

# Spring Web Developer Certification Study Guide Version 4.0.a

COMPLETED: 30 MARCH 2015

# Table of Contents

LOGISTICS .....	4
THE EXAM .....	4
THE EXAM-FAQ .....	4
SPRING WEB OVERVIEW .....	6
SPRING MVC PROGRAMMING MODEL ESSENTIALS .....	8
SPRING MVC VIEWS .....	9
SPRING MVC FORM PROCESSING .....	10
SPRING MVC REST .....	11
EXCEPTIONS.....	12
SPRING WEB FLOW OVERVIEW.....	12
SECURING WEB APPLICATIONS WITH SPRING SECURITY .....	14
TESTING WEB APPLICATIONS.....	15
TOOLS .....	16
CONCLUSION.....	17

# Overview

This guide is designed to help you prepare for the Spring Web Developer 4.0.a certification exam. Please beware it should not be used if you have attended a Spring Web course that was using Spring 3.x (that certification exam is called Spring Web 3 and there is a dedicated certification guide that goes with it dated April 2011).

The certification exam is based on the 4-day Spring Web 4.0.a 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 each training module:

1. Review the slides, making notes of questions
2. Work through the lab
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
6. Reading (at least partially) the reference documentation
7. Memorize the "big pictures", tables, overviews, etc

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.

## LOGISTICS

---

Certification can be taken at any Pearson Vue test center: <http://pearsonvue.com/vtcllocator>. To locate a test center you have to select "Information Technology (IT)" as the category and "Pivotal" as the testing program when you enter the site reachable with the link provided above. When you have found a test center near by you have to make an appointment in advance and provide the voucher you received via email. If you did not receive a voucher (may take up to 2 weeks) please email [education@pivotal.io](mailto:education@pivotal.io).

Arrive at the test center early to reduce your stress. You should at least have one identity card with you (please read the instructions carefully which you have gotten from the test center). You put all your belongings into a locker - you are not allowed to have books, pencils, paper, mobile phone or any other electronic devices with you. The room in which you are doing the exam is usually under camera surveillance.

## THE EXAM

---

The exam itself is a computer based exam. The software which is used to do the exam first gives you some general instructions: how to navigate, how to mark a question, etc - please read it carefully. Once you have agreed that you want to start you have 90 minutes to answer 50 multiple-choice questions. You must answer 38 questions correctly (76%) in order to pass the exam.

## THE EXAM-FAQ

---

1. *Is there anything in the exam, which was not covered in the course?*

No.

2. *Do I have to remember class names and method signatures?*

No. We think that this is why you are using an IDE - for us it's much more important that you've understood the concepts rather than learning method signatures. However you should recognize key Spring Web classes in code examples.

3. *Do I have to write, complete or rearrange source code?*

No. The only thing you should be able to do is read a snippet of code and understand what it's doing. This might be an example of a class implementing a listener and you will then see a couple of related questions. We do not ask you questions on things an IDE can do for you, like checking if the code will compile.

4. *Do I have to know any other APIs like JavaScript or JSP in Detail?*

No. Of course you should be able to read, understand and use JavaScript or JSP wherever it is necessary but this is not an exam about either.

5. *How long is the voucher valid?*

The voucher is valid one year (please also check the expiry date in the email) and it allows you to do one attempt. If you don't pass, then you must purchase another voucher (not the whole course again) for yourself. Also please note - if you fail an attempt - there is a mandatory 6 weeks lockout period until you can have your next try.

6. *Are the advanced slides also part of the exam?*

No. Only the content presented before each chapter lab slide will be on the exam. Any course content presented after any chapter lab will not be on the exam. No content from the optional chapters will be on the exam.

# Topics

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.

## SPRING WEB OVERVIEW

---

- Basic facts about what Spring Web is, what products it consists of and how they relate to each other in terms of dependencies.
- How Spring applications can be loaded in a Servlet container independent of what framework is used (Spring MVC, Struts, etc.) to develop the web layer.
- How does the application context get initialized and from what files?

