

Inhalt

Vorwort zur 3. Auflage	XIII
Vorwort	XV
1 Einleitung	1
1.1 Ziel dieses Buchs	1
1.2 Was sind JavaServer Faces?	2
1.3 Der Leser	4
1.4 Das Buch im Netz	5
1.5 Versionen, Versionen, Versionen	7
1.6 Spezifikationen, Implementierungen, Systeme	7
1.7 Totgesagte leben länger	8
1.8 Aufbau des Buchs	10
1.9 Classic Models	11
2 JavaServer Faces im Detail – die Grundlagen	15
2.1 Bearbeitungsmodell einer JSF-Anfrage	16
2.1.1 Wiederherstellung des Komponentenbaums	18
2.1.2 Übernahme der Anfragewerte	19
2.1.3 Validierung	20
2.1.4 Aktualisierung der Modellobjekte	21
2.1.5 Aufruf der Anwendungslogik	22
2.1.6 Rendern der Antwort	22
2.2 Expression-Language	25
2.2.1 Syntax	25
2.2.2 Werteausdrücke	26
2.2.3 Vergleiche, arithmetische und logische Ausdrücke	29
2.2.4 Methodenausdrücke	31

2.2.5	Vordefinierte Objektnamen	32
2.2.6	Collections	36
2.2.7	Lambdas	37
2.2.8	Konstanten	39
2.2.9	Komponentenbindungen	40
2.2.10	Verwendung der Expression-Language in Java	41
2.2.11	Weniger sinnvolle Verwendungen	41
2.3	Managed Beans	42
2.3.1	Architekturfragen und Namenskonventionen	43
2.3.2	Context and Dependency Injection	44
2.3.3	Architekturfragen zum Zweiten	48
2.3.4	Initialisierung	48
2.4	Validierung und Konvertierung	49
2.4.1	Standardkonvertierer	50
2.4.2	Konvertierung und Formatierung von Zahlen	53
2.4.3	Konvertierung und Formatierung von Kalenderdaten und Uhrzeiten ...	54
2.4.4	Konvertierung von Aufzählungstypen	56
2.4.5	Anwendungsdefinierte Konvertierer	60
2.4.6	Standardvalidierer	63
2.4.7	Validierungsmethoden	65
2.4.8	Anwendungsdefinierte Validierer	67
2.4.9	Eingabekomponenten und das <code>immediate</code> -Attribut	68
2.4.10	Bean-Validierung mit JSR 380	71
2.4.11	Anwendungsdefinierte Constraints mit Bean Validation	75
2.4.12	Gruppenvalidierung mit Bean Validation	77
2.4.13	Validierung auf Klassenebene	78
2.4.14	Fehlermeldungen	81
2.4.15	BV-Fehlermeldungen	89
2.5	Event-Verarbeitung	90
2.5.1	JSF-Events und allgemeine Event-Verarbeitung	91
2.5.2	Action-Events	92
2.5.3	Action-Events und Navigation	98
2.5.4	Befehlskomponenten mit Parametern	102
2.5.5	Befehlskomponenten und das <code>immediate</code> -Attribut	107
2.5.6	Value-Change-Events	108
2.5.7	Data-Model-Events	111
2.5.8	Phase-Events	115
2.5.9	System-Events	117

2.6	HTML5	120
2.6.1	Pass-Through-Attribute	121
2.6.2	Pass-Through-Elemente	122
2.7	Ajax	124
2.7.1	Das <code><f:ajax></code> -Tag	125
2.7.2	Komponentengruppen und Ajax	128
2.7.3	Komponentenabhängigkeiten	129
2.7.4	Validierung	131
3	Contexts and Dependency Injection	133
3.1	Beans, Scopes und Kontexte	135
3.1.1	Scopes	135
3.1.2	JSFs <code>@ViewScoped</code>	136
3.1.3	Kontexte	136
3.1.4	Ein genauerer Blick auf <code>@Inject</code>	140
3.1.5	Bean-Manager und programmatischer Zugriff auf Bean-Instanzen	144
3.2	Mehr Flexibilität mit Qualifiern und Alternativen	145
3.2.1	Qualifier	145
3.2.2	Vordefinierte Qualifier	148
3.2.3	Alternativen	149
3.2.4	Der Deskriptor <code>beans.xml</code>	151
3.3	Producer und Disposer	153
3.3.1	Producer-Methoden	153
3.3.2	Scopes	154
3.3.3	Producer-Field	155
3.3.4	Disposer-Methoden	155
3.4	Der Conversation-Scope	157
3.5	Events	161
3.5.1	Einfache Event-Producer und Observer	162
3.5.2	Spezialisierung durch Qualifier	164
3.5.3	Event-Metadaten	165
3.5.4	Events und Transaktionen	166
3.6	Interceptoren	167
3.7	Und was es sonst noch gibt	169
4	Weiterführende Themen	173
4.1	Templating	173
4.1.1	Der grundlegende Template-Mechanismus	173
4.1.2	Ein realistischeres Beispiel: unsere Projekte	175

