

entwickler.press

# Grails

Sven Haiges

schnell + kompakt



Grails



Sven Haiges

# Grails

Webanwendungen  
mit Groovy und Grails

schnell + kompakt

entwickler.press

Sven Haiges  
Grails  
schnell + kompakt  
ISBN 978-3-939084-83-9

© 2008 entwickler.press,  
ein Imprint der Software & Support Verlag GmbH

1. Auflage, 2008

<http://www.entwickler-press.de>  
<http://www.software-support.biz>

Ihr Kontakt zum Verlag und Lektorat: [lektorat@entwickler-press.de](mailto:lektorat@entwickler-press.de)

Bibliografische Information Der Deutschen Bibliothek  
Die Deutsche Bibliothek verzeichnet diese Publikation in der  
Deutschen Nationalbibliografie; detaillierte bibliografische Daten  
sind im Internet über <http://dnb.ddb.de> abrufbar.

Korrektur: Petra Kienle  
Satz: text & form GbR, Carsten Kienle  
Umschlaggestaltung: Caroline Butz  
Belichtung, Druck und Bindung: M.P. Media-Print Informations-  
technologie GmbH, Paderborn.  
Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktion jeglicher Art (Fotokopie, Nachdruck, Mikrofilm, Erfassung auf elektronischen Datenträgern oder andere Verfahren) nur mit schriftlicher Genehmigung des Verlags. Jegliche Haftung für die Richtigkeit des gesamten Werks kann, trotz sorgfältiger Prüfung durch Autor und Verlag, nicht übernommen werden. Die im Buch genannten Produkte, Warenzeichen und Firmennamen sind in der Regel durch deren Inhaber geschützt.

---

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>7</b>
<b>Kapitel 1: Einführung</b>	<b>9</b>
1.1 Grails 101	10
1.2 Installation	12
1.3 Ihre erste Grails-Applikation	13
1.4 Links & Literatur	17
<b>Kapitel 2: Struktur und Konfiguration</b>	<b>19</b>
2.1 Verzeichnislayout	19
2.2 Datenquellen	22
2.3 Bootstrapping	25
2.4 Logging	26
2.5 URL-Mapping	27
<b>Kapitel 3: Scaffolding</b>	<b>29</b>
3.1 Dynamisches Scaffolding	30
3.2 Statisches Scaffolding	35
3.3 Fazit	37
<b>Kapitel 4: Model – Grails ORM</b>	<b>39</b>
4.1 Benutzerdefiniertes Mapping	50
4.2 Validierung	50
4.3 Links & Literatur	54
<b>Kapitel 5: View</b>	<b>55</b>
5.1 Views und Layouts	56
5.2 Implizite Objekte in GSPs	60
5.3 Layouts	62
5.4 Grails Core Tags	63
5.5 Dynamic Tag Libraries	69

5.6	Links & Literatur	72
<b>Kapitel 6: Controller &amp; Services</b>		<b>73</b>
6.1	Controller	73
6.2	Interceptors	84
6.3	Datei-Uploads	87
6.4	Services	89
6.5	Fazit	91
6.6	Links & Literatur	92
<b>Kapitel 7: Testing</b>		<b>93</b>
7.1	Manuelles Testen	94
7.2	Unit-Testen mit junit	95
7.3	Funktionales Testen	102
<b>Kapitel 8: Grails &amp; AJAX</b>		<b>105</b>
8.1	Grails Ajax-Tags	106
8.2	Die render()-Methode	114
8.3	Groovy Builder & die Grails-render()-Methode	117
8.4	AJAX-Plug-ins	119
8.5	Fazit	120
8.6	Links & Literatur	120
<b>Stichwortverzeichnis</b>		<b>121</b>

---

# Vorwort

Grails ist ein modernes Web-Framework, mit dessen Hilfe Sie in kurzer Zeit J2EE-kompatible Webapplikationen erstellen können. Ähnlich wie das Framework Ruby on Rails, basiert Grails auf den zwei Grundsätzen *Don't Repeat Yourself (DRY)* und *Convention over Configuration (CoC)*. Im Klartext bedeutet dies: agile Webentwicklung und wieder Spaß an der Arbeit. Anstatt Zeit mit dem Schreiben von Konfigurationsdateien zu verschwenden, arbeiten Sie sich bereits am Domain-Modell ein, welches kurze Zeit später dank Scaffolding zum Leben erweckt wird.

