Pro Spring 5

An In-Depth Guide to the Spring Framework and Its Tools

Fifth Edition

Iuliana Cosmina
Rob Harrop
Chris Schaefer
Clarence Ho

Apress®

Pro Spring 5

An In-Depth Guide to the Spring Framework and Its Tools

Fifth Edition

Iuliana Cosmina
Rob Harrop
Chris Schaefer
Clarence Ho

Apress®

Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools

Iuliana Cosmina
Sibiu, Sibiu, Romania

Rob Harrop
Reddish, United Kingdom

Chris Schaefer
Venice, Florida, USA

Clarence Ho
Hong Kong, China

ISBN-13 (pbk): 978-1-4842-2807-4
ISBN-13 (electronic): 978-1-4842-2808-1
DOI 10.1007/978-1-4842-2808-1

Library of Congress Control Number: 2017955423

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.

Cover image by Freepik (www.freepik.com)

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Technical Reviewer: Massimo Nardone
Coordinating Editor: Mark Powers
Copy Editor: Kim Wimpsett

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, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles 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 Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484228074. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

I dedicate this book to my friends, to my godson ștefan, and to all the musicians who have made working on this book easy with their music.

—Iuliana Cosmina

Contents at a Glance

Contents

About the Authors

Iuliana Cosmina is a Spring-certified Web Application Developer and is also a Spring-certified Spring Professional, as defined by Pivotal, the makers of Spring Framework, Boot, and other tools. She has authored books with Apress on core Spring certification and Spring-certified web development. She is a software architect at Bearing Point Software and is an active coder and software contributor on GitHub, Stack Overflow, and more.

Rob Harrop is a software consultant specializing in delivering high-performance, highly scalable enterprise applications. He is an experienced architect with a particular flair for understanding and solving complex design issues. With a thorough knowledge of both Java and .NET, Harrop has successfully deployed projects across both platforms. He also has extensive experience across a variety of sectors, retail and government in particular. Harrop is the author of five books, including *the book you are currently reading, not at its fifth edition*, a widely acclaimed, comprehensive resource on the Spring Framework.

Chris Schaefer is a principle software developer for Spring projects at Pivotal, the makers of Spring Framework, Boot, and other Spring tools.

Clarence Ho is the senior Java architect of a Hong Kong–based software consultancy firm, SkywideSoft Technology Limited. Having been worked in the IT field for more than 20 years, Clarence has been the team leader of many in-house application development projects, as well as providing consultancy services on enterprise solutions to clients.

About the Technical Reviewer

Massimo Nardone has more than 23 years of experience in security, web/mobile development, cloud computing, and IT architecture. His true IT passions are security and Android.

He currently works as the chief information security officer (CISO) for Cargotec Oyj and is a member of the ISACA Finland Chapter board. Over his long career, he has held these positions: project manager, software engineer, research engineer, chief security architect, information security manager, PCI/SCADA auditor, and senior lead IT security/cloud/SCADA architect. In addition, he has been a visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University).

Massimo has a master of science degree in computing science from the University of Salerno in Italy, and he holds four international patents (PKI, SIP, SAML, and proxy areas). Besides working on this book, Massimo has reviewed more than 40 IT books for different publishing companies and is the coauthor of *Pro Android Games* (Apress, 2015).

Acknowledgments

It is a huge honor for me to be the main author of the fifth edition of this book. Would you believe I got this assignment by mistake? I thought I was getting an assignment as a technical reviewer for this book. Only when I received the files did I realize that I was going to be one of the authors of the fifth edition of one of the best Spring books on the market.

Apress has published many of the books I have read and used to improve myself professionally during my studies and even after that. This is my third book with Apress, and it is great to contribute to the education of the next generation of developers.

I am grateful to all my friends who had the patience to listen to me complain about losing sleep, having too much work to do, and encountering writer's block. Thank you all for being supportive and making sure I still had some fun while writing this book.

Also, I would like to give a big thanks to all my favorite singers who made my work easier with their wonderful music, especially John Mayer; I was so determined to finish this book on time just so I could go to the United States to one of his concerts. That is why I changed the topic of the examples in this book to be about singers and their music; it's a tribute to their art and talent.

—Iuliana Cosmina

Introduction

Covering version 5 of the Spring Framework, this book is the most comprehensive Spring reference and practical guide available for harnessing the power of this leading enterprise Java application development framework.

This edition covers core Spring and its integration with other leading Java technologies, such as Hibernate, JPA 2, Tiles, Thymeleaf, and WebSocket. The focus of the book is on using Java configuration classes, lambda expressions, Spring Boot, and reactive programming. We share our insights and real-world experiences with enterprise application development, including remoting, transactions, the web and presentation tiers, and much more.

