

Inhaltsverzeichnis

Geleitwort	1
Vorwort	3
Wege durch das Buch	9

I Einleitung **15**

1 Hallo, Spring Boot	17
1.1 Projektdefinition oder »Project Object Model«	17
1.2 Abhängigkeiten definieren	19
1.3 Hallo, Welt!	21
1.4 Liefern	24
1.5 Fehler erkennen	26

II Spring Boot **29**

2 Projektstruktur	31
2.1 Build-Management-Tools	31
2.1.1 Maven	32
2.1.2 Gradle	37
2.2 Pakete und Klassen strukturieren	39
2.3 Spring Initializr	41
2.4 IDE-Unterstützung	43
2.4.1 Spring Tool Suite	43
2.4.2 NetBeans IDE	47
2.4.3 IntelliJ IDEA	50
2.4.4 Die Qual der Wahl	52
3 Spring-Framework-Grundlagen	53
3.1 Dependency Injection	54
3.1.1 Der Spring-Container	54

3.1.2	Inversion of Control	56
3.1.3	Ausblick	60
3.2	Aspektorientierte Programmierung	60
3.2.1	Einführung	60
3.2.2	Spring und AOP	61
4	Konfiguration	65
4.1	Externe Konfiguration	65
4.1.1	Konfigurationsdateien	67
4.1.2	Zugriff auf Konfiguration	72
4.2	Interne Konfiguration	82
4.2.1	Profile	82
4.2.2	Konfigurationsklassen	84
4.2.3	Automatische Konfiguration	91
5	Die Magie hinter Spring Boot	95
5.1	Grundlagen	96
5.1.1	Namensgebung	97
5.1.2	Factories Loader	97
5.2	Konfiguration, nur unter Bedingung	98
5.2.1	Auf Anwesenheit von Klassen hin testen	99
5.2.2	Auf Anwesenheit von Beans hin testen	100
5.2.3	Die Konfiguration auswerten	101
5.2.4	Weitere Bedingungen nutzen	101
5.2.5	Bedingungen logisch verknüpfen	102
5.2.6	Metadaten bereitstellen	103
5.2.7	Ihren Starter deployen	103
5.3	Fazit	105
6	Logging	107
6.1	Gemeinsame Konfiguration	108
6.2	Erweiterte Konfiguration	110
6.2.1	Logback	111
6.2.2	Log4j 2	112
6.2.3	Java Util Logging (JUL)	114
6.3	Zugriffslogs	114
6.3.1	Tomcat	114
6.3.2	Undertow	115
6.4	Remote-Konfiguration	116
6.4.1	HTTP-Endpunkt	116
6.4.2	JMX	118
6.5	Remote-Logging	118

7	Beschleunigter Entwicklungsprozess mit den devtools ...	121
7.1	Automatische Neustarts	122
7.2	Automatisches Neuladen von Inhalten	125

III Das Spring-Ökosystem 127

8	Webanwendungen	129
8.1	Spring Web MVC	130
8.1.1	Die Grundlagen verstehen	130
8.1.2	Funktionen kennenlernen	132
8.1.3	Spring Web MVC automatisch konfigurieren	139
8.1.4	Eingebetteten Webcontainer nutzen	140
8.1.5	Statische Inhalte ausliefern	145
8.1.6	JSON-Serialisierung und -Deserialisierung steuern ...	146
8.1.7	Hochgeladene Dateien verarbeiten	150
8.1.8	Zusätzliche Scopes beherrschen	153
8.2	Template Engines	156
8.2.1	Thymeleaf-Templates nutzen	158
8.2.2	Groovy-Templates verwenden	162
8.2.3	Ausblick	163
8.3	Über WebSockets kommunizieren	164
8.4	JAX-RS nutzen	165
9	Security	167
9.1	Minimale Autokonfiguration	168
9.2	Die Grundlagen verstehen	168
9.2.1	Authentifizierung	169
9.2.2	Autorisierung	170
9.2.3	Spring Security und Spring Web MVC	171
9.2.4	Methodensicherheit	172
9.2.5	Sicherheit auf HTTP-Header-Ebene	173
9.3	Anwendungsbeispiele	175
9.3.1	Die Standardkonfiguration nutzen	175
9.3.2	Methodensicherheit aktivieren	177
9.3.3	Gesicherte URLs und Methoden testen	178
9.3.4	Eine eigene Benutzerverwaltung anbinden	180
9.3.5	Passwörter sicher speichern	181
9.3.6	HTTP-Sicherheit anpassen	184
9.3.7	Form-Login und Webintegration nutzen	186
9.3.8	Spring Data Repositories absichern	187
9.3.9	OAuth 2 nutzen	188
9.4	Ausblick	196

