

HANSER



Inhaltsverzeichnis

Bernd Weber, Patrick Baumgartner, Oliver Braun

OSGi für Praktiker

Prinzipien, Werkzeuge und praktische Anleitungen auf dem Weg zur
"kleinen SOA"

ISBN: 978-3-446-42094-6

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-42094-6>

sowie im Buchhandel.

Inhaltsverzeichnis

Foreword	XI
Vorwort der Autoren	XIII
Die Autoren	XV
1 Einleitung	1
1.1 Die OSGi-Service-Plattform	2
1.1.1 Bundles	2
1.1.2 Das OSGi-Framework	4
1.2 Geschichte	18
1.3 OSGi Enterprise Design	19
1.3.1 Distributed OSGi	20
1.3.2 JMX-Management	22
1.4 OSGi vs. SOA	23
1.5 Über dieses Buch	24
2 OSGi-Implementierungen	29
2.1 Die OSGi Alliance	30
2.1.1 Vollmitglieder	30
2.1.2 Assoziierte Unternehmen	31
2.1.3 Unterstützer	31
2.2 Zertifizierte Produkte	31
2.3 Open Source-Implementierungen	32
2.3.1 Eclipse Equinox	32
2.3.2 Knopflerfish	35
2.3.3 Apache Felix	37
2.4 OSGi und das Spring-Framework	40
2.5 Fazit	42
3 JobTimer – Bundle-Grundlagen	45
3.1 Voraussetzungen	46

3.1.1	Projekt aufsetzen mit Maven	46
3.1.2	Verwenden als Eclipse-Projekt	48
3.1.3	POM-Datei anpassen	49
3.2	JobTimer implementieren	51
3.3	Verpacken als Bundle	54
3.3.1	Verwendung von bnd wrap	55
3.3.2	Verwendung von bnd build	56
3.4	Bundles installieren und starten	59
3.4.1	Starten der Plattform	59
3.4.2	Installieren von Bundles	60
3.4.3	Bundle Repositories	62
3.4.4	Starten von Bundles	63
3.4.5	Auflösen von Abhängigkeiten	63
3.4.6	Herunterfahren der Plattform	65
3.5	Fazit	65
4	GateKeeper – OSGi-Projekt aufsetzen	67
4.1	Voraussetzungen	67
4.1.1	PAX-Werkzeuge	68
4.1.2	Bundle-Automatik	68
4.1.3	Master POM erstellen	69
4.1.4	Bereitstellen der Bundles	71
4.1.5	Startreihenfolge	75
4.2	Erstellen des GateKeeper-Bundles	76
4.2.1	Erstellen des Maven-Moduls	76
4.2.2	Verwenden der Eclipse IDE	77
4.2.3	GateKeeper implementieren	79
4.2.4	Metadaten definieren	81
4.3	Fazit	82
5	Enunciator – Declarative Services	85
5.1	Voraussetzungen	85
5.1.1	Tücken optionaler Auflösung	86
5.2	Erstellen des Enunciator-Bundles	87
5.2.1	Erstellen des Maven-Moduls	87
5.2.2	Enunciator implementieren	88
5.2.3	Dienst deklarieren	91
5.2.4	Installieren des Bundles	93
5.3	Deklaration einer Komponente	94
5.3.1	Attribute einer Komponente	94
5.3.2	Kind-Elemente einer Komponente	95

5.4	GateKeeper erweitern	97
5.4.1	Zugriff auf das Enunciator-Package	97
5.4.2	Lifecycle-Methoden erweitern	98
5.4.3	Dynamischer Aufruf	99
5.4.4	Exkurs: ServiceTrackerCustomizer	100
5.4.5	Installieren des erweiterten Bundles	102
5.5	Ausführen der Bundles	103
5.5.1	Ausführen mit PAX Provision	103
5.5.2	Ausführen mit dem PAX Runner Plugin	104
5.6	Fazit	106
6	Migration – Java-Archive zu Bundles erweitern	107
6.1	Analyse des Java-Archivs	107
6.1.1	Ermitteln von Abhängigkeiten mit Hilfe von bnd	108
6.1.2	Analyse der Abhängigkeiten	109
6.1.3	Laufzeitumgebung	109
6.2	Standardmigration	111
6.2.1	Festlegen der Metadaten	111
6.2.2	Durchführen der Migration	113
6.2.3	Bereitstellung	114
6.3	Beistellungsmigration	114
6.3.1	Aufnahme von Ressourcen	115
6.3.2	Festlegen der Metadaten	117
6.3.3	Durchführen der Migration	117
6.3.4	Bereitstellung	118
6.4	Transitive Migration	119
6.4.1	Festlegen der Metadaten	119
6.4.2	Migration der BouncyCastle-Archive	120
6.4.3	Gemeinsame Migration aller BouncyCastle-Archive	122
6.4.4	Festlegen der Metadaten	123
6.4.5	Durchführen der Migration	123
6.4.6	Bereitstellung	124
6.5	Versionen und Versionsbereiche	124
6.5.1	Referenzierung über Versionen	125
6.5.2	Konventionen	125
6.6	Fazit	126
7	Worker – Fragmente und Integrationstests	127
7.1	Voraussetzungen	127
7.2	Die WorkerService-API	128
7.2.1	Schnittstellendefinition	128