With Pro Spring 5, you'll learn how to do the following:

- Use inversion of control (IoC) and dependency injection (DI)
- Discover what's new in Spring Framework 5
- Build Spring-based web applications using Spring MVC and WebSocket
- Build Spring web reactive applications with Spring WebFlux
- Test Spring applications using Junit 5
- Utilize the new Java 8 lambda syntax
- Use Spring Boot to an advanced level to get any kind of Spring application up and running in no time
- Use Java 9 features in Spring applications

Because the Java 9 release date kept being postponed, Spring 5 was released based on Java 8. Thus, interoperability with Java 9 is covered in this book based on an early-access build.

There is a multimodule project associated with this book, configured using Gradle 4. The project is available on the Apress official repository: `https://github.com/Apress/pro-spring-5`. The project can be built right after cloning according to the instructions in its `README.adoc` file as long as Gradle is installed locally. If you do not have Gradle installed, you can rely on InteliJ IDEA to download it and use it to build your project by using the Gradle Wrapper. (`https://docs.gradle.org/current/userguide/gradle_wrapper.html`). There is a small appendix at the end of the book describing the project structure, configuration and additional details related to development tools that can be used to develop and run the code samples of the book, which are available on GitHub.

As the book was being written, new release candidate versions of Spring 5 were released, a new version of Intellij IDEA was released, and new versions of Gradle and other technologies used in the book were updated. We upgraded to the new versions to provide the most recent information and keep this book synchronized with the official documentation. Several reviewers have checked the book for technical accuracy, but if you notice any inconsistencies, please send an email to `editorial@apress.com` and errata will be created.

You can access the example source code for this book via the Download Source Code button at `www.apress.com/9781484228074`. It will be maintained, synchronized with new versions of the technologies, and enriched based on the recommendations of the developers using it to learn Spring.

We truly hope you will enjoy using this book to learn Spring as much as we enjoyed writing it.

CHAPTER 1

Introducing Spring

When we think of the community of Java developers, we are reminded of the hordes of gold rush prospectors of the late 1840s, frantically panning the rivers of North America, looking for fragments of gold. As Java developers, our rivers run rife with open source projects, but, like the prospectors, finding a useful project can be time-consuming and arduous.

A common gripe with many open source Java projects is that they are conceived merely out of the need to fill the gap in the implementation of the latest buzzword-heavy technology or pattern. Having said that, many high-quality, usable projects meet and address a real need for real applications, and in the course of this book, you will meet a subset of these projects. You will get to know one in particular rather well—Spring. The first version of Spring was released in October 2002 and consisted of a small core with an inversion of control (IoC) container that was easy to configure and use. Over the years Spring has become the main replacement of Java Enterprise Edition (JEE) servers and has grown into a full-blown technology made up of many distinct projects, each with its own purpose, so whether you want to build microservices, applications, or classical ERPs, Spring has a project for that.

Throughout this book, you will see many applications of different open source technologies, all of which are unified under the Spring Framework. When working with Spring, an application developer can use a large variety of open source tools, without needing to write reams of code and without coupling an application too closely to any particular tool.

In this chapter, as its title indicates, we introduce you to the Spring Framework, rather than presenting any solid examples or explanations. If you are already familiar with Spring, you might want to skip this chapter and proceed straight to Chapter 2.

What Is Spring?

Perhaps one the hardest parts of explaining Spring is classifying exactly what it is. Typically, Spring is described as a lightweight framework for building Java applications, but that statement brings up two interesting points.

First, you can use Spring to build any application in Java (for example, stand-alone, web, or JEE applications), unlike many other frameworks (such as Apache Struts, which is limited to web applications).

Second, the *lightweight* part of the description doesn't really refer to the number of classes or the size of the distribution but rather defines the principle of the Spring philosophy as a whole—that is, minimal impact. Spring is lightweight in the sense that you have to make few, if any, changes to your application code to gain the benefits of Spring Core, and should you choose to stop using Spring at any point, you will find doing so quite simple.

Notice that we qualified that last statement to refer to Spring Core only—many of the extra Spring components, such as data access, require a much closer coupling to the Spring Framework. However, the benefits of this coupling are quite clear, and throughout the book we present techniques for minimizing the impact this has on your application.

I. Cosmina et al., *Pro Spring 5*, https://doi.org/10.1007/978-1-4842-2808-1_1

Evolution of the Spring Framework

The Spring Framework originated from the book *Expert One-on-One: J2EE Design and Development* by Rod Johnson (Wrox, 2002). Over the last decade, the Spring Framework has grown dramatically in core functionality, associated projects, and community support. With the new major release of the Spring Framework, it's worthwhile to take a quick look back at important features that have come along with each milestone release of Spring, leading up to Spring Framework 5.0.

