



Igor Drobiazko

Tapestry 5

Die Entwicklung von Webanwendungen
mit Leichtigkeit



ADDISON-WESLEY



Inhaltsverzeichnis

Einleitung	17
Vorwort von Howard M. Lewis Ship	21
Danksagungen	23
Teil I Tapestry für Einsteiger	25
1 Von Servlets bis MVC-Frameworks	27
1.1 Wie alles begann	27
1.2 Model View Controller	30
1.3 Tapestry und das Model View Controller Pattern	31
1.4 Zusammenfassung	32
2 Tapestry-Schnelleinstieg	35
2.1 Starten von Tapestry-Anwendungen mit Eclipse Web Tools Platform ..	35
2.2 Starten von Tapestry-Anwendungen mit Run-Jetty-Run	40
2.3 Struktur von Tapestry-Anwendungen	42
2.3.1 Tapestry-Seiten	42
2.3.2 Tapestry Markup Language	44
2.3.3 Tapestry-Komponenten	50
2.3.4 Tapestry IoC	56
2.4 Nonstop-Entwicklung mit Tapestry	57
2.5 Fehlerberichte	58
2.6 Zusammenfassung	59
3 Tapestry als ereignisgetriebenes MVC-Framework	61
3.1 Behandeln der Benutzeraktionen	61
3.2 Namenskonvention vs. Annotationen	63
3.3 Kontext eines Ereignisses	64
3.4 Erzeugen einer Antwort mit Handler-Methoden	65
3.5 Auslösen von Ereignissen mit der Komponente EventLink	67
3.6 Programmatisches Auslösen eigener Ereignisse	68
3.7 Event Bubbling	70
3.8 Abfangen von Exceptions aus Handler-Methoden	70
3.9 Zusammenfassung	71

4	Navigation zwischen Seiten	73
4.1	Erstellen von Verweisen zwischen Seiten	73
4.2	Übermitteln von Informationen an eine Zielseite	74
4.3	Übermitteln von Informationen an eine Tapestry-Seite	76
4.3.1	Generieren von Handler-Methoden für den Aktivierungskontext	81
4.4	Navigation durch Aktionen	82
4.5	Zusammenfassung	86
5	Entwicklung von zustandsbehafteten Anwendungen	87
5.1	Lebenszyklus eines Servlets	87
5.2	Lebenszyklus einer JavaServer Page	88
5.3	Lebenszyklus einer Tapestry-Seite	88
5.3.1	Teilnehmen am Lebenszyklus einer Tapestry-Seite	89
5.3.2	Wann werden <code>pageLoaded()</code> , <code>pageAttached()</code> und <code>pageDetached()</code> aufgerufen?	92
5.4	Verwalten des Seitenzustandes	94
5.4.1	Strategien zum Zwischenspeichern der Eigenschaften	95
5.4.2	Verwerfen des Zustands einer Seite	97
5.4.3	Unterbinden des Verwerfens des Seitenzustands	98
5.5	Verwaltung des Anwendungszustandes	98
5.5.1	Erzeugen eines Session State Objects	99
5.5.2	Überprüfung der Existenz eines Session State Objects	100
5.5.3	Arbeiten mit dem Dienst <code>ApplicationStateManager</code>	101
5.5.4	Eingreifen in den Prozess der Instanziierung eines Session State Objects	102
5.6	Zusammenfassung	103
6	Lokalisierung	105
6.1	Internationalisierung und Lokalisierung in Java	105
6.2	Anwendungsweiter Nachrichtenkatalog	106
6.3	Komponenten-Nachrichtenkatalog	106
6.4	Lokalisierte Templates	106
6.5	Zugreifen auf den Nachrichtenkatalog	107
6.6	Unterstützte Sprachen	109
6.7	Lokalisierung statischer Ressourcen	109
6.8	Umschalten zwischen unterstützten Sprachen einer Anwendung	112
6.9	Zusammenfassung	115

