

# Pro Spring MVC:

With Web Flow

**Marten Deinum**  
**Koen Serneels**

with Colin Yates, Seth Ladd,  
and Christophe Vanfleteren

Foreword by Erwin Vervaet,  
Spring Web Flow project founder

**Apress®**

## Pro Spring MVC

Copyright © 2012 by Marten Deinum and Koen Serneels with Colin Yates, Seth Ladd, and Christophe Vanfleteren

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN 978-1-4302-4155-3

ISBN 978-1-4302-4156-0 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Steve Anglin

Technical Reviewers: Manuel Jordan

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Jennifer L. Blackwell and Stephen Moles

Copy Editors: Patrick Meader and James Compton

Compositor: Bytheway Publishing Services

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

Any source code or other supplementary materials referenced by the author in this text is available to readers at [www.apress.com](http://www.apress.com). For detailed information about how to locate your book's source code, go to [www.apress.com/source-code](http://www.apress.com/source-code).

*To my wife for your love and endless support.*

*–Marten Deinum*

*To Sonja Korte, die große liebe meines lebens, thank you for your patience and support. You almost made me believe in automatically refilling coffee mugs. To mum and dad, for always being there for me and buying an 80386 instead of that 8bit NES.*

*–Koen Serneels*

# Contents at a Glance

■ Foreword.....	xvi
■ About the Authors.....	xviii
■ About the Technical Reviewer .....	xx
■ Acknowledgments .....	xxi
■ Introduction .....	xxiii
■ Chapter 1: Configuring a Spring Development Environment .....	1
■ Chapter 2: Spring Framework Fundamentals.....	25
■ Chapter 3: Web Application Architecture.....	51
■ Chapter 4: Spring MVC Architecture.....	65
■ Chapter 5: Implementing Controllers.....	107
■ Chapter 6: Implementing Controllers — Advanced .....	177
■ Chapter 7: REST and AJAX.....	215
■ Chapter 8: Resolving and Implementing Views .....	237
■ Chapter 9: Testing Spring MVC Applications .....	273
■ Chapter 10: Spring Web Flow .....	321
■ Chapter 11: Building Applications with Spring Web Flow .....	373
■ Chapter 12: Advanced Spring Web Flow.....	429
■ Chapter 13: Spring Security.....	477
■ Appendix: Cloud Foundry: Deploying to the Cloud.....	535
■ Index .....	555

# Contents

■ Foreword.....	xvi
■ About the Authors.....	xviii
■ About the Technical Reviewer .....	xx
■ Acknowledgments .....	xxi
■ Introduction .....	xxiii
■ Chapter 1: Configuring a Spring Development Environment .....	1
Prerequisites.....	1
Java Development Kit.....	2
Servlet Container .....	2
Integrated Development Environment.....	2
The Sample Application .....	2
A Bookstore Sample Application .....	3
The Build System.....	4
Building the Sample Application.....	6
Deploying the Sample Application.....	8
SpringSource Tool Suite (STS).....	10
Configuring STS for Gradle projects .....	10
Importing the Sample into STS.....	12
Running the Application on the SpringSource vFabric tc Server.....	17
Editing the Application.....	19
Summary .....	22

■ <b>Chapter 2: Spring Framework Fundamentals</b> .....	<b>25</b>
The Spring Framework .....	25
Dependency Injection .....	29
ApplicationContexts .....	34
Resource Loading .....	38
Component-Scanning .....	40
Scopes .....	41
Profiles .....	41
Enabling Features .....	44
Aspect-Oriented Programming .....	45
Web Applications .....	47
Summary .....	50
■ <b>Chapter 3: Web Application Architecture</b> .....	<b>51</b>
The MVC Pattern .....	51
Application Layering .....	53
Separation of Concerns .....	56
Spring MVC Application Layers .....	56
The Domain Layer .....	56
The User Interface Layer .....	57
The Web Layer .....	58
The Service Layer .....	59
The Data Access Layer .....	62
More Roads to Rome .....	63
Summary .....	64
■ <b>Chapter 4: Spring MVC Architecture</b> .....	<b>65</b>
DispatcherServlet Request Processing Workflow.....	65
The Workflow.....	66

