
Inhaltsverzeichnis

1	Ein Einstieg mit Profil	1
1.1	Java EE 7 – der Standard für Enterprise Java	1
1.1.1	Struktur einer Enterprise-Java-Anwendung	1
1.1.2	Die Java Enterprise Edition (Java EE)	2
1.1.3	Anatomie einer Java-EE-Anwendung	3
1.2	Die Komponentenarchitektur von Java EE 7	4
1.2.1	Die Komponenten der Webschicht	5
1.2.2	CDI – ein Komponententyp mit viel Potenzial	8
1.2.3	Enterprise JavaBeans	9
1.2.4	Java Persistence API (JPA)	10
1.2.5	Die Java-EE-Profile	11
1.2.6	Und ab geht's auf den Applikationsserver!	13
1.3	Die Zielarchitektur	15
1.4	Wie geht es jetzt weiter?	16
1.5	Weiterführende Literatur	18
2	Aufsetzen der Entwicklungsumgebung	19
2.1	Werkzeuge und Server für Java EE	19
2.2	Installation der nötigen Softwarepakete	20
2.2.1	Installation von Java SE 8	21
2.2.2	Installation von Maven 3	22
2.2.3	Installation von WildFly 8.1.0	23
2.3	Vom Projekt zur ausgeführten Webanwendung	24
2.3.1	Anlegen eines leeren Java-EE-7-Webprojektes mit Maven	24
2.3.2	Der erste Build und das erste Deployment	28
2.3.3	Test des Deployments	30

2.4	Eclipse Luna – Mehr als ein Texteditor	31
2.4.1	Installation von Eclipse	31
2.4.2	Eclipse-Plug-in m2e	32
2.4.3	Import von Maven-Projekten	33
2.4.4	Das Deployment	35
2.5	Weitere Werkzeuge des Workshops	37
2.6	Weiterführende Literatur	37
3	Fachlichkeit der Beispielanwendungen »My-Aktion« und	
	»My-Aktion-Monitor«	39
3.1	Einleitung	39
3.2	Übersicht der Anwendungsfälle	39
3.3	Fachklassen	41
3.4	Anwendungsfälle	43
3.4.1	Anmeldung notwendig	44
3.4.2	Aktionen anzeigen und bearbeiten	44
3.4.3	Spendenformular bearbeiten	45
3.4.4	Spendenliste anzeigen	46
3.4.5	Aktion bearbeiten	47
3.4.6	Aktion löschen	47
3.4.7	Neue Aktion anlegen	48
3.4.8	Aktionsdaten editieren	48
3.4.9	Geld spenden	50
3.4.10	Reduzierte Spendenliste anzeigen	52
3.5	Weiterführende Literatur	53
4	Iteration Nr. 1 – JavaServer Faces	55
4.1	Einleitung	55
4.2	Fachklassen als JavaBeans	56
4.2.1	Konto	56
4.2.2	Spende	57
4.2.3	Aktion	58
4.3	Internationalisierung	60
4.4	Das Vorlagensystem	63
4.4.1	Templating mit Facelets	63
4.4.2	Ein Template für My-Aktion	64
4.4.3	Resource Library Contracts	67

4.5	Der Anwendungsfall »Aktionen anzeigen und bearbeiten«	70
4.5.1	Die Backing Beans	70
4.5.2	JSF-Technologien für den ersten Anwendungsfall	75
4.5.3	Unsere erste View!	81
4.5.4	Der Start – ein erster Meilenstein	83
4.6	Weitere JSF- und Java-EE-Technologien	87
4.6.1	@Inject-Annotation	87
4.6.2	Texteingabe mit Validierung, Konvertierung und Fehlermeldungen	88
4.6.3	Das Rasterlayout von UI-Komponenten	95
4.6.4	Auswahlkomponenten	96
4.6.5	Die GET-Parameterverarbeitung in einer View	97
4.6.6	Ajax – Etwas mehr Dynamik gefällig?	98
4.6.7	HTML5 Friendly Markup	100
4.7	Die Implementierung der restlichen Anwendungsfälle	101
4.7.1	Aktion bearbeiten und Aktion neu anlegen	101
4.7.2	Spendenliste anzeigen	110
4.7.3	Spendenformular bearbeiten	114
4.7.4	Geld spenden	120
4.8	Neue Funktionalitäten in Java EE 7	125
4.8.1	Faces Flows	125
4.8.2	Stateless Views	126
4.9	Exkurs: PrimeFaces	127
4.9.1	Installation über Maven	128
4.9.2	Neue Farbauswahl für die Bearbeitung des Spendenformulars	128
4.9.3	Tabs für das Anlegen und Editieren von Aktionsdaten	133
4.9.4	Der Bestätigungsdialo für den Anwendungsfall »Aktion löschen«	137
4.10	Aufgaben	142
4.10.1	Verwendung von PrimeFaces-Komponenten für alle Views (Pflichtaufgabe)	142
4.10.2	Lokalisierung der Schaltfläche »Update URL«	143
4.10.3	Durchgängige Verwendung von Inline-Labels	144
4.10.4	Auslagerung der Eigenschaften des Spendenformulars	144
4.11	Weiterführende Literatur	145