4.1.3	Dynamische Templates	177
4.2	Internationalisierung und Lokalisierung	181
4.2.1	Lokalisierung	182
4.2.2	Dynamische und explizite Lokalisierung	187
4.2.3	Klassen als Resource-Bundles	190
4.2.4	Managed Beans und Lokalisierung	193
4.2.5	Resource-Bundles mit UTF-8-Codierung.....	194
4.2.6	Lokalisierte BV-Fehlermeldungen	194
4.3	Komponenten- und Client-Ids	196
4.3.1	Id-Arten und Namensräume	196
4.3.2	Client- und server-seitige Programmierung mit Ids.....	199
4.4	Verwendung allgemeiner Ressourcen	203
4.4.1	Einfache Ressourcen.....	203
4.4.2	Versionierte Ressourcen und Ressourcen-Bibliotheken	205
4.4.3	Positionierung von Ressourcen	206
4.4.4	Kombination von CSS- und Grafikressourcen	208
4.5	Ajax zum Zweiten	209
4.5.1	JSFs JavaScript-Bibliothek	210
4.5.2	Navigation.....	212
4.5.3	JavaScript mit Java	214
4.5.4	Nicht gerenderte Komponenten	215
4.5.5	Komponentenbibliotheken.....	217
4.5.6	Ajax ohne <code><f:ajax></code>	219
4.5.7	Das <code><h:commandScript></code> -Tag	223
4.5.8	Zu schnelle Benutzer ;-))	225
4.5.9	JavaScript und Expression-Language kombinieren	226
4.6	GET-Anfragen und der Flash-Scope.....	228
4.6.1	Einfache GET-Anfragen	228
4.6.2	View-Parameter	229
4.6.3	View-Actions.....	231
4.6.4	Der Flash-Scope	231
4.7	Zusammengesetzte Komponenten.....	235
4.7.1	Schnittstelle und Implementierung	236
4.7.2	Weitere Möglichkeiten.....	239
4.7.3	Packaging und Wiederverwendung	241
4.8	UI-Komponenten	242
4.8.1	Die Standardkomponenten.....	243
4.8.2	Render-Sätze.....	246

4.8.3	Die JSF-Standard-Bibliotheken	246
4.8.4	Die HTML-Bibliothek.....	247
4.8.5	Die Kernbibliothek.....	250
4.8.6	Die Templating-Bibliothek (Facelets)	252
4.8.7	Die Composite-Component-Bibliothek	252
4.8.8	Die JSTL-Kern- und Funktionsbibliothek.....	252
4.8.9	Komponentenbindungen	253
4.9	Die Servlet-Konfiguration	254
4.9.1	Der Deployment-Deskriptor.....	255
4.9.2	Übersicht Kontextparameter	257
4.9.3	Zustandsspeicherung.....	259
4.9.4	Konfigurationsdateien.....	260
4.9.5	Projektphasen	261
4.9.6	Zugriff auf Konfigurationsdaten	261
5	JavaServer Faces im Einsatz: Classic Models	265
5.1	Datenzugriff und Datenmanipulation	265
5.1.1	Java Persistence API.....	266
5.1.2	Enterprise JavaBeans	270
5.1.3	Transaktionen mit JTA	274
5.1.4	Data-Sources und Persistence-Units	275
5.2	JSF im Einsatz	276
5.2.1	Übersichten	276
5.2.2	Master-Detail-Pattern	283
5.2.3	Dynamische Drop-down-Listen	290
5.2.4	Dynamische UIs.....	292
5.3	Authentifizierung und Autorisierung	299
5.3.1	Zugriffschutz für Ressourcen	299
5.3.2	Identity Store.....	300
5.3.3	Authentifizierungs- und rollenbasierte JSF-Seiten	306
5.4	Datenexport im PDF- und Excel-Format	309
5.4.1	PDF-Erzeugung.....	309
5.4.2	Excel-Erzeugung	312
5.5	Testen.....	313
5.5.1	Arquillian	313
5.5.2	Drone und Graphene	315
5.5.3	Selenium	319
5.6	H2-Web-Konsole.....	321

6	Spezialthemen	323
6.1	Die JSF-Konfiguration	323
6.1.1	XML-Konfigurationsdatei versus Annotationen	330
6.1.2	JSF erweitern: ein eigener Exception-Handler	330
6.1.3	Programmative Konfiguration	332
6.2	Web-Sockets	333
6.2.1	Das Java-API	333
6.2.2	Globaler Server-Push	335
6.2.3	Dedizierte Nachrichten an Clients	338
6.2.4	Zeitaufwendige Berechnungen	341
6.2.5	Konfiguration	343
6.3	Resource-Library-Contracts	344
6.3.1	Globales Mapping von Contracts	345
6.3.2	Contracts auf View-Ebene	349
6.3.3	Programmative Konfiguration	350
6.4	Faces-Flows	353
6.5	Native Komponenten	358
6.5.1	Der einfache Einstieg	359
6.5.2	Komponententyp, Komponentenfamilie und Renderer-Typ	360
6.5.3	Renderer	363
6.5.4	Die Zukunft: Web-Komponenten	366
6.6	JSF als zustandsbehaftetes Komponenten-Framework	373
6.7	Mobile Endgeräte	377
6.7.1	Seiteninhalte für unterschiedliche Bildschirmgrößen und Auflösungen	377
6.7.2	Progressive Web Apps und Web-App-Manifeste	379
6.8	HTTP/2 Server-Push	384
6.9	Single-Page-Applications	386
7	Verwendete Systeme	389
7.1	WildFly und JBoss EAP	390
7.2	Payara	391
7.3	TomEE	392
7.4	WildFly Bootable JAR	393
7.5	Payara Micro	394
8	Ausblick	397
8.1	Wie geht es weiter mit JSF?	397
8.2	Andere JSF-Bücher	398

A	Die Tags der Standardbibliotheken	401
A.1	HTML-Tag-Bibliothek.....	401
A.2	Kernbibliothek	408
A.3	Templating-Bibliothek (Facelets)	412
A.4	Composite-Component-Bibliothek.....	414
A.5	JSTL-Kernbibliothek.....	416
A.6	JSTL-Funktionsbibliothek	417
A.7	Pass-Through-Attribute und -Elemente	419
B	URL-Verzeichnis	421
	Literatur	427
	Stichwortverzeichnis	429