The Request Processing Summary.....	73
<b>The DispatcherServlet .....</b>	<b>74</b>
Bootstrapping the DispatcherServlet.....	74
Configuring the DispatcherServlet.....	78
<b>The Spring MVC Components .....</b>	<b>88</b>
HandlerMapping .....	88
HandlerAdapter.....	94
MultipartResolver .....	96
LocaleResolver .....	97
ThemeResolver .....	99
HandlerExceptionResolver.....	101
RequestToViewNameTranslator .....	102
ViewResolver .....	103
FlashMapManager .....	104
Summary .....	105
<b>■ Chapter 5: Implementing Controllers.....</b>	<b>107</b>
Introducing Controllers .....	107
Interface-based Controllers.....	108
Annotation-based Controllers.....	110
Configuring View Controllers.....	111
Request-Handling Methods .....	112
Supported Method Argument Types.....	115
Supported Method Argument Annotations .....	118
Supported Method Return Values.....	123
Writing Annotation-based Controllers.....	125
A Simple Login Controller .....	125
Book Search Page .....	130
Book Detail Page .....	133

Data Binding .....	136
Customizing Data Binding .....	137
Per Controller Customization .....	139
ModelAttribute .....	141
Type Conversion .....	148
Validating Model Attributes .....	162
Internationalization .....	170
Message Source .....	170
LocaleResolver .....	172
LocaleChangeInterceptor .....	172
Summary .....	176
■ <b>Chapter 6: Implementing Controllers — Advanced .....</b>	<b>177</b>
Using Scoped Beans .....	177
Adding Something to the Cart .....	178
Implementing the Checkout .....	183
Crosscutting Concerns .....	185
Interceptors .....	186
Exception Handling .....	197
SimpleMappingExceptionHandler .....	201
Extending Spring @MVC .....	205
Extending RequestMappingHandlerMapping .....	205
Extending the RequestMappingHandlerAdapter .....	207
Using the RequestDataValueProcessor .....	212
Summary .....	213
■ <b>Chapter 7: REST and AJAX .....</b>	<b>215</b>
Representational State Transfer (REST) .....	215
Identifying Resources .....	215



Working with Resources.....	216
<b>Asynchronous JavaScript and XML (AJAX).....</b>	<b>220</b>
Adding AJAX to Our Application.....	221
Combining AJAX and REST.....	226
Progressive Enhancement.....	227
<b>Handling File Uploads.....</b>	<b>228</b>
Configuration .....	228
Request Handling Method for File Upload .....	231
Exception Handling.....	235
<b>Summary .....</b>	<b>235</b>
<b>■ Chapter 8: Resolving and Implementing Views .....</b>	<b>237</b>
View Resolvers and Views.....	237
View Resolvers .....	238
BeanNameViewResolver.....	239
XmlViewResolver .....	240
ResourceBundleViewResolver.....	241
UrlBasedViewResolver.....	243
InternalResourceViewResolver .....	245
XsltViewResolver .....	245
ContentNegotiatingViewResolver .....	246
Implementing Your Own ViewResolver.....	247
View Technologies.....	248
Java Server Pages.....	249
JavaServer Faces .....	249
Tiles.....	250
Velocity and FreeMarker.....	254
PDF .....	263
Excel.....	265

XML and JSON .....	268
JasperReports .....	270
Summary .....	272
<b>■ Chapter 9: Testing Spring MVC Applications .....</b>	<b>273</b>
Introducing Testing.....	273
Why Should I Bother Writing Tests? .....	273
Promoting Testing Within Your Project.....	276
Different Types of Testing .....	277
Setting Up a Basic Unit Test .....	280
Testing Code Coverage .....	284
Using Spring's Test Support .....	286
Setting Up Our Integration Test .....	286
Testing the JpaBookRepository .....	292
Using Mock Objects .....	294
What Are Mock Objects? .....	295
Testing the AccountService .....	296
Testing Your MVC Logic .....	301
Using Spring Mock Objects .....	302
Introducing Spring MVC Test .....	305
Automated Front-End Testing.....	309
Front-End Testing Using Selenium .....	310
Writing a Selenium Test .....	312
Using the Selenium IDE .....	314
Running the Front-End Tests via Gradle .....	318
Summary .....	319
<b>■ Chapter 10: Spring Web Flow .....</b>	<b>321</b>
Why Web Flow .....	321

