Core Spring 3.x

Certification Study Guide

Completed: July, 2010

Table of Contents

Overview....3
Topics By Subject Area....4
Container -basics....4
General....4
Lifecycle....4
Annotations....4
Miscellaneous....4
JavaConfig....5
Testing....5
AOP....5
Recommendations....5
Pointcuts....5
Advices....5
Configuration....5
Proxies....6
Data Access and transactions....6
General....6
The JdbcTemplate....6
Hibernate....6
Transactions....6
Spring MVC and REST....7
General configuration....7
Controllers....7
REST....7
Advanced topics....7
Remoting....7
Security....7
JMS....8
JMX....8
Resources....9

Overview

This guide is designed to help you prepare for the Core Spring 3.x certification exam. Please beware it should not be used if you have attended a Core-Spring course that was using Spring 2.x (the certification exam is then called Core-Spring 2.5 and there is a dedicated certification guide that goes with it).

The certification exam is based on the SpringSource 4-day Core Spring training and the materials provided with it are the ideal source to use for preparation. Of course as with any certification the most valuable part, besides recognition, is the learning process. Hence we encourage you to take time to experiment and follow your curiosity when questions arise.

A 4-day course contains a lot of material. To help you focus your efforts and to know when you're ready we've put together this guide. The guide contains a list of topics and a list of further resources. Topics are organized by subject area, where each topic contains a description of what you should make sure you know.

The list of topics can be used as a check-list. The training materials can be used as a point of reference and as a learning ground. The list of resources is where you can go further for getting answers.

One possible way to prepare is to do the following for a given training module:

1. Review the slides, making notes of questions
2. Work through the lab (if there is one)
3. Review the list of topics that matches to the module by subject area
4. Use the lab to experiment with anything you need to spend more time on
5. Use the provided list of resources to look for further answers

Of course there are many more ways to organize your efforts. You can pair up with someone else planning to take the exam or review all presentations for a given subject area before going through the labs. Or maybe you have access to actual applications you can review to test your knowledge.

Please keep in mind that you are expected to have good working knowledge of all the topics listed. Most of the questions will be very general, however you will be asked a few advanced questions.

Last but not least we're always interested to hear your feedback. You can send an email to your instructor and/or to training@springsource.com to let us know what you thought.

Topics By Subject Area

The following is a list of topics, each of which is likely to have questions on the exam. The topics are organized by subject area. They do not exactly reflect the module names in the course.

Container -basics

General

- The potential advantages of using Spring's dependency injection
- Dependency injection in XML, using constructor or setter injection
- Default scope for Spring beans. What are the main scopes that can be used alternatively?

Lifecycle

- How to declare an initialization method in a Spring bean
- How to declare a destroy method in a Spring bean
- Default bean instantiation policy: when are beans instantiated?

Note: *you should have a good understanding of the lifecycle in general. What are BeanFactoryPostProcessors and BeanPostProcessors? When are they called in the startup process?*

Annotations

- Enabling component-scanning
- Behavior of the annotation @Autowired with regards to field injection, constructor injection and method injection
- How does the @Qualifier annotation complement the use of @Autowired
- What is the role of the @PostConstruct and @PreDestroy annotations
- Enabling the scanning of annotations such as @Required and @PreDestroy

Miscellaneous

- How to inject scalar/literal values into Spring beans
- How to refer to a collection in a Spring bean definition
- How to create an ApplicationContext. What are the resource prefixes that can be used. How to refer to a Spring configuration file inside a package. The different implementations of ApplicationContext that can be used in an application.
- How to externalize constants from a Spring XML configuration file into a .properties file

- Purpose and usage of bean definition inheritance
- How to use the p namespace
- Difference between “id” and “name” attributes in the <bean> tag
- Purpose of the <aop:scoped-proxy/> tag

JavaConfig

- Usage of the @Bean and @Configuration annotations
- How to write a bean definition method

Testing

- How to use Spring's integration testing support in a JUnit 4 test
- How to a declare that a test should run in a transaction in spring-enabled JUnit 4
- Differences between a Unit test and an Integration test. Which of them is supposed to interact with Spring?

AOP

Recommendations

In order to work successfully with AOP, it is important to understand the theory behind the scenes. Consequently you should understand the four main AOP keywords taught in this course: Aspect, Advice, Pointcut and Joinpoint.

Note: *you will not be asked any question about the “advanced topics” section of the slides.*

Pointcuts

You should have a deep understanding of poincut expressions. It is important to understand all poincut examples in the slides (apart from the ones in the “advanced topics” section at the end of the module).

Advices

- Different kinds of Advice. How do they differ from each other?

Configuration

- Enabling the detection of @Aspect annotated classes. Is it possible to set up Spring AOP using XML configuration only (as opposed to annotations)?

Proxies

- When are they generated in the Spring lifecycle? How do they resemble the target object that they advice? What limitations does Spring-AOP's proxy-based approach have?