### TIP

*An easy way to explore project dependencies is to start the Spring Tool Suite, open a maven pom.xml file for a project that uses Spring MVC or Spring Web Flow, then click on the “Dependency Graph” (or the “Dependency Hierarchy”) tab.*

*Alternatively, browse <https://github.com/spring-projects/spring-mvc-showcase> (sample applications that highlight Spring MVC features).*

## SPRING MVC SERVLET CONFIGURATION

- How to configure the `ContextLoaderListener` and `DispatcherServlet` in web.xml or using pure Java (servlet-3 style)
- Understand `@EnableMvc` and the `WebMvcConfigurerAdapter` class
- Where Spring-based, `DispatcherServlet` configuration is expected to be by convention if a location is not provided.
- What configuration can be provided in the Spring-based, `DispatcherServlet` configuration and what configuration is used by default.
- What features were introduced in Spring 4 to manage static resources?
- What is the Resource Handling Chain?

### TIP

*An important concept to keep in mind is that the `DispatcherServlet` looks for certain types of Spring MVC “infrastructure” beans in its Spring configuration (see **mvc-configuration-options**). Examples of such types of beans include `MessageSource`, `HandlerMapping`, `ViewResolver`, and others. The exam will test your understanding of the purpose of these Spring MVC beans including what their purpose is, how they are discovered (by bean class type or by bean id), what implementations are configured by default, and so on.*

*Understanding Spring MVC “infrastructure” beans and what they do/provide is an important step to learning Spring MVC. One way to learn is to open each type listed in the sections*

*below, review the class-level Javadoc, and explore available subtypes (in Eclipse/STS use the Ctrl+T shortcut to open a class).*

*Another Eclipse/STS exercise is to use Ctrl+Space in Spring configuration files on bean property names in order to see the list of available properties. These properties describe how a given infrastructure type can be customized.*

## HANDLERMAPPING

- The purpose of the HandlerMapping strategy and the details of configuring it.
- How the DefaultAnnotationHandlerMapping and the ControllerClassNameHandlerMapping implementations work and what they would do in the case of specific incoming requests?

## HANDLERADAPTER

- The purpose of the HandlerAdapter strategy and the details of configuring it.

## HANDLERINTERCEPTOR

- The purpose of the HandlerInterceptor strategy and the details of configuring it
- What points in the request lifecycle can be intercepted, what data is available at each point?

## MESSAGESOURCE

- The purpose of the MessageSource strategy and the details of configuring it.

## THE MVC NAMESPACE

- What does the mvc namespace provide, what you can configure?
- What do *mvc:annotation-driven*, *mvc:interceptor*, *mvc:resource-chain* and *mvc:view-controller* actually do?

## CONVENTION OVER CONFIGURATION

- How is Spring MVC configured by default?
- What conventions are available to reduce explicit coding?
- What features, introduced by Spring 3, must be enabled by specifying `@EnableMvc` or *mvc:annotation-driven*?

## SPRING MVC PROGRAMMING MODEL ESSENTIALS

---

The basics of the annotation-based programming model in Spring MVC, configuring `@Controller`-annotated classes, writing request-handling methods, and testing them.

### @REQUESTMAPPING ANNOTATION

- The purpose of the annotation, what can be annotated, and what options (or attributes) it provides.
- How a URL breaks down (web application context, servlet name, path info) as well as which part of the URL is used in Spring MVC for request mapping purposes.

#### TIP

*In the exam you may be given basic examples of URL's and annotated controller methods. You will be expected to predict what methods will be invoked. The best way to experiment is to try it out. Use the course labs, and check the Spring MVC logging information on each request!*

*Also examine the logs for output from HandlerMapping beans. HandlerMapping beans construct URL mappings during startup and log that information. The log level for `org.springframework.web` must be set to `DEBUG`.*

*Below are a couple of specific scenarios to experiment with.*

*Scenario 1: a `DefaultAnnotationHandlerMapping` with one Controller and a single method annotated with `@RequestMapping("/foo")`.*

*Questions: What URL's reach the method successfully? Does adding an extension to the URL (.pdf, .xml) make a difference? How about more segments (/foo/bar)? Passing anything (/other)? Try switching to `@RequestMapping(method=RequestMethod.GET)`, does it still work?*

*Scenario 2: Similar to scenario 1 but involving a `ControllerClassNameHandlerMapping`, a controller (`FooController`) and a method (`bar`) annotated with `@RequestMapping(method=RequestMethod.GET)`*

### REQUEST HANDLING METHODS

- How to write methods to handle requests
- What annotations can be used?
- What the signature of the method can be – input argument types, return types?
- What happens if the method returns void?

