pivotspring

liubo404

Published
with GitBook

Table of Contents

en readme

Chapter 1

Introduction

So here you are: you want to learn how to develop web applications using Spring,and you chose this study guide to help you. This might be one of the best decisons that you could make, as this book was written not noly to help you to understatnd Spring Web, but to love it as well. This study guide may even help you pass the certification exam--if you follow al the instructions properly and do all the exercises. This study guide explores more Spring Web is all about, which you can skip reading, of course; but if you really want to learn Spring to develop web applications like a professional, then it would be wise not to skip this.

Spring and What It Does

When building a project using Java, a lot of functionality needs to be built from scratch. But a lot of useful functionalities are already built and are free to use because of the open source world we live in. A long time ago, when the Java world was still quite small, you would say that you were using a library when you used open source code developed by somebody else, shipped as a *.jar file. But as time passed, the software development world evolved and the libraries grew too. They became frameworks. Because they were no longer one* .jar file that you could import, they became a collection of more-or-less decoupled libraries with different responsibilities, and you had the option to import only what you needed.

Release in October 2002 as an open source framework and an inversion of control container developed using Java, Spring was built for the Java platform. It was concieved with the dependency injection software design pattern in mind, and its main purpose is to make dependency handling easier. A Java application is basically a group of objects exchanging data and influencing each other's behavior. The Spring Framework simplified the way in which objects talks to each other and the way they depend on each other. This is why Spring evangelists claim that the reason Java was invented was so that Spring would come into existence one day. The development of Java applications became easier when Spring emerged, providing comprehensive infrastructure support. Spring makes it easier to compose disparate components into a fully working application.

Spring comes with a lot of default behaviors already implemented(components called infrastructure beans are a default configuration; then can be used to create functional basic applications without extra customization), because the Spring Framework was also built with

the convention over configuration paradigm as principle, which seeks to decrease the number of decisions a developer has to make when writing code, but also makes it easier for the developer to customize the behavior of objects, offering increased flexibility.

Spring is currently the VIP of Java frameworks and it has been growing exponentially, especially since 2009, when VMware acquired SpringSource, the company behind Spring. The VMware and the EMC Corporation in April 2013, now known as Pivotal, was also advantageous for Spring, as it became one of Pivotal's central elements in its strategy to provide innovative and modern software-driven experiences to its customers. Spring is now a full-blown technoloy that can be used to build enterprise-ready applications in a very short time, and it comes in 25 flavors. Figure 1-1 shows a diagram of all Spring-released projects. The following list describes these projects.

- **Spring Framework** provides core support for dependency injection, transaction management, web applications, data access, messaging, and more.
- **Spring Boot** provides compact setups for different types of applications, helping you to focus on your code instead of infrastructure configuration.
- **Spring XD** simplifies the development of Big Data applicatios.

> [n]. You can read about these projects, as well as other pojects that have not released officially(Spring Session, for example) in detail at http://spring.io/projects. ↩

- **Spring Cloud** provides a set of tools for distributed applications.
- **Spring Data** provides a consistent approach to data access. (This study guide uses a subproject called Spring Data JPA to help us manage data easily.)
- **Spring Integration** supoorts the well-known Enterprise Integration Patterns via lightweight messaging and declarative adapters.
- **Spring Batch** simplifies and optimizes the work of processing high-volume batch operations.
- **Spring Security** provides tools for authentication and authorization. (Because *web security is one of the subjects of the certification exam*, there is a section about web security in this study guide that you will have to pay close attention to.)
- **Spring HATEOAS** provides some APIs to help the development of REST representations that follow the HATEOAS principle(Hypermedia as the Engine of Application State,which means that a client interacts with a network application entirely through hypermedia provided dynamically by application servers).
- **Spring Social** provides an API to connect Spring applications to the third-party APIs of social networks like Facebook and Twitter, as well as others.
- **Spring AMQP** provides an API for AMQP-based messaging solutions.
- **Spring Mobile** simplifies the development of mobile applications.
- **Spring for Android** provides key spring components to use in the development of Android applications.

- **Spring Web Flow** supports the building of web application with controlled navigation(Spring Web Flow is another subject in the certification exam.)
- **Spring Web Service** facilitates the development of SOAP-based applications.
- **Spring LDAP** provides tools to develop LDAP applications.
- **Grails** is a powerful open source web framework based on Groovy and inspired by Ruby on Rails. It is used to create web applications that run on the Java Virtual Machine(JVM).
- **Groovy** started as a dynamic language for the Java platform. It brings high-productiviy development features to the JVM, and resembles Python, Ruby, Perl, and Smalltalk in regards to syntax and features. SpringSource has taken over its development and maintenance.
- **Spring Scala** mixed up Spring with Scala language features.
- **Spring Roo** helps define application templates that can be built into a full Java application within minutes.
- **Spring BlazeDS Integration** tools integrate Spring with Adobe BlazeDS.
- **Spring Loaded** reloads classes as files changes, boosting productivily (similar project to JRebel).

