

# Entwicklung von Web-Applikationen mit Lift und Scala

Einführung anhand einer durchgehenden Beispielapplikation



Thomas Fiedler  
Christoph Knabe

20. Januar 2011

**Thomas Fiedler** studierte bis 2010 Medieninformatik an der Beuth-Hochschule für Technik Berlin. Er arbeitet zur Zeit als freiberuflicher Softwareentwickler im Bereich der Web- und Oberflächenprogrammierung.



**Christoph Knabe** ist seit 1990 Professor für Softwaretechnik und Programmierung an der Beuth-Hochschule für Technik Berlin und übt für den Fachbereich VI die Funktion des Web-Beauftragten aus. Seine Interessensgebiete sind Scala, Lift, Web-Programmierung, AspectJ, Ausnahmebehandlung und Software-Qualität. Er unterrichtet im Studiengang Medieninformatik.



<http://public.beuth-hochschule.de/~knabe/>

**Titelbild:** Lift am Green Bank Telescope, Pocahontas County, West Virginia.

**Fotograf:** Alan Aversa.

[http://commons.wikimedia.org/wiki/File:GBT\\_Elevator.png](http://commons.wikimedia.org/wiki/File:GBT_Elevator.png)

Berichte aus der Softwaretechnik

**Thomas Fiedler  
Christoph Knabe**

**Entwicklung von Web-Applikationen  
mit Lift und Scala**

Einführung anhand einer  
durchgehenden Beispielapplikation

Shaker Verlag  
Aachen 2011

**Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Copyright Shaker Verlag 2011

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8322-9824-1

ISSN 1433-9986

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

---

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Aufbau des Buches . . . . .	1
1.2. Grundsätzliches . . . . .	3
1.3. Verwendung der Beispielanwendung . . . . .	3
1.4. Typographische Hervorhebungen . . . . .	4
<b>2. Scala</b>	<b>5</b>
2.1. Eigenschaften . . . . .	5
2.1.1. Objektorientierung . . . . .	5
2.1.2. Funktionale Programmierung . . . . .	6
2.1.3. Erweiterbarkeit . . . . .	6
2.1.4. Statisches Typsystem . . . . .	6
2.1.5. XML-Unterstützung . . . . .	7
2.2. Entwicklungsumgebungen . . . . .	7
2.3. Leistung . . . . .	8
<b>3. Lift</b>	<b>9</b>
3.1. Vorbilder . . . . .	9
3.2. Nutzung von Scala-Features . . . . .	10
<b>4. Maven und Lift</b>	<b>11</b>
4.1. Kurzbeschreibung . . . . .	11
4.2. Konfiguration . . . . .	11
4.3. Repositories . . . . .	12
4.4. Plugins . . . . .	12
4.5. Die POM-Datei . . . . .	13
4.6. Einsatz in Lift . . . . .	14
4.6.1. Erstellung eines neuen Lift-Projekts . . . . .	14
4.6.2. Kompilierung . . . . .	15
4.6.3. Deployment und Inbetriebnahme . . . . .	15
4.6.4. Bearbeitung des Projekts für die Nutzung in einer IDE . . . . .	17
4.6.5. Erstellung einer Quelltext-Dokumentation . . . . .	18

<b>5. Die Beispielanwendung „Lehrkraftnews“</b>	<b>19</b>
5.1. Kurzbeschreibung . . . . .	19
5.1.1. Benutzerverwaltung . . . . .	19
5.1.2. Nachrichten . . . . .	20
5.1.3. Abonnements . . . . .	20
5.1.4. Zugrunde liegendes Datenmodell . . . . .	20
5.1.5. Besonderheiten . . . . .	21
5.2. Installation und Inbetriebnahme . . . . .	22
5.2.1. Installation der benötigten Software . . . . .	23
5.2.2. Inbetriebnahme der Anwendung . . . . .	25
5.2.3. Bedienung . . . . .	29
<b>6. Test-Infrastruktur und Testorientierte Dokumentation</b>	<b>31</b>
6.1. Test-Infrastruktur . . . . .	31
6.1.1. Framework-Auswahl, JUnit, SUnit, specs . . . . .	31
6.1.2. ScalaTest . . . . .	32
6.1.3. Nutzung von ScalaTest/JUnit in Maven und der IDE . . . . .	33
6.2. Testgetriebene Entwicklung . . . . .	33
6.3. Testorientierte Dokumentation . . . . .	35
6.3.1. Extraktion von Codeschnipseln . . . . .	35
6.3.2. Testfälle als Programmbeispiele . . . . .	35
<b>7. Das Mapper-Framework</b>	<b>37</b>
7.1. Grundsätzliches . . . . .	37
7.1.1. Mapper/Record . . . . .	37
7.2. Integration des Mapper-Moduls in das Projekt . . . . .	37
7.3. Konfiguration des Datenbankzugriffs . . . . .	38
7.4. Erstellung von Modell-Klassen . . . . .	41
7.4.1. Mapper- und MetaMapper-Trait . . . . .	45
7.4.2. Primärschlüssel mit LongKeyedMapper und IdPK . . . . .	45
7.4.3. Festlegung des Tabellennamens . . . . .	45
7.4.4. Definition von Datenfeldern . . . . .	45
7.4.5. Hinzufügen von Hilfsfunktionen . . . . .	46
7.4.6. Validierung von Datenfeldern . . . . .	46
7.5. Objekt-Operationen . . . . .	48
7.5.1. Erstellen und Speichern . . . . .	48
7.5.2. Zugriff auf Datenfelder . . . . .	49
7.5.3. Zugriff auf Objekt-Verbindungen . . . . .	49
7.5.4. Aktualisieren . . . . .	50
7.6. Datenbankabfragen . . . . .	50
7.6.1. Methoden . . . . .	50
7.6.2. Vergleichs-Query-Parameter . . . . .	51