- *Spring 0.9*: This is the first public release of the framework, based on the book *Expert One-on-One: J2EE Design and Development*, that offered a bean configuration foundation, AOP support, a JDBC abstraction framework, abstract transaction support, and so on. This version does not have official reference documentation, but you can find the existing sources and documentation on SourceForge.[1]
- *Spring 1.x*: This is the first version released with official reference documentation. It is composed of the seven modules shown in Figure 1-1.

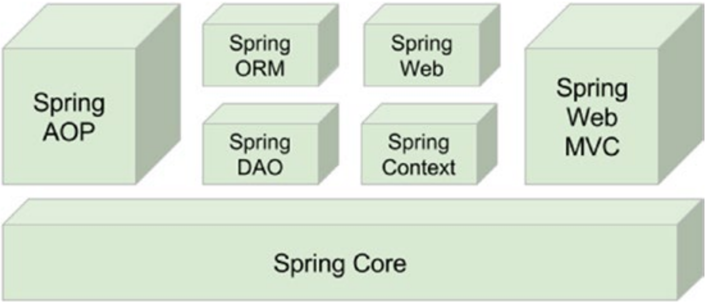

Figure 1-1. *Overview of the Spring Framework, version 1.x*

  - *Spring Core*: Bean container and supporting utilities
  - *Spring Context*: `ApplicationContext`, UI, validation, JNDI, Enterprise JavaBeans (EJB), remoting, and mail support
  - *Spring DAO*: Transaction infrastructure, Java Database Connectivity (JDBC), and data access object (DAO) support
  - *Spring ORM*: Hibernate, iBATIS, and Java Data Objects (JDO) support
  - *Spring AOP*: An AOP Alliance–compliant aspect-oriented programming (AOP) implementation
  - *Spring Web*: Basic integration features such as multipart functionality, context initialization through servlet listeners, and a web-oriented application context
  - *Spring Web MVC*: Web-based Model-View-Controller (MVC) framework

- *Spring 2.x*: This is composed of the six modules shown in Figure 1-2. The Spring Context module is now included in Spring Core, and all Spring web components have been represented here by a single item.

[1] You can download older versions of Spring including 0.9 from the SourceForge site: `https://sourceforge.net/projects/springframework/files/springframework/`.

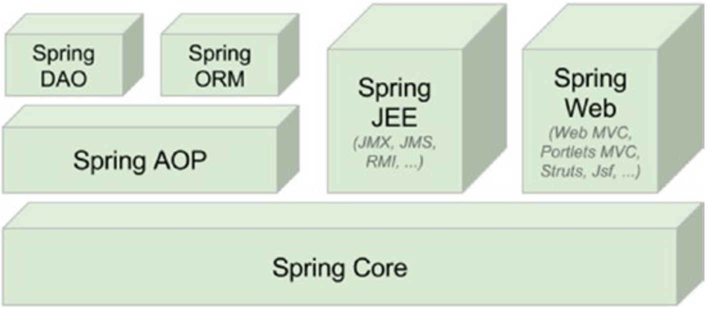

Figure 1-2. *Overview of the Spring Framework, version 2.x*

  - Easier XML configuration through the use of the new XML Schema–based configuration rather than the DTD format. Notable areas of improvement include bean definitions, AOP, and declarative transactions.
  - New bean scopes for web and portal usage (request, session, and global sessions).
  - `@AspectJ` annotation support for AOP development.
  - Java Persistence API (JPA) abstraction layer.
  - Full support for asynchronous JMS message-driven POJOs (for plain old Java objects).
  - JDBC simplifications including `SimpleJdbcTemplate` when using Java 5+.
  - JDBC named parameter support (`NamedParameterJdbcTemplate`).
  - Form tag library for Spring MVC.
  - Introduction of the Portlet MVC framework.
  - Dynamic language support. Beans can be written in JRuby, Groovy, and BeanShell.
  - Notification support and controllable MBean registration in JMX.
  - `TaskExecutor` abstraction introduced for scheduling tasks.
  - Java 5 annotation support, specifically for `@Transactional`, `@Required`, and `@AspectJ`.