#### TIP

*In addition to the annotations listed below, the JavaDoc of the `@RequestMapping` annotation is a good place to start for information on how input arguments and return values are interpreted on `@RequestMapping`-annotated methods.*



## @REQUESTPARAM AND @PATHVARIABLE ANNOTATIONS

- The purpose of the annotation
- What can be annotated?
- What options (or attributes) does it provide?
- Can it handle optional parameters?

## @MODELATTRIBUTE ANNOTATION

- The purpose of the annotation, what can be annotated, what options (or attributes) it provides?
- What is the default name given to a model attribute object if the attribute name is left unspecified?
- What is the default attribute name given to a model attribute object that is an array or a collection?

## SPRING MVC VIEWS

---

- The basics of how views work.
- What do they do?
- How are they typically instantiated through the process of view resolution?
- What is a logical view name?
- What is the default logical view name selected if a controller method does not specify it (method returns void or null).

## VIEW RESOLVERS

- The purpose of the ViewResolver strategy, and the details of configuring it?
- You should be familiar with the several different view-resolvers covered on the course.
- How do ViewResolver chains work?
- How they can be used to render multiple content types – for example to re-use the same controller method to render HTML, PDF, or XML depending on the content type requested by the client.

## SPRING MVC FORM PROCESSING

---

- The basics of working with forms such as how to configure data binding through the `@ModelAttribute` annotation and how data binding is used to populate the form object from request parameter values.
- Can data binding be used on a POST or can it also be done on a GET request (consider search forms vs. forms updating data).
- How can a request handling method get access to the results of data binding?

### @SESSIONATTRIBUTES ANNOTATION

- The purpose of the `@SessionAttributes` annotation, what can be annotated, and what options it provides.

#### TIPS:

*The `@SessionAttributes` annotation provides more than one way to specify what objects should be added to the HTTP Session. Be sure to check them. A good way to learn what options any Spring annotation provides is the Javadoc of the annotation.*

- The lifecycle of attributes specified with the `@SessionAttributes` annotation – how long they remain around, when and how they can be removed from the HTTP Session.
- How the `SessionAttributes` works when it's used with multiple controllers – for example is the data stored globally to the HTTP session (e.g. user preferences) or is it per-controller (e.g. account editing).

### @INITBINDER ANNOTATION AND DATA BINDING CUSTOMIZATIONS

- The purpose of the `InitBinder` annotation, what can be annotated, and what options it provides.
- What customizations can be applied to the data binding mechanism?
- What error codes are generated automatically during data binding?
- What error codes can be used to customize the errors generated during data binding?

### SPRING FORM TAG LIBRARY

- The purpose and the value provided by the Spring form tags, and how to use them.
- How are error-messages handled? In the Controller? Using form tags?

### FORMATTERS AND VALIDATION

- Understand the Spring stateless formatters (introduced by Spring 3.0)
- What can they do, where are they used?
- How do they interact with the formatting annotations?

- How can the form model object be validated?
- How do Spring forms leverage JSR 303 bean validation?
- How can error messages be customized?

## SPRING MVC REST

---

- What does REST stand for?
- What is a resource?
- What are safe operations?
- What are idempotent operations? Why is idempotency important?
- Is REST scalable and/or interoperable?
- What are the advantages of the RestTemplate?
- Which HTTP-Methods does REST use?
- What is an HttpMessageConverter?
- Is REST stateless?
- What does @RequestMapping do?
- Is @Controller a stereotype? Is @RestController a stereotype?
- What is the difference between @Controller and @RestController?
- When do you need @ResponseBody?
- What does @PathVariable do?
- What is the HTTP status code for a delete statement?
- What does CRUD mean?
- Is REST secure? What can you do to secure it?
- Where do you need @EnableWebMvc?
- Name some common http response codes. When do you need @ResponseStatus?
- Does REST work with transport layer security (TLS)?
- Do you need SpringMVC in your classpath?

