

# Inhaltsverzeichnis

<b>Geleitwort</b>	<b>1</b>
<b>Vorwort</b>	<b>3</b>
<b>1 Einleitung</b>	<b>7</b>
1.1 Was ist Vaadin? .....	7
1.2 Historie .....	8
1.3 Wofür ist Vaadin gut? .....	9
1.4 Unsere Erfahrungen mit Vaadin .....	10
1.5 Weitergehende Informationen .....	11
<b>2 Erste Schritte mit Vaadin</b>	<b>13</b>
2.1 Einrichten der Entwicklungsumgebung Eclipse .....	13
2.2 Erste Schritte .....	17
2.3 Verwendung von Maven für Vaadin-Projekte in Eclipse ..	23
2.4 Andere Entwicklungsumgebungen .....	24
<b>3 Die Vaadin-Architektur</b>	<b>29</b>
3.1 Einleitung .....	29
3.2 Überblick .....	30
3.2.1 Serverseitiges Programmiermodell .....	30
3.2.2 Clientseitiges Programmiermodell .....	33
3.2.3 Push .....	36
3.2.4 Themes .....	37
3.2.5 Add-ons .....	38

3.3	Vaadin unter der Haube	38
3.3.1	Der Einstiegspunkt	38
3.3.2	Die Benutzeroberfläche	39
3.3.3	Der erste Aufruf der UI	40
3.4	Konsequenzen der Vaadin-Architektur	45
3.4.1	Serverseitiger Speicherbedarf	45
3.4.2	Clientseitiger Speicherbedarf und Render-Performance	46
3.4.3	Netzwerkkommunikation	46
3.4.4	Latenz	47
<b>4</b>	<b>UI-Komponenten</b>	<b>49</b>
4.1	Grundlegende UI-Komponenten	49
4.1.1	UI als Rahmen für die Applikation	49
4.1.2	Textausgabe mit Label und Notification	50
4.1.3	Button	52
4.1.4	Texteingaben mit TextField, PasswordField und TextArea	53
4.1.5	Formatierte Texteingabe über RichTextArea	56
4.1.6	Datumseingabe mit InlineDatePicker und PopupDatePicker	56
4.1.7	Eingabe von Werten über Slider	57
4.1.8	Auswahlkästchen mit CheckBox	58
4.1.9	Auswahlboxen mit ListSelect, ComboBox, NativeSelect, OptionGroup und TwinColSel	59
4.1.10	Hochladen von Dateien mit Upload	61
4.1.11	Hierarchieanzeige mit der Tree-Komponente	63
4.1.12	Tabellen mit der Table-Komponente	64
4.1.13	Menüzeilen mitMenuBar	65
4.1.14	Fortschrittsanzeige mit der ProgressBar-Komponente	66
4.1.15	Videos und Animationen mit Link, AbstractMedia und AbstractEmbedded	66
4.2	Layout-Klassen	67
4.2.1	Elemente untereinander: VerticalLayout	68
4.2.2	Elemente nebeneinander: HorizontalLayout	71
4.2.3	Zwischenbetrachtung: Schachteln von Layouts	72
4.2.4	GridLayout – Elemente im Gitter	72
4.2.5	FormLayout	75
4.2.6	AbsoluteLayout	76
4.2.7	CssLayout	77
4.2.8	CustomLayout	78