The Flow Concept .....	322
Fine-Grained Scoping .....	323
Automatic State Management .....	326
Request Synchronization .....	327
Post Redirect Get (PRG) .....	330
Controlled Navigation .....	331
When to Avoid Web Flow .....	332
<b>The Basic Ingredients of a Flow .....</b>	<b>332</b>
Flow .....	333
The View State .....	338
State Transitions .....	341
The Evaluate Action .....	342
Expressions .....	343
<b>Configuration .....</b>	<b>345</b>
Dependencies .....	346
Web Flow Configuration .....	346
Gluing Spring MVC and Spring Web Flow .....	350
<b>Building Your First Flow .....</b>	<b>353</b>
Creating the Home Page .....	355
Implementing the Create Order Flow .....	356
Overview .....	371
<b>Summary .....</b>	<b>372</b>
<b>■ Chapter 11: Building Applications with Spring Web Flow .....</b>	<b>373</b>
<b>Important Web Flow Concepts .....</b>	<b>373</b>
Flow Definition .....	373
Different Web Flow Scopes .....	374
Implicit Objects .....	379
<b>Enhancing the Bookstore .....</b>	<b>381</b>

Selecting the Category .....	381
Selecting Books and Delivery Options.....	387
Form Validation Using JSR 303 Annotations .....	388
<b>Setting Variables and Accessing Scopes.....</b>	<b>400</b>
Flow Variables .....	401
Accessing Scoped Variables from Views.....	402
Programmatically Accessing Scopes .....	402
<b>Controlling Action Execution.....</b>	<b>404</b>
<on-start> .....	404
<on-end> .....	404
<on-entry> .....	404
<on-exit> .....	405
<on-render>.....	405
Controlling Action Execution: Sub-elements .....	405
<b>Global Transitions .....</b>	<b>406</b>
<b>Subflows .....</b>	<b>406</b>
<b>Further Enhancing the Bookstore .....</b>	<b>408</b>
Implementing Authentication as a Subflow.....	410
Decision State.....	416
Action State .....	417
Working with Outcome Events .....	419
Overview.....	420
<b>Subflow Input/Output Mapping.....</b>	<b>421</b>
<b>Creating the Order Process as a Subflow .....</b>	<b>423</b>
<b>End State .....</b>	<b>425</b>
<b>Summary .....</b>	<b>427</b>

■ <b>Chapter 12: Advanced Spring Web Flow</b> .....	<b>429</b>
Inheritance.....	429
Flow Inheritance.....	430
State Inheritance.....	434
Web Flow Configuration Customizations.....	434
Execution and Conversation Snapshots.....	435
Changing the Expression Parser.....	436
Web Flow 1 Migration.....	437
Exception Handling.....	439
The On Exception Transition.....	439
Custom Exception Handler.....	440
Explicit Form Binding.....	443
Web Flow AJAX Support.....	445
Configuring Web Flow for AJAX.....	447
Preparing the View.....	447
Adjusting the Flow.....	453
Adding AJAX to the View with Spring JS and JQuery.....	454
Flow Execution Listeners.....	459
Writing Flow Execution Listeners.....	459
Flow Execution Listener Methods.....	460
Flow Managed Persistence Context.....	463
From Database to View.....	463
Prolonging the Persistence Context.....	465
Applying Flow Managed Persistence Context.....	466
Reworking the Orders Overview.....	468
Summary.....	475

■ <b>Chapter 13: Spring Security</b> .....	<b>477</b>
Introducing Security .....	478
What Is Application Security? .....	478
General Security Principles .....	478
What We Will Cover .....	479
Preparing the Example Application.....	480
Securing Our Bookstore.....	482
Adding the Right Dependencies .....	482
Enabling Spring Security .....	484
Defining Which Resources to Secure .....	489
Configuring Access to Resources.....	490
Configuring Authentication.....	494
Putting It All Together .....	497
The Complete Security Configuration .....	503
Going to the Database .....	504
Securing Our Flows, the Right Way .....	506
Adding Access Attributes to Your Flows.....	507
Configuring the SecurityFlowExecutionListener.....	508
Transport Security .....	510
Localization.....	516
Role-Based Access Control.....	517
Authorizing Access .....	523
Using Tag Libraries in Our Pages .....	523
Using Annotations in Our Code .....	531
Summary .....	533
■ <b>Appendix: Cloud Foundry: Deploying to the Cloud</b> .....	<b>535</b>
Cloud Computing .....	535