- *Spring 2.5.x*: This version has the following features:
  - A new configuration annotation called `@Autowired` and support for JSR-250 annotations (`@Resource`, `@PostConstruct`, `@PreDestroy`)
  - New stereotype annotations: `@Component`, `@Repository`, `@Service`, `@Controller`
  - Automatic classpath-scanning support to detect and wire classes annotated with stereotype annotations
  - AOP updates, including a new bean pointcut element and AspectJ load-time weaving
  - Full WebSphere transaction management support

- In addition to the Spring MVC `@Controller` annotation, `@RequestMapping`, `@RequestParam`, and `@ModelAttribute` annotations added to support request handling through annotation configuration
- Tiles 2 support
- JSF 1.2 support
- JAX-WS 2.0/2.1 support
- Introduction of the Spring TestContext Framework, providing annotation-driven and integration testing support, agnostic of the testing framework being used
- Ability to deploy a Spring application context as a JCA adapter

- *Spring 3.0.x*: This is the first version of Spring based on Java 5 and is designed to take full advantage of Java 5 features such as generics, varargs, and other language improvements. This version introduces the Java-based `@Configuration` model. The framework modules have been revised to be managed separately with one source tree per module JAR. This is abstractly depicted in Figure 1-3.

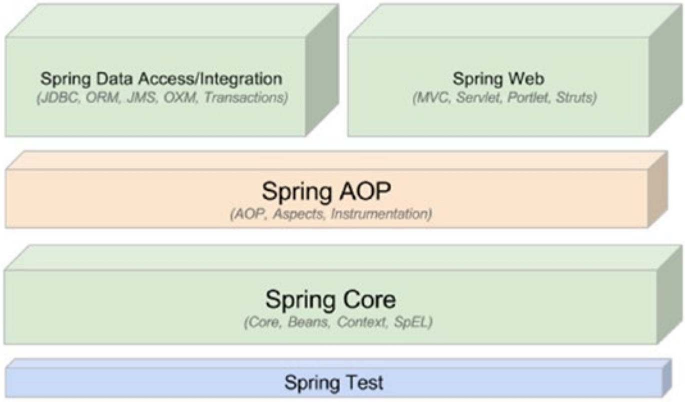
Spring Data Access/Integration
(JDBC, ORM, JMS, OXM, Transactions)
Spring Web
(MVC, Servlet, Portlet, Struts)
Spring AOP
(AOP, Aspects, Instrumentation)
Spring Core
(Core, Beans, Context, SpEL)
Spring Test

Figure 1-3. *Overview of the Spring Framework, version 3.0.x*

- Support for Java 5 features such as generics, varargs, and other improvements
- First-class support for Callables, Futures, `ExecutorService` adapters, and `ThreadFactory` integration
- Framework modules now managed separately with one source tree per module JAR
- Introduction of the Spring Expression Language (SpEL)
- Integration of core Java Config features and annotations
- General-purpose type conversion system and field-formatting system
- Comprehensive REST support
- New MVC XML namespace and additional annotations such as `@CookieValue` and `@RequestHeaders` for Spring MVC

  - Validation enhancements and JSR-303 (Bean Validation) support
  - Early support for Java EE 6, including `@Async/@Asynchronous` annotation, JSR-303, JSF 2.0, JPA 2.0, and so on
  - Support for embedded databases such as HSQL, H2, and Derby

- *Spring 3.1.x*: This version has the following features:
  - New cache abstraction
  - Bean definition profiles can be defined in XML as well as support for the `@Profile` annotation
  - Environment abstraction for unified property management
  - Annotation equivalents for common Spring XML namespace elements such as `@ComponentScan`, `@EnableTransactionManagement`, `@EnableCaching`, `@EnableWebMvc`, `@EnableScheduling`, `@EnableAsync`, `@EnableAspectJAutoProxy`, `@EnableLoadTimeWeaving`, and `@EnableSpringConfigured`
  - Support for Hibernate 4
  - Spring TestContext Framework support for `@Configuration` classes and bean definition profiles
  - c: namespace for simplified constructor injection
  - Support for Servlet 3 code-based configuration of the Servlet container
  - Ability to bootstrap the JPA `EntityManagerFactory` without `persistence.xml`
  - `Flash` and `RedirectAttributes` added to Spring MVC, allowing attributes to survive a redirect by using the HTTP session
  - URI template variable enhancements
  - Ability to annotate Spring MVC `@RequestBody` controller method arguments with `@Valid`
  - Ability to annotate Spring MVC controller method arguments with the `@RequestPart` annotation
- *Spring 3.2.x*: This version has the following features:
  - Support for Servlet 3–based asynchronous request processing.
  - New Spring MVC test framework.
  - New Spring MVC annotations `@ControllerAdvice` and `@MatrixVariable`.
  - Support for generic types in `RestTemplate` and in `@RequestBody` arguments.
  - Jackson JSON 2 support.
  - Support for Tiles 3.
  - `@RequestBody` or an `@RequestPart` argument can now be followed by an `Errors` argument, making it possible to handle validation errors.
  - Ability to exclude URL patterns by using the MVC namespace and Java Config configuration options.
  - Support for `@DateTimeFormat` without Joda Time.