7	Formulare	117
7.1	Erzeugen eines einfachen Login-Formulars	118
7.2	Behandlung der Ereignisse der Komponente Form	121
7.3	Mehrere Formulare auf einer Seite	125
7.4	Überblick über Tapestrys Formularkomponenten	126
7.4.1	Texteingabe in Text- und Passwortfelder	126
7.4.2	Checkboxen	129
7.4.3	Radiobuttons	131
7.4.4	Auswahllisten	134
7.4.5	Palette	139
7.4.6	Eingabe eines Datums	141
7.4.7	Hochladen von Dateien	143
7.4.8	Abschicken von Formularen	146
7.5	Labels für Formularfelder	148
7.6	Eingabevalidierung	150
7.6.1	Überschreiben von Validierungsmeldungen	154
7.6.2	Eigene Validatoren	155
7.7	Null-Werte in Formularfeldern	160
7.8	Umwandlung von Eingaben zwischen Client und Server	162
7.9	Zusammenfassung	167
8	Dynamische Formulare	169
8.1	Variable Anzahl von Formularfeldern	169
8.1.1	Beispielszenario	170
8.1.2	Implementierung des Beispielszenarios	171
8.2	Erweitern der Formulare durch Benutzerinteraktionen	176
8.2.1	Beispielszenario	176
8.2.2	Implementierung des Beispielszenarios	176
8.3	Teilformulare	179
8.4	Zusammenfassung	182
9	Arbeiten mit JavaBeans	183
9.1	Formulare für JavaBeans	183
9.1.1	Don't Repeat Yourself	183
9.1.2	Generieren von Formularen mit BeanEditForm	185
9.1.3	Verstecken bestimmter Eigenschaften von JavaBeans	187

9.1.4	Angepasste Formularfelder	189
9.1.5	Reihenfolge der Formularfelder	191
9.1.6	Virtuelle Eigenschaften	192
9.1.7	Programmatisches Ändern von Metadaten einer JavaBean	192
9.1.8	Eingabvalidierung	194
9.2	Darstellen von JavaBeans	196
9.3	Darstellen mehrerer JavaBeans	197
9.3.1	Paging	200
9.3.2	Zugriff auf die Werte der aktuellen Iteration	201
9.3.3	Überschreiben der Darstellung von Spalten	202
9.4	Datentypen von Eigenschaften	203
9.4.1	Neue Datentypen zum Editieren von Eigenschaften	203
9.4.2	Neue Datentypen zum Darstellen von Eigenschaften	205
9.5	Zusammenfassung	207
10	Multimedia-Inhalte	209
10.1	Senden eines einfachen Textes als Bytestrom	209
10.2	Anbieten einer Download-Funktionalität	210
10.3	Anzeigen eines Diagramms	212
10.4	Zusammenfassung	215
Teil II	Tapestry für Fortgeschrittene	217
11	Entwicklung wiederverwendbarer Komponenten	219
11.1	Bestandteile einer Komponente	219
11.2	Parameter von Komponenten	219
11.3	Bidirektionale Parameter	221
11.4	Standard-Bindings der Parameter	222
11.4.1	Standard-Binding-Methoden	224
11.4.2	Generierung von Standard-Binding-Methoden	225
11.5	Vererbung von Bindings	226
11.6	Eigene Binding-Präfixe	228
11.7	Markup-Erzeugung einer Komponente	232
11.8	Informelle Parameter	236
11.9	Vererben von Templates	237
11.10	Umgebungsdienste	239