Cloud Foundry.....	536
Deploying Our Application .....	537
Installing the Cloud Foundry Plug-in .....	537
Making Some Adjustments.....	538
Deploying.....	544
Configuring the Services .....	547
How Does It Work? .....	551
Other Configuration Options .....	551
Deploying Locally.....	552
Debugging with Cloud Foundry.....	552
Summary .....	554
■ <b>Index .....</b>	<b>555</b>

# Foreword

I still remember first learning about the Spring framework back in 2004. I had been using J2EE and Struts heavily, and had struggled with many difficulties in effectively using those technologies while building Java enterprise applications. Instead of trying to do away with J2EE altogether, Spring tried to make using J2EE drastically simpler and more productive by offering a large collection of best practice implementations and an inversion of control container gluing it all together. An exciting proposition indeed!

While in general Spring did not try to reinvent the wheel by providing solutions competing with existing parts of J2EE or other established frameworks (notably Hibernate), there was one important exception to this rule: web application development. Spring shipped with its own web framework, Spring MVC, a fully functional web application framework that can serve as a direct alternative to something like Struts. At the time, I saw further opportunity to enhance Spring-based web application development by adding a page flow component to Spring MVC called Spring Web Flow. Spring MVC and Spring Web Flow are the prime subjects of this book.

The design of Spring MVC benefited from the lessons learned working with earlier frameworks. Flexibility and long-term productivity were core design goals. Well-designed Spring MVC applications can grow and change while remaining manageable and maintainable. Still, all was not good. Many found the initial learning curve when adopting Spring, and more specifically Spring MVC, too steep. New frameworks such as Ruby on Rails popped up and focused on short term-productivity and making it very easy to get up-and-running. The Spring developer community recognized this shortcoming, and the recent 3.0 and 3.1 releases of Spring MVC have largely addressed it: extensive use of convention-over-configuration and annotations have made the framework easier to use than ever, as this book will demonstrate! It is a real testament to the flexibility and design quality of Spring MVC that all of this has been possible while remaining 100% backward-compatible. You now have a framework offering the best of both worlds: easy to get started with while the fundamental principles underpinning it all pay dividends in the long term. Spring Web Flow is currently undergoing a similar evolution.

Since 2004, Spring MVC and Spring Web Flow have seen a steady and continued increase in popularity. They are now mature and well established Java development frameworks. If you build a Java web application on top of these frameworks, you can rest assured that you are building on a very solid foundation!

In this book, Marten and Koen take a practical approach to introducing Spring MVC and Spring Web Flow. Of course, they help you in setting up a productive development environment and guide you while getting started developing with Spring MVC and Web Flow. But they don't stop there. I really appreciate how they did not simply cover the technical details, but took the time to explain many of the underlying concepts, bringing a deeper understanding to you, the reader. Furthermore, the book also covers other important topics such as enforcing security constraints and making sure your web applications are well tested. After having read this book you should be well prepared to develop real-life web application based on Spring technologies.



I'm excited to see that to this day people are still benefiting from Spring and the small component I added to it back in 2004. I highly recommend this book for all those eager to learn about Spring MVC and Spring Web Flow, and I applaud the excellent job the authors have done in helping developers learn about these exciting Spring technologies.

Erwin Vervaet  
Spring Web Flow project founder

# About the Authors



■ **Marten Deinum** is a Java/software consultant working for Conspect. He has developed and architected software, primarily in Java, for small and large companies. He is an enthusiastic open source user and longtime fan, user and advocate of the Spring Framework. He has held a number of positions including Software Engineer, Development Lead, Coach, and also as a Java and Spring Trainer. When not working or answering questions on the Spring Framework Forums, he can be found in the water training for the triathlon or under the water diving or guiding other people around.



■ **Koen Serneels** is a Senior JAVA software engineer with several IBM, Cisco, and Oracle certifications. For more than 10 years he has developed enterprise solutions using Java(EE), Spring, Spring MVC, Web Flow, JSF, and Hibernate. He holds a keen interest in system architecture and integration, data modeling, relational databases, security and networks. Beginning his career with the Belgian federal government, Koen developed highly transactional Java-based applications with legacy integration. Currently, he is employed by Hewlett-Packard and is a Java software consultant and technical lead for the Flemish government in Belgium. He also teaches a graduate course on “Software Development with Java” at the Groep T Engineering School in Leuven, Belgium.