5	Iteration Nr. 2 – Funktionale Tests	147
5.1	Einleitung	147
5.2	Testframeworks Arquillian und Graphene konfigurieren	148
5.3	Wiederwendbare Klassen erstellen	150
5.3.1	Testdaten für Aktionen und Spenden bereitstellen	151
5.3.2	Anwendung My-Aktion mit Arquillian deployen	152
5.3.3	Webseiten mit Graphene steuern	153
5.4	Anwendungsfälle testen	156
5.4.1	Den Anwendungsfall »Neue Aktion anlegen« testen	156
5.4.2	Den Anwendungsfall »Geld spenden« testen	161
5.5	Aufgaben	165
5.5.1	SetupDatabase erweitern	166
5.5.2	Testfall für bisher gespendeten Betrag hinzufügen	166
5.5.3	Weitere Testfälle erstellen	166
5.6	Literaturverzeichnis	166
6	Iteration Nr. 3 – CDI	167
6.1	Warum CDI?	167
6.2	Der Laufzeit-Container	168
6.2.1	Der Sichtbarkeitsbereich (Scope) einer Bean	169
6.2.1.1	RequestScope	169
6.2.1.2	SessionScope	170
6.2.1.3	ViewScope	170
6.2.1.4	DependentScope	171
6.2.1.5	Setzen der Scopes für unsere Anwendung	171
6.2.2	Beans referenzieren über Dependency Injection	172
6.2.3	Der Lebenszyklus	174
6.2.3.1	Lebenszyklusmethode PostConstruct	174
6.2.3.2	Lebenszyklusmethode PreDestroy	175
6.2.4	Beliebige Klassen als Beans mit Producer-Methoden	175
6.3	Anwendungsweite Nachrichten	180
6.3.1	Events senden und empfangen	180
6.3.2	Nachrichten gleichen Typs mit Qualifiern unterscheiden	181
6.4	Services	184
6.4.1	Die Mock-Methode in eine Serviceklasse auslagern	184
6.4.2	Die verwendete Serviceimplementierung mit Qualifiern auswählen	187

6.5	Was noch nicht behandelt wurde	190
6.5.1	ApplicationScope und ConversationScope	190
6.5.2	Der FlowScope	191
6.5.3	Interzeptoren und Dekoratoren	191
6.5.4	Stereotypes	192
6.5.5	Der BeanManager	192
6.5.6	Erweiterungen	193
6.5.7	Disposer-Methoden	193
6.6	Aufgaben	194
6.6.1	Fachliches und technisches Log unterscheiden	194
6.6.2	Nachricht zur Aktualisierung einer Aktion hinzufügen	194
6.7	Weiterführende Literatur	195
7	Iteration Nr. 4 – Java Persistence API	197
7.1	Einleitung	197
7.2	Entitäten annotieren	198
7.2.1	Als Entität markieren und Primärschlüssel festlegen	198
7.2.2	Relationen der Entitäten festlegen	200
7.2.3	Abhängige Entitäten einbinden	202
7.3	Services um Datenzugriffe erweitern	203
7.3.1	EntityManager einbinden	203
7.3.2	Datenbankabfragen hinzufügen	204
7.3.3	CRUD-Operationen ausführen	206
7.3.4	CampaignListProducer um Servicedelegation erweitern	208
7.4	Daten transaktional speichern	210
7.4.1	Datenquelle konfigurieren	210
7.4.2	Services um Transaktionen erweitern	213
7.5	Spenden persistieren und anzeigen	214
7.5.1	Service zur Bearbeitung von Spenden erstellen	214
7.5.2	Anwendungsfall <i>Geld spenden</i> finalisieren	216
7.5.3	Den bisher gespendeten Betrag berechnen	217
7.5.4	Spenden in der Spendenliste anzeigen	219
7.6	Eingaben über Bean Validation überprüfen	221
7.6.1	Entitäten mit Bedingungen annotieren	222
7.6.2	Validatoren aus Views entfernen	226
7.6.3	Zugeordnete Spenden beim Löschen einer Aktion automatisch entfernen	227
7.7	Eigene In-Memory-Datasource für Tests benutzen	229