4.3	Sonderlayouts .....	79
4.3.1	Scrolling mittels Panel .....	79
4.3.2	Splitter mit Horizontal- und VerticalSplitPanel ..	80
4.3.3	Karteireiter im TabSheet .....	82
4.3.4	Accordions .....	83
4.3.5	Popup-Fenster mit Window .....	84
4.4	Eigene Komponenten und Eingabefelder .....	85
4.4.1	CustomComponents mit dem visuellen Editor ..	86
<b>5</b>	<b>Data Binding</b>	<b>91</b>
5.1	Einführung .....	91
5.2	Details zum »Data Binding«-Prozess .....	94
5.3	Das Vaadin-Container-Datenmodell im Allgemeinen ..	101
5.4	Validierung in der GUI .....	102
5.4.1	Nutzung von JSR-303-Validatoren .....	103
5.4.2	Nutzung Vaadin-spezifischer Validatoren .....	108
5.5	Erkennen von Änderungen .....	110
<b>6</b>	<b>Real-Time-Webapplikationen mit Vaadin Server Push</b>	<b>113</b>
6.1	Ein kleines Beispiel .....	113
6.2	Ein Mini-Chat .....	115
<b>7</b>	<b>Layout und Styling</b>	<b>119</b>
7.1	Auswahl eines Theme .....	119
7.2	Erstellung eines eigenen Theme .....	120
7.2.1	Struktur eines Theme .....	120
7.2.2	Erstellung eines CSS-basierten Theme .....	121
7.2.3	Erstellung eines SCSS-basierten Theme .....	122
7.2.4	Theme-Vorlagen .....	125
7.2.5	CSS-Selektor-Strategien .....	125
7.2.6	SCSS-Sprachkonstrukte nutzen .....	132
7.2.7	CSS Injection .....	137
<b>8</b>	<b>Navigation in Ajax-Anwendungen</b>	<b>139</b>
8.1	Einleitung .....	139
8.2	URI-Fragmente .....	140
8.3	Format der URI-Fragmente .....	141
8.4	URI-Fragmente und Applikationszustand .....	141
8.5	Manipulation und Überwachung des URI-Fragments ..	142

<b>8.6</b>	<b>Die Navigator-API</b>	<b>143</b>
8.6.1	Der Navigator	144
8.6.2	Die ViewProvider	146
8.6.3	Der NavigationStateManager	147
8.6.4	Ablauf eines View-Wechsels	149
<b>9</b>	<b>Anwendungsarchitektur mit Vaadin</b>	<b>153</b>
9.1	Einleitung	153
9.2	Überblick	154
9.3	Was verstehen wir unter Architektur?	154
9.4	Architekturmuster	155
9.4.1	Ein kurzer Überblick	155
9.4.2	Die Schichtenarchitektur	156
9.4.3	Die MVx-Architekturmuster	157
9.5	Auf in die Praxis	163
9.5.1	Die Beispielanwendung	163
9.5.2	Das Beispielprojekt	164
9.5.3	Das Domänenmodell der Beispielanwendung	165
9.6	Der Login in der MVP-Passive-View-Variante	166
9.6.1	Schnittstellendefinition und Implementierung	167
9.7	Entkopplung mit einem Event-Bus	185
9.7.1	Der Event-Bus von Guava	187
9.7.2	Umstellung auf den Event-Bus	188
9.7.3	Vor- und Nachteile von Event Handling	192
9.8	Dependency Injection	192
9.8.1	Die Umstellung auf CDI	194
9.8.2	Umstellung der Anwendung	195
9.9	Der Profileditor in der MVP-Variante	201
9.9.1	Schnittstellendefinition und Implementierung	202
9.10	MVVM am Beispiel der Anmeldekomponente	212
9.10.1	Umbau des Presenter in ein Presentation Model	212
9.10.2	Die View und die Integration in der Anwendung	216
9.11	Zusammenfassung	219
<b>10</b>	<b>Add-ons</b>	<b>221</b>
10.1	Einleitung	221
10.2	Vaadin-Directory	221
10.3	Add-on verwenden	225

<b>11</b>	<b>Vaadin und Maven</b>	<b>231</b>
11.1	Einleitung . . . . .	231
11.2	Überblick Archetypes . . . . .	232
11.3	Maven-Projekt erstellen . . . . .	232
11.3.1	Kommandozeile . . . . .	232
11.3.2	Eclipse . . . . .	233
11.4	Projektstruktur im Detail . . . . .	235
11.5	Das Vaadin-Maven-Plugin . . . . .	237
<b>12</b>	<b>Automatisiertes Testen von Vaadin-Anwendungen</b>	<b>239</b>
12.1	Einleitung . . . . .	239
12.2	Unit-Tests . . . . .	239
12.2.1	Interaktion mit Vaadin-Komponenten . . . . .	239
12.2.2	Interaktion mit ThreadLocals . . . . .	242
12.3	GUI-Tests mit Selenium . . . . .	245
12.3.1	Ein erster Test . . . . .	245
12.3.2	Locator . . . . .	247
12.3.3	Warten will gelernt sein . . . . .	249
12.3.4	Das Page-Object-Pattern . . . . .	253
12.3.5	Unabhängigkeit von Einzeltests . . . . .	258
<b>13</b>	<b>Zusammenfassung</b>	<b>261</b>
<b>14</b>	<b>Literatur</b>	<b>263</b>
	<b>Index</b>	<b>267</b>