■ **Colin Yates** is a J2EE principal architect who specializes in web-based development. He has been a freelance consultant for the past three years and has worked in a number of environments, both structured and chaotic. Since graduating with a software engineering degree in 1997, he has held a number of positions, including development lead, principal systems engineer, mentor, and professional trainer. His principal skill set includes mentoring others, architecting complex problems into manageable solutions, and optimizing development processes.

Colin was first introduced to the Spring Framework in January 2003 by his mentors, Peter Den Haan and David Hewitt, and he has never looked back.

■ **Seth Ladd** is a software engineer and professional Spring Framework trainer and mentor specializing in object-oriented and testable web applications. He started his own company building websites at age 17, but now enjoys having a real job. Currently working for Camber Corporation, Seth has built and deployed systems for NEC, Rochester Institute of Technology, Brivo Systems, and National Information Consortium. He has architected and developed enterprise applications in Java and C for both the server and remotely connected embedded devices. He enjoys speaking and teaching, and is a frequent presenter at local Java user groups and at corporate developer conferences. Seth is very thankful for living and working in Kailua, Hawaii, with his wife.

■ **Christophe Vanfleteren** is a software engineer who has been working with Java technologies since 2001. Working for EDS / HP, he has mostly worked on projects for the Flemish government in Belgium, including some large-scale, event-driven applications like the one tracking the Flemish elections in 2006 and the Flemish fiscal platform, which collects several different taxes from 6 million people. He is able to work all the way up the stack, from heavy transactional code all the way to JavaScript-based front-ends. Since 2010, Christophe has been an independent contractor.

# About the Technical Reviewer



■ **Manuel Jordan Elera** is an autodidactic developer and researcher who enjoys learning new technologies for his own experiments and creating new integrations.

Manuel won the 2010 Springy Award – Community Champion. In his little free time, he reads the Bible and composes music on his guitar. Manuel is a Senior Member in the Spring Community Forums, where he is known as `dr_pompeii`.

Manuel is the Technical Reviewer for these books (All published by Apress):

*Pro SpringSource dm Server*, by Gary Mak, Daniel Rubio (2009)

*Spring Enterprise Recipes*, by Gary Mak, Josh Long (2009)

*Spring Recipes*, by Gary Mak, Daniel Rubio, Josh Long (Second Edition, (2010)

*Pro Spring Integration*, by Dr. Mark Liu, Mario Gray, Andy Chan, Josh Long (2011)

*Pro Spring Batch*, by Michael T. Minella (2011)

*Pro Spring 3*, by Clarence Ho, Rob Harrop (2012)

Read and contact him through his blog at <http://manueljordan.wordpress.com/> and follow him on his Twitter account, `@dr_pompeii`.

# Acknowledgments

One thing I quickly learned during this endeavor is that writing a book is not something you do alone. I would like to thank the whole team at Apress for their support; without you this book would have never seen the light of day. I also learned that writing a book is harder than I ever expected. Nevertheless, it was quite an experience. I thank Manuel Jordan for introducing me to the people at Apress, which started this endeavor.

I owe a big thanks to Koen Serneels for helping me out writing this book and sitting it out all the way, you wrote more than you, initially, signed up for. Thanks for helping me out Koen!

Another big thanks goes to Chris Nelson for keeping all of us focused and for his advice on the book; it was a pleasure working with you. The same thanks go to the coordinating editors, Stephen Moles and Jennifer Blackwell, who in addition to the focus on quality tried to keep everything within the schedule (sorry for slipping past the initial date).

This book would never have seen the light without the comments and suggestions given by Manuel Jordan and Erwin Vervaeke. Although at times the comments drove me to desperation, without your comments and suggestions this book would never have become what it is now. So thanks for the comments and suggestions and keeping a clear vision.

The appendix was written entirely by Christophe Vanfleteren, who took on this task with great dedication and determination. He did an excellent job of making our application deploy on Cloud Foundry and writing down the steps it took. Thanks, Christophe, for fulfilling this task.

Thanks also to my family and friends for the times they missed out on me, and to my dive-buddies for all the dives and trips I missed in the last months.

Last but definitely not least I thank my wife, Djoke Deinum, for her endless support, love and dedication, despite the long evenings, sacrificed weekends and holidays to finish the book. Without your support I probably long ago would have abandoned the endeavor.

Marten Deinum

I thank Erwin Vervaeke for giving me the chance to coauthor this book and for being a great colleague and mentor over the last years. With Erwin on your side there are no problems, only solutions. Thank you Erwin.