## EXCEPTIONS

---

- How can exceptions be handled in the MVC framework?
- What does a ControllerAdvice class do?
- How do you set the HTTP status of a response in the event of an exception (to avoid the default 500 error)
- Would you understand the configuration of a HandlerExceptionResolver if you saw it?
- What can a SimpleMappingExceptionHandler be used for?
- How can a RESTful request return an error?

## SPRING WEB FLOW OVERVIEW

---

**NOTE:** All certification questions relate to the *webflow-overview* section, **not** the optional modules at the back (which cover the same material, and more, but in extra detail).

### FLOW BASICS

- What is the motivation for the existence of Spring Web Flow?
- What are the common problems it solves?
- A basic understanding of how Spring Web Flow fits into Spring MVC.

### FLOW CONFIGURATION

- How Spring Web Flow configured?
- Consider both Spring MVC and Spring Web Flow-specific configuration.

#### **TIPS:**

*Like Spring MVC, Spring Web Flow provides and expects some infrastructure classes to be configured as Spring beans. Some of these infrastructure beans (FlowHandlerMapping, FlowHandlerAdapter) are Spring MVC-specific, while others (FlowExecutor, FlowRegistry, and others) are specific to Spring Web Flow. Listed below are the infrastructure types you should review and understand.*

### FLOW HANDLER MAPPING

- The purpose of the FlowHandlerMapping strategy
- The details of configuring it

- How it decides whether an incoming URL matches to any registered flows.

## FLOW HANDLER ADAPTER

- The purpose of the FlowHandlerAdapter strategy?

## FLOW EXECUTOR

- The purpose of the FlowExecutor?
- What configuration options does it provide?

## FLOW REGISTRY

- The purpose of the FlowRegistry
- How are flows registered and how are id's assigned to them?

### TIPS:

*In the exam you may be given example FlowRegistry configuration and incoming URL's. You're then expected to know which URL would match to a given flow.*

*Specifically consider the possibility for registering flows one by one and many at once using a pattern. In each case how does a flow get assigned an id?*

## FLOW BUILDER SERVICES

- The purpose of the FlowBuilderServices?
- What configuration options does it provide?
- What is the development mode option?

## FLOW DEFINITION

- What a Web Flow view state is and what it represents. What happens when a view state is reached. How a view state is resolved to a specific view such as a JSP page. What the attributes of a view state are.
- What a transitions in Web Flow is. What triggers a transition, what happens when a transition is triggered, and whether a transition can be prevented once it's been triggered. What global transitions are.
- How a button or a link on an HTML page can be used to raise a specific Web Flow event.
- What a Web Flow end state is. What happens when an end state is reached. What Web Flow does by a default for a flow that has ended. What is a good practice for what to do when a flow ends.
- How to write a Web Flow unit test, what base class to extend, what methods you'd expect to override, and how to implement test methods that navigate through the flow.

## FLOW ACTIONS

- What scopes Web Flow provides and how long variables in each scope live.
- The 3 ways to create a variable in a flow definition, what the scope of the created variable is and whether it will be dependency injected in each case.
- How Web Flow resolves variables encountered in the EL expressions within a flow definition. For example will it search the various web flow scopes or in Spring configuration, and are there reserved words? Also important is the order in which Web Flow goes to try to resolve variables encountered in EL.

### TIPS:

*To make this easier, open an existing flow definition, and try to locate all the EL expressions in evaluate elements. For each expression determine what variables are used what their origin is – a reserved keyword, a scoped variable, a Spring bean? Also notice how when referencing scoped variables you don't have to specify what scope they come from – flash, view, flow, etc. Web Flow will automatically look in all scopes and try to locate the matching variable.*

- What the 4 different types of Web Flow actions are. Where actions can be embedded in a flow definition. What actions are invoked when. For example what is the difference between on-render actions and on-entry actions? How to transitions trigger actions?
- Understand basic form handling in Web Flow.

## SECURING WEB APPLICATIONS WITH SPRING SECURITY

---

- What web.xml configuration is required to enable Spring Security and what the mechanism Spring Security uses to protect web applications.
- What Spring Security related configuration is needed to secure a web application.
- How should URL patterns be configured?

