The Definitive Guide to Spring Web Flow

Erwin Vervaet

Apress®

The Definitive Guide to Spring Web Flow

ISBN-13 (pbk): 978-1-4302-1624-7

ISBN-13 (electronic): 978-1-4302-1625-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Steve Anglin
Technical Reviewers: Keith Donald, Karl David Moore

Senior Project Manager: Kylie Johnston
Copy Editor: Heather Lang
Associate Production Director: Kari Brooks-Copony
Senior Production Editor: Laura Cheu
Compositor: Pat Christenson
Proofreader: Lisa Hamilton
Indexer: Becky Hornyak
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

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

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at http://www.apress.com.

Contents at a Glance

Contents

About the Author

ERWIN VERVAET is an independent consultant based in Leuven, Belgium. Erwin has been using Java since its inception and has extensive experience applying it in a wide range of application domains. He also runs his own software and consultancy company, Ervacon (`http://www.ervacon.com`).

Erwin enjoys writing, teaching, and speaking about Java- and Spring-related subjects. As the originator of the Spring Web Flow project, he currently co-leads its development together with Keith Donald.

About the Technical Reviewers

KEITH DONALD is a principal and founding partner of SpringSource (formerly Interface21), the company behind Spring. He is best known in the Spring community for creating Spring Web Flow with Erwin Vervaet. At SpringSource, Keith is the lead of the web application development products team. His team, based in Melbourne, Florida, helps sustain the development of Spring Web MVC and Web Flow and their associated integrations and is also responsible for future innovations in the domain of web application development frameworks.

Keith, together with Jay Zimmerman of NoFluffJustStuff Software Symposiums, is also the director of the Spring Experience conference series. He is responsible for the technical content of the conference, which takes place in Florida each December.

In addition, Keith is the principal architect behind SpringSource's state-of-the-art Spring training curriculum. This curriculum has provided practical training on Spring to over 3,000 students worldwide.

Keith, an experienced enterprise software developer and mentor, has built business applications for customers spanning a diverse set of industries including banking, network management, information assurance, education, and retail. He is particularly adept at translating business requirements into technical solutions.

Keith's blog can be found at `http://blog.springsource.com/main/author/keithd`.

KARL DAVID MOORE is a software engineer with over six years' commercial development experience. He holds a first class honors degree in software engineering from Sheffield Hallam University.

Karl has been a Spring Framework user since early 2004 and is an active contributor to the Spring forums, with over 8,000 posts. During his career, Karl has been extensively involved in all aspects of software development and has a particular passion for refactoring legacy systems. This has helped him foster a very strong interest in test-driven development, code quality, and attention to detail.

Karl enjoys researching and developing approaches to aid and simplify development, and evangelizing about techniques and tools to improve the skills of other developers.

You can find Karl's LinkedIn profile at `http://www.linkedin.com/in/karldmoore`.

Acknowledgments

This book, and Spring Web Flow itself, only exist thanks to the input and help of many different people.

Spring Web Flow was lucky and gained an active community right from the start. A special thanks goes out to the early adopters and forum members, whose invaluable feedback has helped mold Spring Web Flow into what it is today. Several people deserve special credits for their active involvement in the project: Juergen Hoeller, Colin Sampaleanu, Rod Johnson, Ben Hale, Christian Dupuis, J. Enrique Ruiz, C_sar Ordiñana, Rossen Stoyanchev, Jeremy Grelle, Rob Harrop, Seth Ladd, Colin Yates, Steven Devijver, Greame Rocher, Sam Brannen, Maxim Petrashev, Marten Deinum, and Dave Syer. It goes without saying that countless others also deserve credit.

Writing a book is a big undertaking that could not be done without the support and feedback of several people. I would like to thank Keith Donald for his invaluable input into Spring Web Flow, as well as for his review efforts. Karl David Moore also went above and beyond the call of duty, doing extraordinarily thorough chapter reviews and helping me polish my quirky English. The fine people at Apress that molded this book into its current form also deserve accolades: Heather Lang, Laura Cheu, and Kylie Johnston. Several other people also directly or indirectly contributed to this book: Philip Van Bogaert, Henri Shih, Kris Meukens, and Mike Seghers. Thank you all.