> n. Pivotal decided to stop funding this project in March 2015. ↩

> n. Funding for this project also enned din March 2015. ↩

- **Spring Shell** provides the capability to build command-line apps.
- **REST Shell** makes the writing and testing of RESTful application easier with CLI-based resource discovery and interaction.

The Focus of this Study Guide

As this study guide is being written, the Spring Framework consists of features organized into about 20 modules grouped into following: Core Container, Data Access/Integration, Web, AOP(aspect-oriented programming), Instrumentation, Messaging, and Test. The topics covered in this study guide are Spring Framework's support components for the presentation tier(and specifically web-based presentatation tires). A bonus in this book is the Spring WebSocket chapter, which was added to the Spring Framework in version 4 and is also an optional part of the official Spring Web course not featured in the certification exam. In the Figure 1-2 you can see the SprngMVC stack, a tiered representation of the modules commonly used to create Spring web applications.

Figure 1-2. The Spring Web Stack(those with dotted lines will not be covered in this study guide)

This study guide focuses on helping developers understand how Spring's web infrastructure is designed, and how to write Spring web applications in a few easy steps by maximizing Spring's potential.The study guide's objective are as follows:

- Use Spring to develop web applications
- Use Spring Web Flow to implement stateful interactions
- Use Spring Security to secure web applicatons
- Use Spring Test and other test frameworks(JUnit, JMock) to test web applications
- Create Spring web applications using Gradle[4]

[4]Gradle is an automated build tool that is easy to cnfigure and use on any type of application. Its build files are written using JSON and Groovy. Gradle combines the power and flexibility of Ant with dependency management and conventions of Maven into a more effective way to build. Read more about it at https://www.gradle.org.

Who Should Use this Study Guide

This study guide is designed to help any Spring developer become familiar and comfortable with Spring-associated technologies for web development. It can also be a big help to a developer who wants to become a **Certified Spring Web Application Developer**[5]. That is why every topic in the official VMware Spring Web study guide is given the attention that it deserves. You do not have to be a Certified Spring Professional to use this study guide; you just need minimal knowledge of Spring. Because this study guide has a full chapter dedicated to the Spring component, it might be possible for a non-Spring developer to use this study guide too, but the *Spring Framework Reference* official documentation should be consulted to retrieve any missing pieces of information. In a nutshell, this study guide was written to be used by the following audiences:

- Spring Core developers who want a taste of Spring Web
- Spring developers(Certified Spring Professionals or not) who are looking forward to becoming Certified Spring Web Application Developers
- Java developers who are curious about working with Spring technologies and want to start fast.

About the Spring Web Certification Exam

If you are interested in becoming a **Certified Spring Web Application Developer**, the first step is to go to the VMware official learning site(http://pivotal.io/training) and search for the Spring Certification section. There you will find all the details you need regarding the official trainings, including where and when they take place. The training is four days long. There is

online training available as well. After creating an account on the VMware site, you can select your desired training. After you make the payment, if you choose an online training, after about a month you will receive (through the mail) an official training kit that consists of the following:

- A paire of conference headphones(usually Logitch) to be used during training to hear your trainer talk and to ask questions.
- A professional webcam(usually Logitech) to be used during training so that your trainer and colleagues can see you, thus simulating a classroom experience.[8]
- A Spring study guide book containing the printed version of the slides your tutor will use during training.
- A Spring study lab book containing explanations and instructions for the practical exercises you will do during taining.
- A SpringSource official flash drive containing the following:
  - A JDK installer.
  - Sources necessary for your tarining. Each study lab has s small Spring web application with missing configuration and code; the student's task is to complete it to become a working application. The same model is used in the code associated with this book.
  - An install for the most recent stable version of the Spring Tool Suite(STS). The version on the flash drive is mandatory for the course because the install sets up a local Maven repository with all the needed dependencies, and a full eclipse project configuration with the lab sources. The STS also has an internal tc Server to run the lab application.
  - An HTML or PDF version of the Spring Study Lab.

[5]Keep in mind that attending a Spring Web taining course by Pivotal or at a VMware Authorized Training Certer is prerequisite to becoming a Certified Spring Web Application Developer, as stated on the official site at http://mylearn.vmware.com/mgrReg/plan.cfm?plan=31111ui=www_cert.

[6]The Spring Framework Reference is at http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/.

[7]Depending on the area and the training certer, this item is optional.

[8]Depending on the area and the training certer, this item is also optional.

If you do not choose to do online training, you will not receive the headphones nor the webcam. The training kit and the rest of the materials are given to you when you arrive at the location where the training is taking place. After your training, you receive a free voucher that is required to schedule the certification exam at an approved exam center near you.