Thanks to Marten Deinum for having me as coauthor and Chris Nelson, Manuel Jordan, Jennifer Blackwell and Stephen Moles for their endless dedication and advice. Without you guys this project would never have reached this paper version; thank you all. After all these months of hard work I realize that writing a book is even harder than writing good software. In fact, writing a book is comparable to a software project without the luxury of automated testing.

Christophe Vanfleteren did an excellent job investigating Cloud Foundry and helping us out writing the appendix. I had the pleasure to work together with Christophe before, and he is by far one of the most talented developers I have worked with in my career. Thank you Christophe for your time and dedication.

Special thanks to everyone who keeps Spring alive, by developing for it or by using it. I have been using Spring since 2004, and to this day it still keeps me amazed. I have used many frameworks, but Spring remains a top-notch framework with its superior code base, agility, liveness, good design and completeness. It is my believe that Spring has many years to go and will be able to serve us on a daily basis to make our work easier, faster and help us create quality projects.

I thank my girlfriend, Sonja Korte, for bearing with me and forgiving me sacrificing our holidays and nearly all evenings and weekends over the last 4 months, and my family and friends for missing out on me.

Last but not least, I want to thank you for reading this book. I'm also a reader of many IT books, so I know what is good and what isn't. I really hope we have succeeded at bringing the good stuff into practice and offer you a top-quality book with exiting technology that will make your daily job easier.

Koen Serneels

# Introduction

Welcome to the first edition of *Pro Spring MVC with Web Flow*, the first Pro Spring book focused entirely on web development using the Spring Framework 3.1 ecosystem.

## What This Book Will Teach You

This book will teach you everything you need to know in order to get started building enterprise-quality web applications using version 3.1 of the Spring Framework. Topics include but are not limited to:

- The building blocks of the Spring MVC components
- Configuring your development environment
- Providing a web front end to a Spring based application
- A pragmatic approach to testing the web front end
- Deploying to a local web server and to a remote cloud-based deployment platform
- An introduction to Spring Web Flow
- How to build applications with Spring Web Flow

After reading this book you will be familiar with the Spring MVC toolkit and capable of building your own web application from scratch or providing a new web interface to an existing application.

All too often trivial examples are used to demonstrate the power of a framework; it is only when you start using the framework in real situations that its limitations appear. The intention of this book is to show how Spring MVC answers those hard questions (as well as the easy questions!) that web developers are faced with, like managing state in non-trivial multi-page use cases, for example, or providing multiple views of the same resource for different consumers.

And because the real-world problems this book tackles are hard, sometimes the answers Spring MVC provides are not as easy as one would like. This book will not shy away from highlighting those issues or providing pragmatic advice on how to do the right thing.

## Who Is This Book For?

This book is for those who are familiar with Spring and want to gain an in-depth understanding of Spring MVC. While it is primarily aimed at those new to Spring MVC, there is information here that will take even expert MVCers by surprise!

The typical reader will be a web developer who has some understanding of the core Spring framework (after reading *Pro Spring* for example) and wants to investigate Spring MVC in more detail.

If you are unfamiliar with Spring, then by all means continue reading; particularly Chapter 2. However if you find that is insufficient then you might find the Spring Reference Guide<sup>1</sup> or *Pro Spring 3* (Apress, 2012).

The original Spring book, *Expert One-on-One J2EE Design and Development* (Wrox, 2002)<sup>2</sup>, by Rod Johnson (the "father" of the Spring framework), is fairly old now but still relevant and full of wisdom.

## How to Read This Book

This book will take the reader through the thought process of designing, implementing, and deploying a Java web application. The order of the chapters follows the chronological order defined by the development lifecycle. During these chapters we use a sample application to illustrate the topics discussed in the chapter.

Each chapter will address a real-world problem by introducing a new concept or capability which is then used to upgrade the sample application.

---

■ **Note** There is a never-ending dilemma that authors face: should we show the answers first and then the questions, or show the questions and then the answers? The first approach risks overloading the reader, while the second approach can be frustrating and slow for those already familiar with the subject matter. The authors believe that the pace of this book is sufficient for those unfamiliar with Spring MVC; more experienced readers may want to skip certain chapters. For example, if you already have a development environment configured and are familiar with Spring then you can skim Chapter 1 to get hold of the sample application and then jump straight to Chapter 3.