A very special thanks goes out to my wife Bieke. She tirelessly kept the kids busy and did more than her fair share while I slaved away at this book. I can honestly say that without Bieke's support, this book would not have been possible.

Introduction

When I started working on Spring Web Flow at the end of 2004, web applications already accounted for a large part of the Java enterprise development space. I had used Struts on several projects at that point but always felt something was missing. Working with a proprietary framework on a few projects in the financial industry sparked my interest. The framework I was using included a work flow engine, a fairly typical feature for frameworks targeted at high-end enterprise applications. What was novel about it, however, was that the work flow engine could also be used to define *page flows* in web applications. This brought a refreshingly intuitive approach to Java web application development.

Using a state diagram as the basis for page flows in web applications seemed much more natural than the request-centric solutions offered by the mainstream frameworks of the time. This was especially true for the more complex use cases that required the user to pass through a number of different steps in the completion of a business process. Over the course of 2004, I had been learning about the Spring Framework, which was gaining momentum at the time, and had been impressed by its design and implementation quality. I set out to add a page flow controller to the Spring Web MVC framework and created what would later become Spring Web Flow.

Initially, Spring Web Flow focused on using a state-diagram–based approach to make defining page navigation in web applications easy and intuitive. This gave web application developers a powerful way to express page navigation rules. Expressive page flow definitions also highlighted the need for better *navigational control*. The infamous Back button problem caused all sorts of difficulties in web applications that tried to control page navigation. Spring Web Flow clearly needed to address this issue.

Around that time, I read an article discussing the use of continuations to solve navigational problems in web applications. This struck me as fitting very well with Spring Web Flow's flow execution model and provided the missing link. Spring Web Flow now combined two very attractive and complementary features:

- An intuitive and easy to use method of *expressing page navigation rules* in web applications.

- A powerful and robust *navigational control* system.

Bringing a third attractive feature to the table required only a small step:

- Encapsulate page flows as black-box *application modules* with a well-defined input-output contract.

Spring Web Flow has come a long way in the last two years. It has grown from a simple "flow controller" for the Spring Web MVC framework into "a next generation Java web application controller framework that allows developers to model user actions as high-level modules called flows. The framework delivers improved productivity and testability while providing a strong solution to enforcing navigation rules and managing application state" (Johnson et al 2003). The next generation of the framework, Spring Web Flow 2, introduces exciting new features to help you build and run rich web applications. Clearly, Spring Web Flow is now a mature project that is used in many production deployments and has an active user community.

I wrote this book not only to teach you how to work with Spring Web Flow but also to help you understand the rationale and motivation behind the framework. I hope you enjoy reading this book and have fun working with Spring Web Flow!

About the Spring Web Flow Project

The original incarnation of the Spring Web Flow project was a small open source project started by Erwin Vervaet in October 2004 called Ervacon Spring Web Flow (`http://www.ervacon.com/products/springwebflow`). The project caught the attention of Keith Donald, one of the developers on the Spring Framework team, and became an official Spring Framework subproject in February of 2005. After almost two years of active development and a number of preview releases and release candidates, the project released its first production ready 1.0 version in October of 2006. This version is the subject of this book.

Building on the solid foundation set by Spring Web Flow 1, development continued on the next generation of the product. Spring Web Flow 2, released in June of 2008, underwent a few architectural changes allowing it to more seamlessly integrate into a rich web environment. It has impressive support for JSF and AJAX techniques and further simplifies the flow definition syntax.

Spring Web Flow uses the well-known Apache 2 license, a free/open source software license (Apache Software Foundation 2004). The Apache 2 license is used by many other open source projects, such as the Spring Framework itself and the Apache HTTP Server. It allows use of Spring Web Flow for any purpose, whether commercial or noncommercial, and even allows for modification and redistribution.