11.11	Komponentenbibliotheken	241
11.12	Zusammenfassung	243
12	Mixins	245
12.1	Was sind Mixins?	245
12.2	Instanz-Mixins	249
12.3	Implementierungs-Mixins	251
12.4	Parameter von Mixins	252
12.5	Mixins in Tapestry	253
12.6	Zusammenfassung	254
13	Ajax	255
13.1	Einfache Ajax-Funktionalität	255
13.2	Aktualisieren von Zones mit zusätzlichem Inhalt	258
13.3	Visuelle Rückmeldung über Ajax-Updates	261
13.4	Aktualisierung mehrerer Zones	261
13.5	Ajax-Unterstützung in Tapestry-Komponenten	264
13.6	Ajax und JavaScript	265
13.7	Optimierung der Ladezeit durch Ajax	266
13.8	Erzeugen eines Autovervollständigers	269
13.8.1	Erweitern von Autocomplete	271
13.9	Eigene Ajax-Komponenten	275
13.10	Zusammenfassung	278
14	Integration von Hibernate und Spring	281
14.1	Integration von Hibernate	281
14.1.1	Konfiguration von Hibernate in Tapestry-Anwendungen	281
14.1.2	Zugriff auf die Hibernate-Session	282
14.1.3	Commit von Transaktionen	283
14.1.4	Hibernate-Entitäten als Aktivierungskontext	284
14.1.5	Zwischenspeichern von Entitäten mit @Persist	284
14.1.6	Darstellen der Hibernate-Entitäten mit Komponente Grid	285
14.1.7	Zugriff auf Hibernate-Metriken	286
14.1.8	Suche nach Hibernate-Entitäten	287
14.1.9	Teilnahme an der Hibernate-Konfiguration	288
14.1.10	Konfiguration der Hibernate-Integrationsbibliothek	289

14.2	Integration von Spring	289
14.2.1	Konfiguration von Spring	290
14.2.2	Benutzen von Spring-Beans in Tapestry-Anwendungen	292
14.2.3	Benutzung von Tapestry-Diensten in Spring-Beans	294
14.2.4	Benutzung von Spring-Beans in Seiten und Komponenten	296
14.3	Zusammenfassung	296
15	Testen von Tapestry-Anwendungen	299
15.1	Unit-Tests für Dienste	299
15.2	Unit-Tests für Seiten und Komponenten	302
15.2.1	Einfacher Unit-Test für eine Seite	302
15.2.2	Unit-Tests zur Überprüfung von Markup	304
15.2.3	Testen von Links	305
15.2.4	Testen von Formularen	306
15.2.5	Überschreiben von Diensten zur Verbesserung der Testbarkeit	309
15.2.6	Testen von Komponenten	311
15.3	Integrationstests	312
15.3.1	Ein einfacher Integrationstest	312
15.3.2	Klicken auf Links in Integrationstests	313
15.3.3	Testen von Formularen in Integrationstests	314
15.3.4	Testen von Ajax-Funktionalität in Integrationstests	316
15.4	Zusammenfassung	318
16	Sicherheit	319
16.1	Schützen der Seiten durch HTTPS	319
16.1.1	Basis-URLs	320
16.1.2	HTTPS und Entwicklungsmodus	322
16.2	Authentifizierung	322
16.3	Zusammenfassung	324
Teil III	Tapestry für Profis	325
17	Tapestry IoC und Dependency Injection	327
17.1	Einführung	327
17.2	Inversion of Control	327
17.3	Dependency Injection	328
17.3.1	Setter-Injektion	328
17.3.2	Konstruktor-Injektion	329

17.3.3	Feld-Injektion	329
17.3.4	Singleton-Antipattern	330
17.4	Warum braucht Tapestry IoC?	330
17.5	Überblick über Tapestry IoC	331
17.5.1	Modul	332
17.5.2	Bind-Methoden	332
17.5.3	Build-Methoden	334
17.5.4	Abhängigkeiten von Diensten	334
17.6	Lebenszyklus eines Dienstes	336
17.7	Eifriges Laden von Diensten	337
17.8	Scope eines Dienstes	338
17.9	Eindeutigkeit von Diensten	339
17.9.1	Eindeutigkeit durch explizite IDs	339
17.9.2	Eindeutigkeit durch Marker-Annotationen	341
17.10	Konfiguration von Diensten	343
17.10.1	Erweiterungspunkte in Tapestry	344
17.10.2	Ungeordnete Konfigurationen	345
17.10.3	Geordnete Konfigurationen	347
17.10.4	Schlüssel-Wert-Paare als Konfiguration	349
17.11	Konfiguration von Anwendungen mit Symbolen	350
17.11.1	Symbole	350
17.11.2	Bereitstellen von Symbolen	352
17.12	Überschreiben von Diensten	357
17.13	Diensterzeugung für Fortgeschrittene	357
17.13.1	Implementieren von Chains	357
17.13.2	Implementieren von Pipelines	359
17.14	Zusammenfassung	362
18	AOP mit Tapestry	365
18.1	Grundbegriffe von AOP	365
18.2	Tapestry und AOP	366
18.3	Autorisierung der Methodenaufrufe von Diensten	368
18.4	Überschreiben von Methodenparametern der Dienste	369
18.5	Überschreiben der Rückgabewerte von Dienstmethoden	370
18.6	Dekorieren von Komponentenmethoden	371
18.7	Zusammenfassung	372