---

## The Application Example

While the example cannot be a true enterprise application (because we need to keep this book under 10K pages ;)), it is realistic enough to highlight the typical design challenges that are faced in real-world projects.

The example in question is the site for a web-enabled bookstore. It offers users the following capabilities:

- An anonymous user can browse the site.
- A logged-in user can submit an order for one or more books.

---

<sup>1</sup> <http://static.springsource.org/spring/docs/current/spring-framework-reference/html/>

<sup>2</sup> <http://www.wrox.com/WileyCDA/WroxTitle/productCd-0764543857.html>



- A logged-in user can see his previous orders.
- An author can manage the books.
- An administrator can manage the categories

Throughout the rest of this book, each chapter adds a new aspect to the sample application. The simplest way of capturing this in the example was to have a new project for each chapter.

## The Sample Code

We need to explain a little about the coding and referencing style used throughout the book. You'll find that some of the code listings are complete and can be taken directly from the book to the development environment. In others we omit some imports or methods that are mentioned in earlier listings; we've marked those with `// Imports omitted` or `// Methods omitted`. Most include a reference to the code listing in which the methods or imports can be found. This approach was needed for readability, as some code listings contain only a few new lines and others would span multiple pages.

Another thing to mention is that we did not include the fully qualified package names for common classes like `java.lang.String` or `java.util.HashMap` (most of the `java.util` packages aren't included), as a Java developer in general knows where to find those classes. Also, we only mention the fully qualified classname on the first significant reference (after the main heading); for example, the package first identified as `org.springframework.web.servlet.DispatcherServlet` will in subsequent references be called simply `DispatcherServlet`. This is again for readability, but we also feel the need to show you the location of the class and believe this was an acceptable approach.

## How This Is Book Structured

The book consists of a series of chapters, each explaining a part of the framework or how to use a certain technology. The first four chapters are quite theoretical and are used to explain some of the more general concepts and how they apply or work within Spring MVC (the same goes for the start of the Spring Web Flow part of the book; Chapter 10 is also more or less theoretical).

After that introduction to the concepts behind Spring MVC, the remaining chapters follow a more practical and hands-on approach as we start to develop an application.

The chapters are:

- Chapter 1, “Configuring a Spring Development Environment.” This chapter presents the prerequisites for your development environment, which you can use to develop the sample application. The structure and purpose of the sample application (an online bookstore) are also explained in this chapter.
- Chapter 2, “Spring Framework Fundamentals,” provides a broad overview of the fundamental building blocks of the Spring framework. It introduces the concepts of Dependency Injection (DI) and Inversion of Control (IoC). This chapter is particularly recommended if you are unfamiliar with Spring.
- Chapter 3, “Web Application Architecture.” In this chapter we take you on a slight detour to explain web application architecture. We will explain the different layers that (generally) make a web application and we will explain the Model View Controller triad.

- Chapter 4, “Spring MVC Architecture.” This chapter is the first “down and dirty” chapter dealing with Spring MVC. It defines exactly what MVC is, how web applications are typically structured, or layered, and it dives into the powerhouse of the Spring MVC engine: the wonderful `DispatcherServlet`.
- Chapter 5, “Implementing Controllers.” At this point, you’re probably ready to say, “Show me the code!” In this chapter we will take the knowledge from the previous chapters and start writing controllers; we’ll also get more insight into the internals of Spring @MVC. At the end we will have the start of the sample application. Exciting!
- Chapter 6, “Implementing Controllers – Advanced.” Every application requires the same behavior in a number of different places, as well as behavior that isn’t really part of your core application but needs to be stuck in somewhere. This chapter will introduce Aspect Oriented Programming (AOP) and how Spring MVC easily solves some common web problems. We will also explore the internals of Spring MVC a bit more and explain how to extend the existing infrastructure and how to tailor it to our needs.
- Chapter 7, “REST and AJAX.” Now that our bookstore is taking off, we want to add some nifty behavior to our application and we also want to expose our controllers as REST web services so that others might be able to integrate with us. For this we are going to explore REST and AJAX and apply those techniques to our application.
- Chapter 8, “Resolving and Implementing Views.” Chapter 8 digs a bit deeper into how views are resolved within Spring MVC and builds on the information you’ve learned so far. In this chapter you will get to revisit the `ViewResolver` infrastructure and start to see the power of the MVC architecture shine through when you re-use the same infrastructure to provide different renditions of the same model.
- Chapter 9, “Testing Spring MVC Applications.” Now that the craving to get something done has been somewhat satisfied and we’ve written some code, it is time to look at testing. This chapter explains *how* to test Spring MVC applications but also crucially *what* to test. This chapter will include a good discussion of different strategies for testing, including how to ensure you are testing only what you need to test. You will also be given the chance to test-drive the HTML as well!
- Chapter 10, “Spring Web Flow.” Up to now all of the page interactions have been pretty simple; each use-case was one or two pages. Now the manager wants to introduce some nontrivial use-cases. Chapter 10 introduces Spring Web Flow— a companion partner to Spring MVC that provides some pretty nifty features for managing web conversations with clients.
- Chapter 11, “Building Applications with Spring Web Flow.” Following the previous chapter’s introduction to Spring Web Flow and the problems it helps to solve, Chapter 11 explores in practice how to build an application with Spring Web Flow.
- Chapter 12, “Advanced Spring Web Flow,” builds on the information gained in Chapter 11 and demonstrates how you can dig a little deeper into Spring Web Flow and find some real gems.

