

Spring Power Workshop

Seminarunterlage

Version: 1.08



Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Karl-Schurz-Straße 19a
D-33100 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
An der alten Ziegelei 5
D-48157 Münster
Tel.: (+49) 02 51 / 9 24 35 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Welser Straße 9
D-86368 Gersthofen
Tel.: (+49) 08 21 / 507 492 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Kreuzberger Ring 13
D-65205 Wiesbaden
Tel.: (+49) 06 11 / 7 78 40 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Wikingerstraße 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: seminare@ordix.de

Inhaltsverzeichnis

1	Architektur-Überblick.....	7
1.1	Spring.....	8
1.2	Warum Spring ?	9
1.3	Überblick	10
1.4	Core Container.....	11
1.5	AOP und Instrumentation.....	12
1.6	Messaging.....	13
1.7	Data Access/Integration.....	14
1.8	Web.....	15
1.9	Test.....	16
1.10	Spring-Module im Überblick.....	17
1.11	Maven Dependency Management.....	18
1.12	Maven „Bill Of Materials“ → BOM.....	19
1.13	Getting Started.....	20
2	Getting started & Basics.....	21
2.1	Spring – Getting started	22
2.2	Spring – Bean, ApplicationContext, Metadata	23
2.3	Spring - Bean Definition (metadata).....	24
2.4	Spring – Bean Scope	25
2.5	Spring – Bean Lifecycle callbacks	26
2.6	Spring – Dependency Injection	27
2.7	Spring – Bean Konfiguration per XML – (xml based configuration).....	29
2.8	Spring - <property-placeholder>	30
2.9	Spring – Bean Konfiguration per Annotation.....	31
2.10	Spring – Bean Konfiguration per Java	33
2.11	Spring – @PropertySource	37
3	Schnellstart mit SpringBoot.....	38
3.1	Spring Boot – Worum geht es?	39
3.2	Spring Boot – Applikationen.....	40
3.3	Spring Boot – Systemanforderungen/Versionsinfos	41
3.4	Spring Boot – Einstieg Maven-POM	42
3.5	Spring Boot – HelloWorld als RESTful Web Service	43
3.6	Spring Boot - @RestController und @RequestMapping	44
3.7	Spring Boot - @EnableAutoConfiguration	45
3.8	Spring Boot – Klasse SpringApplication und main()	46
3.9	Spring Boot – Applikation starten mit spring-boot:run	47
3.10	Spring Boot – Applikation starten mit java -jar	48
3.11	Spring Boot – Starter-POMs (Auszug).....	49
3.12	Spring Boot – @SpringBootApplication	50
3.13	Spring Boot – Zugriff auf Argumente	52
3.14	Spring Boot – ApplicationRunner/CommandLineRunner	53
3.15	Spring Boot – Konfiguration	54
3.16	Spring Boot – Logging.....	59
4	Dependency Injection	61
4.1	Objektnetze in OO-Systemen	62
4.2	Dependency Injection.....	65
4.3	Das Framework.....	67
4.4	ApplicationContext	68
4.5	Spring-Konfiguration - XML.....	69
4.6	Spring-Konfiguration - Annotations	70
4.7	Spring-Konfiguration - Java-Config.....	71
4.8	Typen von Injektionen.....	72
4.9	Typen von Injektionen - Field-Injection	73
4.10	Typen von Injektionen - Konstruktor-Injection.....	74
4.11	Typen von Injektionen - Setter-Injection	75

4.12	Scopes	76
4.13	Lebenszyklus einer Spring-Bean	77
4.14	Zusammenfassung.....	79
5	Konfigurationsmöglichkeiten.....	80
5.1	Themen	81
5.2	Annotation: Definition von Beans.....	82
5.3	Annotations als Konfiguration festlegen.....	83
5.4	Annotations: Injizieren von Properties	84
5.5	Annotations: Init- und Destroy-Methoden	85
5.6	Annotations: Dependency-Injection	86
5.7	JSR-330 – DI for Java.....	87
5.8	JC: Definition von Beans.....	88
5.9	JC: Dependency-Injection	89
5.10	Spring Expression Language.....	90
5.11	Elvis Operator	91
6	Aspektororientierte Programmierung mit Spring AOP	92
6.1	Konzept	93
6.2	AOP-Begriffe	94
6.3	AOP - Aspect Oriented Programming	95
6.4	AOP mit Spring	96
6.5	Möglichkeiten zur Konfiguration von Aspekten	97
6.6	@AspectJ-Support	98
6.7	@AspectJ-Support	99
6.8	Erstellung von Pointcuts.....	100
6.9	Die AspectJ-Notation	101
6.10	Pointcut und Advice kombinieren.....	102
6.11	Typen von Advices	103
6.12	Beispiel für einen vollständigen Aspekt	104
6.13	Zusammenfassung.....	106
7	Aufbau einer Persistenzschicht mit der Java Persistence API (JPA).....	107
7.1	Spring Boot JPA Support	108
7.2	Definition einer Data Source	109
7.3	Erzeugen der DB-Struktur.....	110
7.4	Weitere JPA-Einstellungen	111
7.5	Java Persistence API (JPA).....	112
7.6	Charakteristik von Entities.....	113
7.7	JPA / Entities	114
7.8	@Table.....	115
7.9	@Column	116
7.10	Zu persistierende Felder	118
7.11	@Transient.....	119
7.12	@Basic.....	120
7.13	@Lob.....	122
7.14	@Temporal	123
7.15	Eingebettete Objekte.....	124
7.16	Primärschlüssel.....	126
7.17	JPA-Konfiguration	131
7.18	Der Entity Manager	132
7.19	Detachment.....	134
7.20	Der Lebenszyklus einer Entity	135
7.21	Operationen	137
7.22	Abfragen.....	138
7.23	Pfad/Join-Ausdrücke	142
7.24	Parameter.....	143
8	Relationen	144
8.1	Kardinalitäten	145

