.2	Konversationen mit detached Objekten	431
11.2.3	Erweitern einer Session für eine Konversation	434
11.3	Konversationen mit JPA.....	440
11.3.1	Kontextpropagation in Java SE.....	441
11.3.2	Merging von detached Objekten in Konversationen.....	443
11.3.3	Erweiterung des Persistenzkontexts in Java SE.....	444
11.4	Konversationen mit EJB 3.0.....	448
11.4.1	Kontextpropagation mit EJBs.....	448
11.4.2	Erweiterter Persistenzkontext mit EJBs.....	451
11.5	Zusammenfassung.....	456
12	Effiziente Bearbeitung von Objekten	459
12.1	Transitive Persistenz	460
12.1.1	Persistence by Reachability	460
12.1.2	Kaskadierung auf Assoziationen anwenden	461
12.1.3	Die Arbeit mit dem transitiven Zustand.....	465
12.1.4	Transitive Assoziationen mit JPA.....	471
12.2	Bulk- und Batch-Operationen	473
12.2.1	Bulk-Anweisungen mit HQL und JPA QL.....	473
12.2.2	Batch-Verarbeitung	476
12.2.3	Die Arbeit mit einer stateless Session.....	478
12.3	Datenfilterung und Interception	480
12.3.1	Dynamische Datenfilter	481
12.3.2	Abfangen von Events in Hibernate	485
12.3.3	Das Core-Event-System	491
12.3.4	Entity-Listener und Callbacks	493
12.4	Zusammenfassung.....	495
13	Fetching und Caching optimieren.....	497
13.1	Definition des globalen Fetch-Plans.....	497
13.1.1	Optionen für das Auslesen der Objekte	497
13.1.2	Der Fetch-Plan: Default und Lazy	501

13.1.3	Die Arbeit mit Proxies	501
13.1.4	Deaktivieren der Proxy-Generierung	504
13.1.5	Eager Loading von Assoziationen und Collections	505
13.1.6	Lazy Loading mit Interception.....	507
13.2	Wahl einer Fetching-Strategie	509
13.2.1	Prefetching von Daten in Batches.....	510
13.2.2	Collections mit Subselects prefetchen	513
13.2.3	Eager Fetching mit Joins.....	514
13.2.4	Optimieren des Fetchings für Sekundärtabellen	516
13.2.5	Leitfaden zur Optimierung.....	519
13.3	Grundlagen des Caching.....	526
13.3.1	Geltungsbereiche und Strategien für das Caching	527
13.3.2	Die Cache-Architektur von Hibernate	531
13.4	Caching in der Praxis.....	535
13.4.1	Wahl einer Strategie für die Concurrency-Steuerung	535
13.4.2	Die Arbeit mit Cache-Bereichen.....	537
13.4.3	Einrichten eines lokalen Cache-Providers	538
13.4.4	Einrichten eines replizierten Caches	539
13.4.5	Steuerung des Second-level-Caches	543
13.5	Zusammenfassung	545
14	Abfragen mit HQL und JPA QL.....	547
14.1	Erstellen und Starten und Abfragen.....	548
14.1.1	Vorbereiten einer Abfrage	548
14.1.2	Ausführen einer Abfrage	557
14.1.3	Die Arbeit mit benannten Abfragen.....	561
14.2	HQL- und JPA QL-Abfragen	564
14.2.1	Selektion	564
14.2.2	Restriktion	566
14.2.3	Projektion	571
14.3	Joins, Reporting-Abfragen und Subselects.....	573
14.3.1	Zusammenführen von Relationen und Assoziationen.....	574
14.3.2	Reporting-Abfragen.....	584
14.3.3	Die Arbeit mit Subselects	589
14.4	Zusammenfassung	591
15	Fortgeschrittene Abfrageoptionen.....	593
15.1	Abfragen mit Criteria und Example.....	594
15.1.1	Grundlegende Abfragen mit Kriterien	594
15.1.2	Joins und dynamisches Fetching.....	599
15.1.3	Projektion und Berichtsabfragen.....	604
15.1.4	Query by Example	607
15.2	Native SQL-Abfragen	610
15.2.1	Automatischer Umgang mit dem Resultset.....	610
15.2.2	Auslesen skalarer Werte	611
15.2.3	Natives SQL in Java Persistence.....	613

15.3	Filtern von Collections	615
15.4	Caching der Abfrageergebnisse.....	617
15.4.1	Aktivieren des Caches für das Abfrageergebnis	618
15.4.2	Funktionsweise des Abfrage-Caches	618
15.4.3	Wann sollte der Abfrage-Cache benutzt werden?.....	619
15.4.4	Cache-Lookups von natürlichen Identifikatoren.....	620
15.5	Zusammenfassung.....	622
16	Erstellen und Testen von mehrschichtigen Applikationen.....	623
16.1	Hibernate in einer Webapplikation.....	624
16.1.1	Der Use Case für eine mehrschichtige Applikation	624
16.1.2	Einen Controller schreiben	624
16.1.3	Das Entwurfsmuster Open Session in View	626
16.1.4	Design von smarten Domain-Modellen	630
16.2	Eine Persistenzschicht erstellen.....	632
16.2.1	Ein generisches Muster für das Data Access Object.....	633
16.2.2	Das generische CRUD-Interface implementieren.....	635
16.2.3	Implementierung von Entity-DAOs.....	637
16.2.4	Die Verwendung von Data Access Objects	638
16.3	Das Command-Muster	642
16.3.1	Die grundlegenden Interfaces	642
16.3.2	Ausführung von Command-Objekten.....	644
16.3.3	Varianten des Command-Musters	646
16.4	Das Design einer Applikation mit EJB 3.0.....	648
16.4.1	Mit stateful Beans eine Konversation implementieren	648
16.4.2	DAOs mit EJBs schreiben	650
16.4.3	Einsatz der Abhängigkeitsinjektion	651
16.5	Testen	653
16.5.1	Die verschiedenen Testarten.....	653
16.5.2	Die Arbeit mit TestNG	654
16.5.3	Die Persistenzschicht testen.....	658
16.5.4	Ein paar Überlegungen zu Performance-Benchmarks	665
16.6	Zusammenfassung.....	667
	Anhang A: SQL-Grundbegriffe	669
	Anhang B: Mapping-Schnellreferenz	673
	Quellenangaben	675
	Register	677