Grails-basierte Webapplikationen sind J2EE kompatibel. Jeder Ihnen bekannte J2EE Web Application Server kann Ihre mit Grails erstellte Applikation ausführen. Wieso das wichtig ist? Skalierbarkeit, Performance, Ausgereiftheit der J2EE-Technologie – nicht gerade unwichtige Faktoren, die jedoch meist erst in den Mittelpunkt rücken, wenn die Applikation aus der Entwicklungs- und Testphase heraus ist und produktiv vom Kunden genutzt wird. Da Grails Gebrauch von der Groovy-Skriptsprache macht, bedeutet dies auch, dass sämtliche Java-Bibliotheken von Grails (genauer Groovy-Code) aus ansprechbar sind.

Dieses kompakte Buch möchte Ihnen die ersten Schritte mit Grails noch einfacher machen, als der Anfang mit Grails es ohnehin schon ist. Sie sollten dieses Buch problemlos an einem Nachmittag neben Ihrem offenen Laptop (um ein paar Beispiele auszuprobieren) lesen können. Insofern halten Sie gerade nicht die allumfassende Referenz zu Grails in der Hand, sondern vielmehr einen kompakten Begleiter, der Ihnen die Scheu vor diesem neuen Web-Framework nimmt. Wir setzten den Schwerpunkt ganz klar auf ein kompaktes Format, mit dem Sie die wichtigsten Aspekte von Grails schnell erkunden können.

Dank Grails können Sie in kürzester Zeit Ihre Ideen auf Basis von kleinen Prototypen umsetzen. Nutzen Sie diese Chance und konsumieren Sie nicht einfach Seite um Seite, sondern werden Sie selbst aktiv. Machen Sie mit! Wie Sie gleich sehen werden, schreiben Sie Ihre erste, kleine Grails-Applikation in wenigen Minuten.

Sven Haiges  
November 2007

## Einführung

1.1	Grails 101	10
1.2	Installation	12
1.3	Ihre erste Grails-Applikation	13
1.4	Links & Literatur	17

Die Webapplikations-Entwicklung mit bestehenden Java-Frameworks wie Struts oder JavaServer Faces wird von vielen Entwicklern mittlerweile als zu kompliziert und langwierig empfunden. Viele, die einen Blick über den Tellerrand hin zu Ruby (on Rails) und anderen Skriptsprachen und deren Frameworks geworfen haben, suchen nach einem Web-Framework, mit dem sie ähnlich effizient entwickeln können, ohne gleich sämtliche Java-Kenntnisse und die unbestrittenen Vorteile der Java-Plattform über Bord werfen zu müssen. Sie können sich freuen, denn Sie haben Grails gefunden!

In diesem ersten Kapitel erhalten Sie einen kurzen Überblick über die Kerntechnologien dieses Framework und Sie werden auch sogleich Ihre erste Grails-Applikation erstellen. Ein Tipp: Nehmen Sie sich die Zeit, um mit Grails zu experimentieren. Dieses Kapitel lädt Sie dazu ein.

Grails wurde zu Beginn stark von Ruby on Rails [2] inspiriert, geht aber in vielerlei Hinsicht mittlerweile konsequent seinen eigenen Weg. Beibehalten wurden jedoch die grundlegenden Prinzipien *Don't repeat yourself (DRY)* und *Convention over Configuration (CoC)*. Und richtig: Das „G“ in Grails steht für Groovy [3],

eine mächtige und für Java-Entwickler einfach zu erlernende Skriptsprache, die zahlreiche dynamische Features dank ihres Meta-Object-Protocol (MOP) ermöglicht.

## 1.1 Grails 101

Obwohl ich nicht lange um den heißen Brei herumreden möchte, sind ein paar Infos zur Entstehungsgeschichte dieses Framework sowie zu dessen Kerntechnologien für das Verständnis sicherlich hilfreich. Grails begann mit einer E-Mail-Diskussion zwischen Guillaume LaForge (Groovy-Projektleiter), Steven Devijver und Graeme Rocher (aktueller Grails-Projektleiter). Aus dem Enthusiasmus heraus, etwas Ähnliches wie Ruby on Rails mit der Groovy-Skriptsprache zu verwirklichen, entstand Ende März 2006 das 0.1 Release von Grails. Die wesentlichen Eckpfeiler dieses Framework sind allesamt robuste und wohlbekanntere Technologien:

- Groovy: Dank der Groovy-Skriptsprache genießt Grails eine Integration erster Klasse mit sämtlichem Java-Code sowie mit der Java-Plattform selbst. Mit Groovy können Sie problemlos und direkt auf sämtliche Java-Elemente zugreifen. Groovy wurde speziell für die JVM entwickelt.
- Spring: Grails benutzt das Spring Framework [4] gleich in mehrerer Hinsicht. Zunächst wird Spring's Dependency-Injection-Mechanismus eingesetzt. Des Weiteren wird Spring MVC als stabiles und gleichzeitig sehr flexibles Web-Framework benutzt. Seit Grails 0.6 wird auch Spring Webflow eingesetzt.
- Hibernate: Wenn ein Persistenz-Framework den Titel „De-facto-Standard“ verdient hat, dann ist es Hibernate [5]. Die transparente Persistenz für POJOs wurde sogar vom EJB3-Standard weitgehend übernommen und ist unangefochten ein sehr robuster OR-Mapper.