8.2	One-to-Many Beziehungen	146
8.3	Update von One-to-Many.....	148
8.4	Kaskadieren von Operationen	149
9	Transaktionen.....	151
9.1	Grundlagen Transaktion.....	152
9.2	Isolation-Level	153
9.3	Transaktionen	154
9.4	Transaktionsattribute.....	155
9.5	TransactionDefinition - Propagation -Attribute	156
9.6	TransactionDefinition Lesemodus, Timeout, Rollback.....	157
9.7	Deklarative Transaktionsverwaltung	158
9.8	JPA-Transaktionsmanager.....	159
9.9	Deklaration über Annotations.....	160
9.10	Programmierbares Transaktionsmanagement.....	161
9.11	PlatformTransactionManager.....	162
9.12	TransactionDefinition - Beispiel	163
9.13	Deklaratives vs. Programmierbares TM.....	164
9.14	Zusammenfassung.....	165
10	Einfacher Datenzugriff mit Spring Data JPA	166
10.1	Einführung.....	167
10.2	Aktivierung von Spring Data JPA	168
10.3	Konzept.....	169
10.4	Basisfunktionalität des JpaRepository	170
10.5	Einfache Abfragen.....	173
10.6	Eigene Abfragen	176
10.7	Änderungsabfragen.....	177
10.8	Custom implementations.....	178
11	Testen mit Spring	179
11.1	Agenda.....	180
11.2	Unit- vs. Integration-Tests	181
11.3	Unit Tests	182
11.4	Spring Support für Unit-Testing	183
11.5	MockEnvironment - Beispiel.....	184
11.6	Integration Tests	185
11.7	Spring Support für Integration-Testing.....	186
11.8	@RunWith (org.junit.runner.RunWith)	187
11.9	Annotations	188
11.10	@ContextConfiguration.....	189
11.11	@WebAppConfiguration	190
11.12	@ContextHierarchy	191
11.13	@ActiveProfiles.....	192
11.14	@TestPropertySource.....	193
11.15	@DirtiesContext.....	194
11.16	@Commit / @Rollback.....	195
11.17	@IfProfileValue (nur JUnit 4 mit SpringRunner)	196
11.18	@Timed / @Repeat (nur JUnit 4 mit SpringRunner)	197
11.19	Context-Caching	198
11.20	Spring MVC Test Framework.....	199
11.21	Spring MVC Test-Support.....	200
11.22	Out-of-Container vs. End-to-End Integrationstests.....	201
11.23	Spring MCV Test-Support	202
11.24	Spring Boot Test Support.....	203
11.25	Spring Boot Test Support.....	204
11.26	@SpringBootTest.....	205
11.27	Test-Konfiguration	206
11.28	Random-Port.....	207
11.29	TestRestTemplate.....	208

12	Absicherung der eigenen Anwendung mit Spring Security	209
12.1	Was ist Spring Security?	210
12.2	Begriffsabgrenzung	211
12.3	Grundlegende Objekte	212
12.4	Beispielhafter Ablauf	213
12.5	Konfiguration: Maven Dependencies	214
12.6	Konfiguration: Filter	215
12.7	Filter Chain	216
12.8	Konfiguration: JSF Support	217
12.9	Java Konfiguration - WebSecurityConfig	218
12.10	Java Konfiguration – Authentifizierung	219
12.11	Java Konfiguration – Definition der Autorisierungsregeln	220
12.12	Java Konfiguration – Beispiel Regeldefinition	221
12.13	Java Konfiguration – Authorize Requests	222
12.14	Login Konfiguration	223
12.15	Logout Konfiguration	224
12.16	UserDetailsService	225
12.17	UserDetailsService - Beispiel	226
12.18	Eigenen UserDetailsService verwenden	227
12.19	Annotation Support	228
12.20	Annotation PreAuthorize / Secured	229
12.21	Annotation PreAuthorize - Beispiele	230
12.22	Annotation Secured - Beispiele	231
12.23	Annotations für Tests	232
12.24	Weiterführende Projekte	233
13	Task Execution und Scheduling	234
13.1	Scheduling	235
13.2	Scheduling Beispiel	236
13.3	Asynchrone Ausführung	237
13.4	Future Interface	238
13.5	Asynchrone Ausführung Beispiel	239
14	RESTful Web Services (mit Spring MVC)	240
14.1	Allgemeines	241
14.2	Warum REST?	242
14.3	Interoperabilität	243
14.4	Entkoppelung	244
14.5	Skalierbarkeit und Performance	245
14.6	Wiederverwendung	246
14.7	Vergleich WebServices SOAP vs. REST	247
14.8	HATEOAS - REST Anwendungsarchitektur	248
14.9	HTTP-Methoden	249
14.10	HTTP-Status Codes	251
14.11	REST - Server	252
14.12	Annotations	253
14.13	HttpMessageConverter	256
14.14	REST - Client	257
14.15	RestTemplate	258
14.16	Jackson	259
14.17	Testen auf dem Server	260
14.18	MockMvc	261
14.19	MockMvc – Beispiel	262
14.20	Testen auf dem Client	264
14.21	MockRestServiceServer	265