---

7.6.3.	Steuer-Query-Parameter . . . . .	54
7.6.4.	Native SQL-Statements . . . . .	54
7.6.5.	Zusammenfassung . . . . .	56
7.7.	Erzeugung des Datenbankschemas mit Schemifier . . . . .	57
7.8.	CRUD-Funktionalität durch den CRUDify-Trait . . . . .	57
7.9.	ProtoUser und MegaProtoUser . . . . .	61
7.9.1.	ProtoUser . . . . .	61
7.9.2.	MegaProtoUser und MetaMegaProtoUser . . . . .	62
<b>8.</b>	<b>Views und Templates</b> . . . . .	<b>71</b>
8.1.	View First-Pattern . . . . .	71
8.2.	Arbeit mit Templates . . . . .	72
8.2.1.	Pfadstruktur und Benennung . . . . .	72
8.2.2.	surround/bind-Tag . . . . .	73
8.2.3.	embed-Tag . . . . .	75
8.2.4.	ignore-Tag . . . . .	75
8.2.5.	"templates-hidden"-Ordner . . . . .	75
8.2.6.	Aufruf von Snippet-Methoden . . . . .	76
8.2.7.	Einbettung von CometActors . . . . .	76
8.2.8.	Internationalisierung mit Templates . . . . .	77
8.2.9.	Head Merging . . . . .	77
8.3.	Arbeit mit Views . . . . .	77
8.3.1.	LiftView . . . . .	77
8.3.2.	InsecureLiftView . . . . .	81
8.4.	Ausgabe von Hinweisen, Warnungen und Fehlermeldungen . . . . .	82
<b>9.</b>	<b>Snippets</b> . . . . .	<b>85</b>
9.1.	bind . . . . .	87
9.1.1.	Extraktion eines Parameters aus der URI . . . . .	87
9.1.2.	Ersetzen von Platzhaltern durch bind . . . . .	88
9.1.3.	Iteration und bind . . . . .	89
9.1.4.	bind und chooseTemplate . . . . .	90
9.2.	Formulare . . . . .	92
9.3.	RequestVar . . . . .	93
9.4.	SessionVar . . . . .	97
9.5.	Stateful Snippets . . . . .	97
<b>10.</b>	<b>AJAX und Comet</b> . . . . .	<b>103</b>
10.1.	AJAX . . . . .	103
10.2.	Comet . . . . .	106
10.2.1.	Autonome CometActors . . . . .	107
10.2.2.	Koordination von CometActors . . . . .	116

<b>11. Die Boot-Klasse</b>	<b>121</b>
11.1. LiftRules	121
11.2. Mailer-Konfiguration	123
11.3. SiteMap	125
11.3.1. Erweiterte Zugriffssteuerung	126
11.3.2. Die SiteMap-DSL ab Lift 2	127
11.3.3. Generierung von Menüs	131
11.3.4. Festlegung des zu verwendenden Templates	135
11.3.5. Übersicht der Menüstruktur der Beispielanwendung	137
11.4. URI-Pfad-Rewriting	138
11.5. Konfiguration des Datenbankzugriffs	140
11.6. Schemifier	140
11.7. Initialisierung von Widget-Klassen	140
<b>12. Ausnahmebehandlung und Zentrales Ausnahmemelden</b>	<b>141</b>
12.1. Ausnahmebehandlung in der Anwendungssoftware	141
12.2. Zentrales Ausnahmemelden bei Full-Page-Requests	142
12.3. Zentrales Ausnahmemelden bei Ajax-Requests	146
12.4. Zentrales Ausnahmemelden bei Comet-Requests	151
12.5. Ausnahmemelden bei Hintergrund-Actors	152
12.6. Zusammenfassung	154
<b>13. Internationalisierung</b>	<b>157</b>
13.1. Internationalisierte Templates	157
13.2. properties-Dateien	157
13.2.1. lift-core.properties	157
13.2.2. lift.properties	158
<b>14. Fazit</b>	<b>161</b>
<b>A. Anhang</b>	<b>163</b>
A.1. Bearbeitung der Beispielapplikation mit einer IDE	163
A.2. Logging in der Applikation mit SLF4J über Log4J	163
A.3. Box, Full und Empty	165
A.4. Paginierung	166
A.5. Das TableSorter-Widget	169
A.6. Scala-Actors	172
A.7. Das S-Objekt	175
A.8. CometMailer	175
<b>Literatur-, Abbildungs- und Programmauszugs-Verzeichnisse</b>	<b>181</b>