10	Persistenz	197
10.1	Spring Data	197
10.2	Relationale Datenbanken	198
10.2.1	Datasources	198
10.2.2	Datenbankinitialisierung und -migration	202
10.2.3	Transaktionen	208
10.2.4	Datenbankzugriff	214
10.3	NoSQL-Technologien	227
10.3.1	Spring Data MongoDB	227
10.3.2	Weitere Spring-Data-Module	232
10.4	Fazit	234
11	Caching	235
11.1	Aspekte eines Cache	235
11.2	Deklaratives Caching	236
11.2.1	Caching aktivieren	237
11.3	Caching mit Spring Boot	238
11.4	Beispiele und Sonderfälle	239
11.4.1	Cachen von 3rd-Party-Abhängigkeiten	240
11.4.2	Synchrones Caching	241
11.4.3	Caching von Web-Controller-Methoden	242
12	Messaging	243
12.1	Über JMS kommunizieren	246
12.1.1	JMS-Grundlagen kennen	247
12.1.2	Einen JMS-Broker konfigurieren	249
12.1.3	JmsTemplate und Listener verstehen	249
12.2	AMQP nutzen	256
12.3	Redis als Messaging-System verwenden	258
12.4	Apache Kafka anbinden	260
12.5	Ausblick: Spring Cloud Stream	261
13	E-Mail	265
13.1	Mit Spring Boot E-Mails verschicken	265
14	Reaktive Programmierung	267
14.1	Reactive Streams und Project Reactor	269
14.2	WebFlux-Modul	273
14.2.1	Klassische Annotationen verwenden	275
14.2.2	Datenquellen	277
14.2.3	Funktionales Programmiermodell	282
14.2.4	Reaktive Methoden testen	288
14.2.5	Reaktive Views benutzen	290
14.2.6	Spring Security mit WebFlux verwenden	291

14.3	Ausblick: Spring Cloud Function	292
15	Tests und Dokumentation	295
15.1	Spring-Boot-Starter-Test	296
15.2	Unit-Tests	297
15.3	Integrationstests	302
15.3.1	Grundlage: Das Frameworkmodul »spring-test«	302
15.3.2	Spring-Boot-Anwendungen testen	307
15.3.3	Automatische Mock-Objekte	313
15.3.4	Explizite Tests technischer Schichten	315
15.3.5	Erweiterte Testkonfiguration	320
15.3.6	Hilfsmittel	326
15.3.7	Fazit	327
16	Dynamische JVM-Sprachen und polyglotte Programmie- rung	329
16.1	Groovy	330
16.1.1	Das Spring-Boot-Commandline-Interface	332
16.2	Kotlin	336
IV	Produktivsetzung	339
17	Actuator	341
17.1	Bereitstellen des Spring Boot Actuator	341
17.2	Übersicht	342
17.2.1	Security	344
17.2.2	Konfiguration der HTTP-Endpunkte	347
17.2.3	Konfiguration der JMX-Endpunkte	347
17.3	Allgemeine Informationen	348
17.4	Health-Status	350
17.5	Metriken mit Micrometer aufzeichnen	352
17.5.1	Verfügbare Metriken	354
17.5.2	Eigene Metriken erfassen	355
17.5.3	Den Metrics-Endpoint benutzen	356
17.5.4	Metriken exportieren und auswerten	358
17.6	Eigene Endpunkte	359
18	Verteilung	361
18.1	Artefakte	361
18.2	Klassische Installation	367
18.2.1	Als Unix/Linux Service	367
18.2.2	Als Windows-Dienst	371

18.3	Servlet-Container	376
18.3.1	War-Deployment	376
18.3.2	Klassische Spring-Anwendungen migrieren	379
18.4	Verteilung in die Cloud	380
18.4.1	Cloud Foundry	381
18.4.2	Andere Plattformen und Anbieter	387

V Microservices mit Spring Cloud: ein Einstieg 389

19	Einführung und Übersicht	391
19.1	Was ist eine Microservice-Architektur?	391
19.2	Was gehört alles zu Spring Cloud?	392
19.2.1	Spring Cloud in Ihr Projekt einbinden	393
19.3	Kontexthierarchien	394
19.3.1	Der Bootstrap-Kontext	394
19.4	Der Wochenplaner: ein verteiltes Beispiel	396
20	Konfiguration in verteilter Umgebung	397
20.1	Den Konfigurationsserver starten	397
20.2	Clients anbinden	399
21	Service Discovery mit Netflix Eureka	401
21.1	Einen Eureka-Server betreiben	401
21.2	Services am Eureka-Server anmelden	403
21.3	Services aus Eureka benutzen	405
22	Circuit Breaker	411
22.1	Spring Cloud Hystrix nutzen	411
23	Weitere Aufgaben lösen	417
A	Lebenszyklus von Spring-Anwendungen und Beans	419
A.1	Auf Ereignisse im Lebenszyklus einer Anwendung reagieren .	419
A.2	Den Lebenszyklus von Beans verstehen	422
B	Erweiterte Konfiguration von Datasources	425
C	Upgrade von Spring Boot 1 auf 2	431
C.1	Spring 5	431
C.1.1	Neue Voraussetzungen: Spring 5 und Java 8	431
C.1.2	Neue Funktionen und Verbesserungen	432
C.2	Änderungen in Spring Boot 2	434

Glossar	439
Abkürzungen	443
Index	445