- SiteMesh: SiteMesh [6] ist ein flexibles Layout-Framework, welches bei Grails out-of-the-box zum Einsatz kommt.

Diese Aufzählung ist mit Sicherheit beeindruckend; viel beeindruckender ist jedoch, dass Sie diese Technologien im Normalfall *nicht* konfigurieren müssen. Denken Sie mal darüber nach, wie viel Zeit Sie schon damit verbracht haben, Hibernate Mapping Files (HBM-Files) zu schreiben. Oder die 1000 Zeilen XML-Konfiguration für Ihr letztes Spring-Projekt. Sie können dank Grails Spring's Dependency Injection benutzen oder Ihr Domain-Modell persistieren, ohne eine Zeile XML-Konfiguration schreiben zu müssen. Das ist ein erster Vorgeschmack auf „Convention over Configuration“. Anhand von Konventionen (z.B. Benennung der Controller-Klassen) kann Grails die nötige Konfiguration für Sie erstellen. Das Ganze passiert transparent im Hintergrund, ohne Sie von der eigentlichen Arbeit abzuhalten. Sie konzentrieren sich auf die Geschäftslogik Ihrer Applikation.

Allerdings können wir uns leider oftmals nicht den Luxus leisten, ein bestehendes DB-Modell zu ignorieren. In vielen Projekten ist es immens wichtig, eine gute Integration mit bestehenden Elementen wie Datenbanken, bestehenden Spring-Artefakten oder Java-APIs herzustellen. Auch hier zeigt Grails seine Stärken: Sie *können* für Ihr Domain-Modell eigene Hibernate-Mappings erstellen. Ebenso *können* Sie bestehende Spring-Konfigurationen einfach übernehmen oder jederzeit bestehende Java-APIs aus Grails heraus benutzen.

Was Sie an dieser Stelle verstanden haben sollten: Grails eignet sich hervorragend für neue Webprojekte. Die meisten Konfigurationsdetails werden für Sie transparent generiert, Sie können in kürzester Zeit einen lauffähigen Prototypen erstellen. Aber: Grails integriert sich – auf Wunsch – auch erstklassig mit bestehenden Elementen.

## 1.2 Installation

Probieren wir es aus. Wir nutzen dieses Kapitel, um gemeinsam eine erste, einfache Grails-Applikation zu erstellen. Gehen Sie zunächst auf [grails.org/Download](http://grails.org/Download) und laden Sie das neueste, stabile Release herunter. Mit großer Sicherheit wird dies zum Erscheinen dieses Buchs bereits die Version 1.0+ sein, da Grails sich sehr rasch entwickelt. Die Beispiele sollten dennoch gültig sein. Bei Problemen rund um Grails können Sie übrigens auch jederzeit auf die Grails User List zurückgreifen; die Installation ist auch unter [grails.org/Installation](http://grails.org/Installation) detailliert erklärt.

Extrahieren Sie nun das Archiv, z.B. nach `C:\grails-1.0` oder `/home/user/grails-1.0`. Erstellen Sie eine `GRAILS_HOME`-Umgebungsvariable, die auf das eben erstellte Verzeichnis zeigt, und fügen Sie das `GRAILS_HOME/bin`-Verzeichnis Ihrer Path-Variablen hinzu (unter Windows mit `%GRAILS_HOME%\bin` und unter Linux per `$GRAILS_HOME/bin`). Unter Linux müssen Sie nun noch zwei Dateien ausführbar machen, wenn Sie nicht das `tar.gz`-Archiv heruntergeladen haben:

```
chmod u+x $GRAILS_HOME/bin/grails
$GRAILS_HOME/ant/bin/ant
```

Auf der Kommandozeile können Sie nun „grails“ eintippen und bestätigen. Sie sollten einen Überblick über die Grails-Kommandos bekommen. Herzlichen Glückwunsch, Sie sind nur noch ein paar Zeilen von Ihrer ersten Grails-Applikation entfernt.