Data Access and transactions

General

- The rationale for having the DataAccessException hierarchy
- Definition of a DataSource in the Spring configuration

The JdbcTemplate

- Usage of the JdbcTemplate with regards to exception handling, querying, resultset parsing...

Hibernate

- Configuration of a SessionFactoryBean in xml

Note: *you will not be asked any question about the Hibernate syntax itself.*

Transactions

- Configuration to declare a local transaction manager
- Configuration to declare a JTA transaction manager
- The different ways to do declarative transaction management with Spring (xml, annotations)
- Usage of TransactionTemplate for programmatic transaction management (*you simply need to understand a basic example based on the TransactionTemplate*).
- Transactions and exception management: what is the default rollback policy? Can it be overridden?
- The @Transactional annotation: what are the main attributes that can be used for a transaction definition? Can this annotation also be used with a JTA Transaction Manager?
- Regarding propagation modes, what is the difference between PROPAGATION_REQUIRED and PROPAGATION_REQUIRES_NEW
- Regarding isolation levels, what is the difference between READ_COMMITTED and READ_UNCOMMITTED?

Note: *you will not be asked any question about the "advanced topics" section at the end of the "Transaction management with Spring" slides.*

Spring MVC and REST

General configuration

This module shows how to configure a ViewResolver,a HandlerMapping and the DispatcherServlet. You won't be asked any question on how to configure those classes. However, you need to know what the goal of each of those components is.

You also need to understand the relationship between a DispatcherServlet ApplicationContext and a root ApplicationContext.

Controllers

- Bootstrapping a root WebApplicationContext using the ContextLoaderListener
- General usage of the @Controller and @RequestMapping annotations
- A method annotated with @RequestMapping can return a String. What does it refer to? What are the main parameter types that this method can accept? (based on what you've seen in the class)
- Goal of the @RequestParam annotation

REST

- Differences between GET, POST, PUT and DELETE
- Usage of the @PathVariable annotation
- What is the RestTemplate? How should it be used?
- Purpose of the @ResponseStatus and @ExceptionHandler annotations

Advanced topics

Remoting

- Advantages of using Spring Remoting rather than plain RMI?
- Goal and general configuration of the RMI Service Exporter
- Goal and general configuration of RMI Proxy Generator
- Difference between the RMI Service Exporter and the HttpInvoker
- Does the HttpInvoker require to run a web server on the client side? On the server side?

Security

- What is the “Security filter chain” used in Spring Security?

- Syntax to configure Spring Security to intercept particular URLs? *(you need to have a good understanding of the various examples using intercept-url in the course)*
- Syntax to configure method level security
- Method level security using @Secured or @RolesAllowed
- Possible mechanisms to store user details: database? LDAP? Others?
- When working with an authentication provider, is it possible to use a password encoder?
- In the security tag library, what is the purpose of the <security:authorize /> tag?

JMS

- Purpose of the JmsTemplate
- How to declare a JMS Listener? Can a JMS Listener be a POJO?
- Is it possible to run JMS inside a plain Servlet container without full Java EE support (such as Tomcat or Jetty)?

JMX

- Role of the MBeanExporter
- Using Spring JMX, is it possible to export a POJO as an MBean?
- Using Spring JMX, is it possible to automatically register an existing MBean with an MBeanServer?
- Purpose of the @ManagedResource, @ManagedAttribute and @ManagedOperation annotations

Resources

This section contains a list of resources relating for learning.

Spring Community Forums – look for existing discussions or start your own, take advantage of one of the best parts of Spring: its community.

SpringSource Blog – point your favorite RSS reader or come back every so often for detailed, quality posts by Spring developers.

Reference Documentation – add bookmarks in your browser to the reference documentation pages for Spring.

Spring Samples – a subversion repository with projects that can be built with maven and imported into STS/Eclipse. Some of the samples have associated blog posts on the SpringSource Blog listed above. You could check by searching on the keywords: “sample name” springsource blog.

Spring By Example – another good repository with complete code samples and the ability to contribute your own samples.

Web Sites – it's hard to single out individual web sites. There are so many. If we had to name a few they would include Infoq, Dzone, JavaWorld, Spring Hub among many others.

Books – there are many books. It's also hard to find ones that are up-to-date because it takes so much effort to write them or keep them up-to-date, so check the date of the last edition and the version of the framework covered. Publishers like (e.g. Apress, Manning) provide early access to book chapters in PDF as they are being written.

Books about Spring have chapters on Spring MVC at least. To name a few quality Spring books: "Pro Spring 2.5", "Spring In Action", "Spring In Practice".

Spring Projects JIRA – most likely not the first place to come to in the beginning but overall a great learning resource when looking up information on very specific issues or new features. You can read comments, leave comments, as well as vote. Sometimes discussions on the community forums result in the creation of issues in JIRA.