Spring Web Flow is sometimes mistakenly written as "Spring WebFlow." This confusion has its origin in the Spring Web Flow package name, `org.springframework.webflow`, which writes `webflow` in one word. "Spring Web Flow" is also often abbreviated as "SWF."

The official Spring Web Flow home page is located at `http://www.springframework.org/webflow`. It is an essential resource for Spring Web Flow users wanting to keep an eye on the evolution of the project. If you're still left with questions after reading this book, you'll have a good chance of getting an answer on the very active Spring user forums: `http://forum.springframework.org`. The Ervacon Spring Web Flow Portal (`http://www.ervacon.com/products/swf`) also offers useful information such as Spring Web Flow tips and tricks and a practical introduction.

About This Book

This book aims to teach you how to work with Spring Web Flow. It covers both basic and advanced use cases and provides an in-depth reference to all features Spring Web Flow currently offers. You'll also learn to extend the framework to take it beyond its out-of-the-box feature set. Once you've finished this book, you'll be able to call yourself a Spring Web Flow expert!

Spring Web Flow 1 and Spring Web Flow 2

Before we continue, one important point needs to be clarified. This book deals with Spring Web Flow 1. The next generation of the framework, Spring Web Flow 2, is subject matter for another book.

Versions 1 and 2 are essentially two separate products. The core concepts are the same, but the two versions are quite different technically. As a result, Spring Web Flow 2 is not backward compatible with Spring Web Flow 1. Moving from version 1 to version 2 would be a migration rather than a simple upgrade. This book's Epilogue discusses the differences between the two versions in more detail and will help you decide which version is best for you.

Spring Web Flow 1 will be referred to as just "Spring Web Flow" in this book, omitting the version number. When talking about Spring Web Flow 2, the version number will be explicitly mentioned.

Target Audience

This book is intended to be a reference for both new and advanced Spring Web Flow users. If you're a new user, you'll learn how to get started using Spring Web Flow and leverage all of its powerful features. As an advanced user, you'll learn about extending the framework and many of its best practices, and you'll find this book provides very interesting insights into the design of Spring Web Flow.

Before reading this book, you should have a solid understanding of Java and Java web application development including topics such as servlets and JavaServer Pages (JSP). Many of the samples in this book use the Spring Web MVC framework. However, if you're familiar with any web Model-View-Controller (MVC) framework (for instance Struts or WebWork), you should have no problem following along.

A basic knowledge of the Spring Framework and its guiding principles, such as the Inversion of Control pattern and dependency injection, is also assumed. You don't need to be a Spring expert to read this book, but you'll have an easier time if you have at least played around with Spring classes like `ApplicationContext` and `BeanFactory` and understand how Spring wires together beans.

Rather than bloating this book with a detailed description of Java web applications, Spring Web MVC, or even Spring in general, I refer you to the Apress books *Beginning Spring 2: From Novice to Professional* by Dave Minter (2005); *Pro Spring 2.5* by Jan Machacek, Jessica Ditt, Aleksa Vukotic, and Anirvan Chakraborty (2008); and *Pro Java™ EE Spring Patterns: Best Practices and Design Strategies Implementing Java EE Patterns with the Spring Framework* by Dhrubojyoti Kayal (2008). Find other Apress books at `http://www.apress.com`.

Overview

This book provides both introductory material and in-depth coverage of Spring Web Flow. The following overview will help you focus on the chapters most relevant to you. You can also read this book from cover to cover, and I recommend doing so if you're new to Spring Web Flow. If you're already familiar with the framework, you can skip the first two chapters and head directly to Chapter 3.

Chapter 1: Introducing Spring Web Flow

This chapter takes a high-level view and examines the problem Spring Web Flow was designed to solve. It will explain the context in which Spring Web Flow lives and give you a conceptual understanding of what exactly Spring Web Flow is.

Chapter 2: Getting Started

After the broad introduction of Chapter 1, this chapter will help you to hit the ground running. It explains all the practical details to get started using and experimenting with Spring Web Flow. The environment setup in this chapter should enable you to easily follow along with the examples covered in later chapters and to *try things for yourself*.