Basically, this voucher or voucher code is proof that you have attended official Spring Web training. !The exam duration is **90 minutes** and consists of **50 questions**. There are both single-anser and multiple-choice questions. The questions cover(roughly) the following topics:

- Spring overview(Spring core notions)
- MVC essentials(contifurations, beans to use, converions)
- MVC forms and views
- Webflow
- Web security
- REST

The passing score for the exam is **76%**. This means that **38** correct answers are needed to pass. Most of the questions present you with a piece of Java code or configuration and then ask you what it does, so make sure that you understand the code attached to this book and

Chapter 2 Spring Fundamentals

This chapter is necessary for building a Spring backgroud, which will be very helpful in the upcoming chapters. This chapter vill help you get familiar with the Spring container, context,beans,and most Spring core modules and how they work togather to allow developers to focus on solving problems instead of building up support.

The Basics

Any application system is make of components that work together to solve a problem. In object-oriented design they are called *classes*. Figure2-1 depicts the sequence of operations necessary to create a Person instance. Bacause this chapter is about Spring Core, a web application is not needed, so request to manipulate Person instances will be directed to implementations of the PersonManager interface. Implementations of this interface will provide access to the database using an implementation of PersonRepository interface. The operatin is pretty simple and the setup to write and execute the code should be too. This is where Spring comes in--providing a way to build an application using **p**lan **o**ld **J**ava **o**bject(POJOs)[1] and applying enterprise services(transaction execution, remote execution) noninvasively.

[1]A software term introduced by Martin Fowler, Rebeca Parsons, and Josh MacKenzie in September 2000 to refer to ordinary Java objects not bound by any restriction.

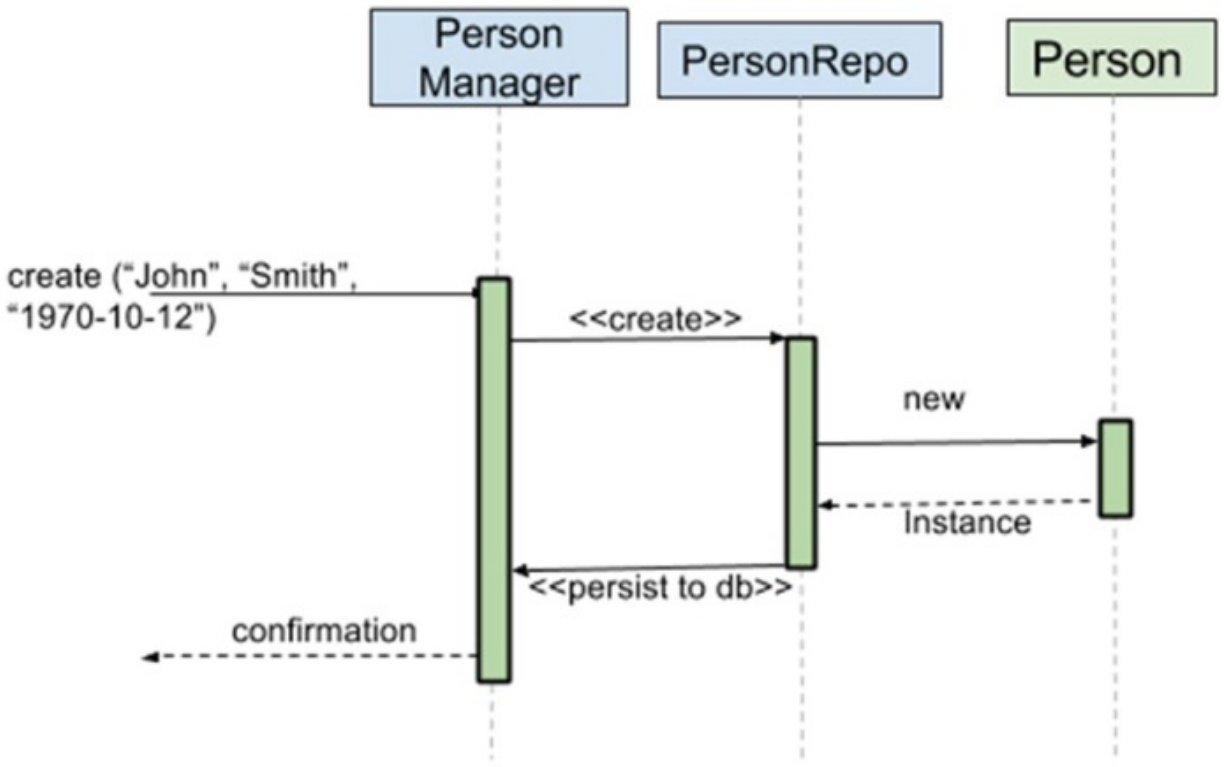

***Figure2-1**. UML sequence of operations necessary to create a Person instance*

The components making up an application interact and depend on one another. Defining how those objects are composed is quite difficult using plain Java. Even with the help of all the design patterns defined by experts in the software industry, the work is still cumbersome,

as the pattern components still have to be implemented before used. The Spring *invertion of control*(IoC) container was degined to help developers compose objects into fully working applications, ready ro use[2].