19	Typumwandlung/Type Coercion	375
19.1	Verstehen der Typumwandlung	375
19.2	Erweitern des Umwandlungsgraphen	376
19.3	Durchführen der Typumwandlung	377
19.4	Zusammenfassung	379
20	Verarbeitung von Anfragen durch Tapestry	381
20.1	Die HttpServletRequestHandler-Pipeline	381
20.1.1	Bereitstellen eines eigenen HttpServletRequestFilters	383
20.2	Die RequestHandler-Pipeline	384
20.2.1	Bereitstellen eines eigenen RequestHandlers	384
20.3	Chain of Command des Dienstes Dispatcher	385
20.3.1	Behandlung von Anfragen an den Kontextpfad	385
20.3.2	Behandlung von Anfragen an Tapestry-Seiten	386
20.3.3	Auslösen von Komponentenergebnissen	386
20.3.4	Behandlung von Anfragen an statische Ressourcen	387
20.3.5	Bereitstellen eines eigenen Dispatchers	387
20.4	Zusammenfassung	387
21	Bytecodemanipulation mit Tapestry	389
21.1	Das zu implementierende Szenario	389
21.2	Bytecodemanipulation von Seitenklassen	390
21.3	Zusammenfassung	395
A	Anhang	397
A.1	Komponenten	397
A.1.1	ActionLink	397
A.1.2	AddRowLink	397
A.1.3	AjaxFormLoop	398
A.1.4	Any	399
A.1.5	BeanDisplay	399
A.1.6	BeanEditForm	400
A.1.7	BeanEditor	401
A.1.8	Checkbox	402
A.1.9	DateField	403
A.1.10	Delegate	404
A.1.11	Errors	404
A.1.12	EventLink	404

A.1.13	ExceptionDisplay	405
A.1.14	Form	405
A.1.15	FormFragment	406
A.1.16	FormInjector	407
A.1.17	Grid	408
A.1.18	Hidden	410
A.1.19	If	410
A.1.20	Label	411
A.1.21	LinkSubmit	411
A.1.22	Loop	412
A.1.23	Output	413
A.1.24	OutputRaw	413
A.1.25	PageLink	414
A.1.26	Palette	414
A.1.27	PasswordField	415
A.1.28	ProgressiveDisplay	416
A.1.29	Radio	416
A.1.30	RadioGroup	417
A.1.31	RemoveRowLink	417
A.1.32	RenderObject	417
A.1.33	Select	418
A.1.34	Submit	419
A.1.35	SubmitNotifier	419
A.1.36	TextArea	419
A.1.37	TextField	420
A.1.38	TextOutput	421
A.1.39	Unless	421
A.1.40	Upload	421
A.1.41	Zone	422
A.2	Mixins	422
A.2.1	Autocomplete	422
A.2.2	DiscardBody	423
A.2.3	NotEmpty	423
A.2.4	RenderDisabled	423
A.2.5	RenderInformals	423
A.2.6	TriggerFragment	424

A.3	Seiten	424
A.3.1	PropertyDisplayBlocks	424
A.3.2	PropertyEditBlocks	424
A.3.3	ServiceStatus	424
A.3.4	Statistics	424
	Stichwortverzeichnis	425