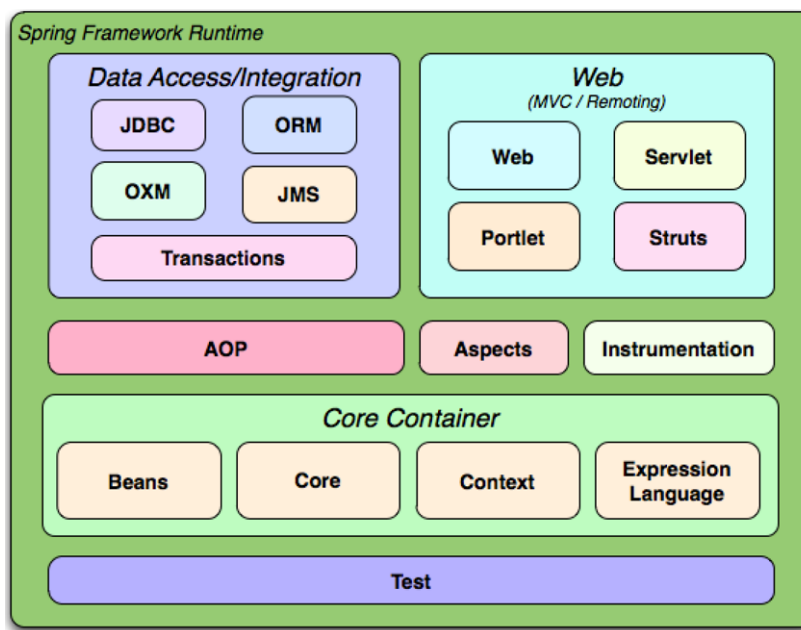
- Chapter 13, “Spring Security,” shows how to keep the scruffy hackers out of our web-application through the use of another well-established tool in the Spring toolbox, Spring Security.
- Appendix A, “Cloud Foundry—Deploying to the Cloud.” In this appendix we explain the steps needed to deploy the application to the cloud, especially to Cloud Foundry, as that integrates seamlessly with our chosen development environment.

## A Thousand-Mile View of the Spring Ecosystem

So before going any further let's take the first peek at Spring MVC and where it fits into the existing Spring ecosystem.

The first piece of good news is that Spring MVC *is* Spring. You configure Spring MVC using the existing powerful Spring container. Beans defined in Spring MVC are just like any other beans.

The following image from <http://static.springsource.org/spring/docs/3.1.0.M2/spring-framework-reference/html/overview.html> is very helpful:



As you can see— Spring MVC is powered by the rest of Spring!

## Onward!

Now that you have an idea about the style and purpose of this book let's waste no more time in getting you set up with a development environment. We'll see you in Chapter 1.

## Contacting the Authors

Marten Deinum can be contacted at [marten@deinum.biz](mailto:marten@deinum.biz) for queries and suggestions for this book. His blog can be found at <http://mdeinum.wordpress.com>. Feel free to contact him if you have found a mistake, want to ask a question, or want to discuss anything else related to the book.

Koen Serneels can be contacted at [koen.serneels@error.be](mailto:koen.serneels@error.be) for anything related to this book. If you have suggestions, remarks, or questions, or have found something to be inaccurate, feel free to drop him a message. His blog can be found at <http://www.error.be>.