7.8	Was noch nicht behandelt wurde	231
7.8.1	CriteriaQuery als Alternative zur NamedQuery	231
7.8.2	LazyInitialisationException über FetchType vermeiden	233
7.8.3	Object Locking	234
7.8.3.1	Optimistische Sperre	234
7.8.3.2	Pessimistischen Sperre	235
7.8.4	Lebenszyklusmethoden bei Zustandsänderungen	236
7.9	Neue Funktionalitäten in Java EE 7	238
7.9.1	Bean Validation auf Methodenebene	238
7.9.2	Erweiterte CDI-Unterstützung	238
7.9.3	Aufruf gespeicherter Prozeduren und Funktionen	239
7.9.4	Schreibende Massenoperationen mit der Criteria API	240
7.10	Zusammenfassung	240
7.11	Aufgaben	241
7.11.1	JPQL-Abfragen mit Criteria API ersetzen	241
7.11.2	Zeitstempel der Aktualisierung hinzufügen	241
7.11.3	Minimalen Spendenbetrag der Aktion überprüfen	242
7.11.4	Organisator über Spendenziel informieren	242
7.12	Weiterführende Literatur	242
8	Iteration Nr. 4 – Enterprise JavaBeans	243
8.1	Einleitung	243
8.2	Sicherheit	244
8.2.1	Organisatoren speichern	244
8.2.2	Security-Domain für WildFly anlegen	250
8.2.3	Benutzeroberfläche absichern	252
8.2.4	Eigene Aktionen für jeden Organisator	254
8.2.5	Services über Annotationen absichern	256
8.2.6	Mehr Sicherheit durch SSL	259
8.2.7	Eigenen Anmeldebildschirm festlegen	261
8.2.8	Automatisches Login für die Tests	264
8.3	Transaktionssteuerung	266
8.3.1	Transaktionssteuerung durch den Container	266
8.3.2	Transaktionssteuerung über die Bean	269
8.3.3	Transaktionssteuerung über Interzeptoren	270
8.4	Zeitgesteuerte Abläufe realisieren	274
8.4.1	Spenden überweisen	274
8.4.2	Hintergrundjob zeitlich gesteuert aufrufen	275

8.5	Vergleich EJBs und CDI	277
8.5.1	Unterschiedlicher Lebenszyklus	277
8.5.2	Vorteile der jeweiligen Technologie	279
8.6	Was nicht behandelt wurde	280
8.6.1	Stateful Session Beans	280
8.6.2	Entfernter Methodenaufruf	282
8.6.3	Nebenläufigkeit	283
8.6.4	Asynchroner Methodenaufruf	285
8.6.5	Message-Driven Beans	286
8.6.6	Entity Beans	287
8.7	Aufgaben	287
8.7.1	Registrierungsformular für Organisatoren erstellen	287
8.7.2	Services auf CDI umstellen	288
8.7.3	Ausführungszeit einer Methode messen	288
8.8	Weiterführende Literatur	289
9	Iteration Nr. 5 – Kommunikation und HTML5	291
9.1	Einleitung	291
9.2	Neue Technologien	291
9.2.1	Webservices	291
9.2.2	WebSockets	293
9.3	Kommunikationsarchitektur	293
9.4	REST-Webservice-Schnittstelle für My-Aktion	295
9.4.1	REST-Webservice für das Management von Aktionen	298
9.4.1.1	Liste aller Aktionen eines Organisations abfragen	299
9.4.1.2	Aktion löschen	301
9.4.1.3	Neue Aktion hinzufügen	303
9.4.1.4	Aktion aktualisieren	305
9.4.2	REST-Webservice zur Abfrage und Ausführung von Spenden	306
9.4.2.1	Spendenliste einer Aktion abfragen	307
9.4.2.2	Geld spenden	308
9.5	Die Webapplikation My-Aktion-Monitor	314
9.5.1	Anwendungsserver Glassfish installieren	315
9.5.2	Bestandteile der Anwendung My-Aktion-Monitor	320
9.5.2.1	Spendenliste einer Aktion anzeigen	321
9.5.2.2	SOAP-Webservice für My-Aktion-Monitor erstellen	341
9.5.2.3	SOAP-Webservice testen	343
9.5.2.4	Spendenliste über SOAP aktualisieren	346

9.6	Nicht behandelt	348
9.6.1	JMS	348
9.6.2	Java Mail	349
9.6.3	JMX	350
9.7	Aufgaben	352
9.7.1	Erweiterung der REST-Services um XML	352
9.7.2	Eigenen REST-Client erstellen	352
9.8	Literaturverzeichnis	353
10	Java EE 7 und die Cloud	355
10.1	Einleitung	355
10.2	Was ist eine Cloud?	355
10.2.1	Standardisierungsbedarf für Java-EE-Cloud-Szenarien	357
10.3	My-Aktion in OpenShift installieren	358
10.3.1	Zugang für OpenShift anlegen	358
10.3.2	Notwendige Werkzeuge installieren	358
10.3.3	Neue Anwendung in OpenShift anlegen	361
10.3.4	My-Aktion in OpenShift deployen	362
10.3.5	Anpassungen für OpenShift vornehmen	367
10.4	Literaturverzeichnis	370
A	Anhang	371
A.1	Vom Servlet zu JSF mit CDI und EJB	371
A.1.1	Ein paar Hinweise vorweg	371
A.1.2	Java Servlets	372
A.1.3	JavaServer Pages und JavaBeans	373
A.1.4	JavaServer Faces und JSF Managed Beans	375
A.1.5	JavaServer Faces und CDI	376
A.1.6	JSF, CDI und Enterprise JavaBeans	377
A.2	My-Aktion und My-Aktion-Monitor auf GitHub	378
A.2.1	Git installieren	378
A.2.2	Git-Repository des Workshops verwenden	380
A.2.3	Literatur zu Git	381
	Index	383