Chapter 3: Spring Web Flow's Architecture

Chapters 1 and 2 superficially touch on some of the Spring Web Flow concepts. Chapter 3 digs a little deeper. It explains the Spring Web Flow architecture, giving you a detailed understanding of the different subsystems involved and setting the stage for an in-depth study of the Spring Web Flow feature set in the following chapters.

Chapter 4: Spring Web Flow Basics

This chapter covers basic Spring Web Flow features, needed in most, if not all, use cases. You'll learn how to design and implement a web flow using both the XML- and Java-based flow definition languages.

Chapter 5: Advanced Web Flow Concepts

As a follow up to the basic concepts covered in Chapter 4, this chapter will detail more advanced functionality. It explains how to reuse flows as subflows from inside other flows, realizing Spring Web Flow's promise of modularity. Handling HTML form data is also covered.

Chapter 6: Flow Execution Management

When working with Spring Web Flow, most of the development effort revolves around defining flows. Also very important, however, is understanding how Spring Web Flow manages flow executions and the associated data; these topics will be discussed in this chapter.

Chapter 7: Driving Flow Executions

This chapter focuses on integrating Spring Web Flow into hosting frameworks like Spring Web MVC and JSF and driving flow executions from those environments. Developing views for a web flow will also be discussed.

Chapter 8: Testing with Spring Web Flow

Unit testing Spring Web Flow applications is explained in this chapter. You'll learn how to perform integration tests with your flow definitions and how to test flow artifacts (such as actions) in isolation.

Chapter 9: The Sample Application

To give you an example of a nontrivial application that combines both free browsing and controlled navigation situations, this chapter will document a sample application: Spring Bank. Spring Bank is a simple electronic banking application that allows users to do things such as manage their bank accounts or enter payments. Use cases of this application will be used throughout this book to help explain and illustrate Spring Web Flow's feature set.

Chapter 10: Real-World Use Cases

This chapter covers some frequently asked questions related to use cases occurring in the real world. You'll learn about such things as securing your flows, tracking breadcrumbs, or stress testing Spring Web Flow applications.

Chapter 11: Extending Spring Web Flow

The last chapter covers extending and customizing Spring Web Flow. You'll also learn how to build Spring Web Flow from the sources.

Epilogue

To conclude this book, the epilogue leaves you with some parting thoughts and takes a look at what's new and improved in Spring Web Flow 2.

Typographical Conventions

This book uses simple and easy-to-understand typographical conventions. All text to be interpreted in a literal sense, like Java class names, code fragments, file names, or XML elements uses a `fixed width font`. *Italics* indicate topics of particular importance, and new or important pieces of a code fragment are highlighted in **`boldface fixed-width text`**. Commands to be entered on the command line also use a **boldface font**.

Many of the program listings presented in this book have been formatted for readability and to make them fit nicely on the page. In some cases, additional line breaks had to be introduced. A backslash (`\`) line continuation marker is used whenever an extra line break had to be added.

About the Examples

All the sample applications presented in this book use Java 5, Servlet API 2.4, and JSP 2.0. Make sure you have a Java 5 Development Kit (JDK) and an appropriate application server or Servlet engine installed on your computer (for instance, Tomcat 5 or Jetty 6).

Spring Web Flow 1.0.6, together with Spring 2.5.4, was used to develop the sample applications. Future Spring Web Flow 1.x and Spring 2.x versions will be compatible. Chapter 2 provides a detailed overview of setting up a build and development environment you can use to run the sample applications.

The Spring Bank sample application, together with the other samples discussed in this book, can be found in the public Ervacon Subversion repository at `https://svn.ervacon.com/public/spring`. You can browse the source code by just pointing your browser at this address. Alternatively, you can check out the entire source tree using any Subversion client, for instance, TortoiseSVN (`http://tortoisesvn.tigris.org`) if you are using Microsoft Windows. The source code for this book is also available to readers at `http://www.apress.com` in the Downloads section of this book's home page. Please feel free to visit the Apress web site and download all the code there.