### TIPS:

*Pay special attention to the order in which URL patterns are provided. Does it matter if you put the more general (e.g. /accounts/\*) or the more specific (e.g. /accounts/edit) pattern first?*

- How method level security can be added to an application.
- How authentication and authorization relate to each other – for example does the choice of authentication affect authorization?
- What security options are available within Spring Web Flow. What elements can be made secure.

## TESTING WEB APPLICATIONS

---

- How can the MVC layer be tested?
- How do we test Controller *logic*?
- How do we test if a Controller works properly inside the MVC framework?
- What framework artifacts/components does Spring's Mock MVC allow you to test? (Refer to *Testing Each Layer* slide)
- If presented with the code for a Mock MVC test would you understand what it is doing? – we *don't* expect you to know the API from memory.

## Resources

This section contains a list of resources relating for learning.

- The best place to go for help is at [StackOverflow](#) – look for existing discussions or start your own, take advantage of one of the best parts of Spring: its community.
- A list of Spring topics from StackOverflow can also be found at [Spring Questions](#).
- The old Spring forums are no longer used, but they are available as an archive at [Spring Community Forums](#).
- [Spring 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](#) (namely the chapter on [Spring MVC](#)), [Spring Web Flow](#), and [Spring Security](#).
- The Spring Framework API [Javadoc](#).
- The Spring IO website has many [Spring Guides](#).
- [Spring By Example](#) – a great repository with complete code samples and the ability to contribute your own samples.
- [Spring Samples](#) – an older `svn` repository with projects that can be built with maven and imported into STS/Eclipse. Some of the samples have associated blog posts on the Spring Blog listed above. You could check by searching the blog for a specific sample project by name. *Warning:* these samples are quite old, but you will usually find a newer version of the same/similar code on [Github](#) – just do a search.
- It's hard to single out individual web sites, there are so many, but we had to name a few they would include [Infoq](#), [Dzone](#), [JavaWorld](#) among many others.

- There are also many books and even the best become out of date quite quickly. Publishers like (e.g. Apress, Manning) provide early access to book chapters in PDF as they are being written. All the well-known books have chapters on Spring MVC: Pro Spring, Spring in Action and Spring in Practice, but there are many others.
- [Spring books at Apress](#)
- [Spring in Action](#) and [Spring in Practice](#) at Manning. Other [Spring Books at Manning](#).
- Books about Spring Web Flow are harder to come by and your mileage with them will vary. There are several at [Apress](#).
- [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, new features or potential bugs. You can read comments, leave comments, as well as vote. Sometimes discussions on the community forums result in the creation of issues in JIRA.

## TOOLS

---

### [Spring Tool Suite](#)

Spring project templates, tutorials, bean diagrams, code completion and more. See the PDF files under “Related Content” for details on available features. STS is updated often and it's free. It also supports Spring Roo and Grails.

### [Spring Roo](#)

A rapid web application development tool based on most of the technologies taught in the course and also a great tool for “rapid” learning. However it currently (October 2014) generates Spring 3 code and is not yet updated for Spring 4.

### [Grails](#)

A rapid web-application development platform for Groovy that also uses Spring Web technologies. Grails has a diverse set of [plugins](#) that can introduce you to many web technologies, techniques, and approaches.

### [Spring Insight](#)

When you download STS from the link above, it includes a version of *tc Server* (Pivotal's supported Tomcat Server) bundled with the Spring Insight webapp. It gives you lightweight tracing of your Spring application including links to the source code. A great tool for learning that can be used with existing applications with no changes to the application.

### [Cloud Foundry](#)

Pivotal's open *Platform as a Service* (PaaS) is a great place to run Spring and Java applications (among others). [Pivotal Web Services](#) is our public cloud offering and a trial account is free for 60 days and pretty cheap thereafter. If you want to run Spring applications



*and* make them available to friends or colleagues world-wide, PWS is easy to use and requires no hosting by you.

## CONCLUSION

---

Once you have worked through this guide and know all the answers, we are pretty sure you will pass the certification. It's recommended to do it as soon as possible and we wish you good luck for it.

Thank you very much again for choosing Pivotal as your education partner and also good luck with your projects.

If you have encountered any errors or if you have any other suggestions please don't hesitate contacting your trainer or write an email to [education@pivotal.io](mailto:education@pivotal.io).

**Pivotal, March 2015**