7.2.2	POM-Datei	129
7.3	WorkerService implementieren	130
7.3.1	Vorüberlegungen	130
7.3.2	Der Quellcode	130
7.3.3	Komponente deklarieren	136
7.3.4	POM-Datei anpassen	137
7.4	GateKeeper erweitern	138
7.5	Laufzeitverhalten	141
7.5.1	Bundle stop	142
7.5.2	Bundle start	143
7.5.3	Bundle update	143
7.5.4	Bundle refresh	144
7.6	Fragmente	145
7.6.1	Verwendung im Projekt	145
7.6.2	Erstellung des Bundles	146
7.6.3	Festlegen der Metadaten	146
7.6.4	Test des Fragments	146
7.7	Integrationstest mit PAX Exam	147
7.7.1	Test-Bundle erstellen	148
7.7.2	POM-Datei anpassen	148
7.7.3	Generierter Integrationstest	151
7.7.4	Einfacher Integrationstest	151
7.7.5	Erweiterter Integrationstest	154
7.8	Fazit	160
8	SkilledWorker – Blueprint Services	161
8.1	Voraussetzungen	162
8.2	Erstellen des SkilledWorker-Bundles	162
8.2.1	Erstellen des Maven-Moduls	162
8.2.2	Verwenden der Eclipse IDE	163
8.2.3	SkilledWorker implementieren	163
8.3	Blueprint Services	169
8.4	Spring Dynamic Modules for OSGi	172
8.4.1	GateKeeper erweitern	175
8.4.2	Services filtern	176
8.5	Fazit	178
9	WebMonitor – Webapplikationen mit OSGi	181
9.1	Voraussetzungen	181
9.2	Erstellen des WebMonitor-Webbundles	182
9.2.1	Anpassen des POMs	182

9.2.2	Services und Controller bereitstellen	184
9.2.3	JSP View erzeugen	188
9.2.4	Manifest anpassen	189
9.2.5	OSGi-spezifisches web.xml	190
9.3	Fazit	192
10	Remote Services – Informationen über Distanz austauschen	193
10.1	Voraussetzungen	193
10.2	Erstellen des RemoteLogServer-Bundles	194
10.2.1	Erstellen des Maven-Moduls	194
10.2.2	Implementierung des RemoteLogServices	194
10.3	Bundles starten mit PAX Runner	196
10.3.1	Installation von PAX Runner	197
10.3.2	Starten des RemoteLogServer mit PAX Runner	197
10.4	Webservice testen	198
10.5	Remote Service konsumieren	199
10.6	Enunciator mit Fragment erweitern	199
10.7	GossipMonger starten	200
10.8	Fazit	203
11	Police – System-Monitoring im laufenden Betrieb	205
11.1	Installation der Apache Felix Web Console	205
11.2	Konfiguration der Web Console	208
11.2.1	Benutzer/Passwort ändern	208
11.2.2	HTTP-Ports anpassen	209
11.3	Fazit	210
12	Big Brother – Überwachung und Steuerung von außen	211
12.1	JobTimer-Intervall über JMX verändern	211
12.2	JobTimer-Intervall im Bundle-Cache persistieren	215
12.3	Fazit	218
13	Launcher – Starten von OSGi-Anwendungen	219
13.1	Framework Launching API	219
13.1.1	Schnittstellen	220
13.1.2	Framework Eigenschaften	220
13.2	Erstellen des Launchers	222
13.2.1	Maven-Projekt erstellen	222
13.2.2	Der Launcher im Quelltext	222
13.2.3	Framework starten	227
13.2.4	FrameworkFactory definieren	227
13.2.5	Framework konfigurieren	228

13.2.6	Infrastrukturmaßnahmen	228
13.2.7	Applikations-Bundles	233
13.3	Launcher bereitstellen	234
13.3.1	Abhängigkeiten kopieren	234
13.3.2	Manifest erweitern	235
13.3.3	Assembly erstellen	235
13.3.4	Launcher testen	236
13.3.5	Bereitgestellte ApplicationBundles	237
13.4	Fazit	237
14	Aliens – Bundles from Outer (Java) Space	239
14.1	Scala	239
14.2	ScalaModules	241
14.3	Das Chirp-Bundle	243
14.4	Fazit	248
	Stichwortverzeichnis	249