



Spring Framework Reference Documentation

4.3.5.RELEASE

Rod Johnson , Juergen Hoeller , Keith Donald , Colin Sampaleanu , Rob Harrop , Thomas Risberg , Alef Arendsen , Darren Davison , Dmitriy Kopylenko , Mark Pollack , Thierry Templier , Erwin Vervaeet , Portia Tung , Ben Hale , Adrian Colyer , John Lewis , Costin Leau , Mark Fisher , Sam Brannen , Ramnivas Laddad , Arjen Poutsma , Chris Beams , Tareq Abedrabbo , Andy Clement , Dave Syer , Oliver Gierke , Rossen Stoyanchev , Phillip Webb , Rob Winch , Brian Clozel , Stephane Nicoll , Sebastien Deleuze

Copyright © 2004-2016

Copies of this document may be made for your own use and for distribution to others, provided that you do not charge any fee for such copies and further provided that each copy contains this Copyright Notice, whether distributed in print or electronically.

Table of Contents

I. Overview of Spring Framework	1
1. Getting Started with Spring	2
2. Introduction to the Spring Framework	3
2.1. Dependency Injection and Inversion of Control	3
2.2. Modules	3
Core Container	4
AOP and Instrumentation	5
Messaging	5
Data Access/Integration	5
Web	5
Test	6
2.3. Usage scenarios	6
Dependency Management and Naming Conventions	9
Spring Dependencies and Depending on Spring	11
Maven Dependency Management	11
Maven "Bill Of Materials" Dependency	12
Gradle Dependency Management	12
Ivy Dependency Management	13
Distribution Zip Files	13
Logging	13
Not Using Commons Logging	14
Using SLF4J	14
Using Log4J	15
II. What's New in Spring Framework 4.x	17
3. New Features and Enhancements in Spring Framework 4.0	18
3.1. Improved Getting Started Experience	18
3.2. Removed Deprecated Packages and Methods	18
3.3. Java 8 (as well as 6 and 7)	18
3.4. Java EE 6 and 7	19
3.5. Groovy Bean Definition DSL	19
3.6. Core Container Improvements	19
3.7. General Web Improvements	20
3.8. WebSocket, SockJS, and STOMP Messaging	20
3.9. Testing Improvements	21
4. New Features and Enhancements in Spring Framework 4.1	22
4.1. JMS Improvements	22
4.2. Caching Improvements	22
4.3. Web Improvements	23
4.4. WebSocket Messaging Improvements	24
4.5. Testing Improvements	24
5. New Features and Enhancements in Spring Framework 4.2	26
5.1. Core Container Improvements	26
5.2. Data Access Improvements	27
5.3. JMS Improvements	28
5.4. Web Improvements	28
5.5. WebSocket Messaging Improvements	29
5.6. Testing Improvements	29

6. New Features and Enhancements in Spring Framework 4.3	32
6.1. Core Container Improvements	32
6.2. Data Access Improvements	32
6.3. Caching Improvements	32
6.4. JMS Improvements	33
6.5. Web Improvements	33
6.6. WebSocket Messaging Improvements	33
6.7. Testing Improvements	33
6.8. Support for new library and server generations	34
III. Core Technologies	36
7. The IoC container	37
7.1. Introduction to the Spring IoC container and beans	37
7.2. Container overview	37
Configuration metadata	38
Instantiating a container	39
Composing XML-based configuration metadata	40
Using the container	41
7.3. Bean overview	41
Naming beans	42
Aliasing a bean outside the bean definition	43
Instantiating beans	44
Instantiation with a constructor	45
Instantiation with a static factory method	45
Instantiation using an instance factory method	45
7.4. Dependencies	47
Dependency Injection	47
Constructor-based dependency injection	47
Setter-based dependency injection	49
Dependency resolution process	50
Examples of dependency injection	51
Dependencies and configuration in detail	53
Straight values (primitives, Strings, and so on)	53
References to other beans (collaborators)	55
Inner beans	56
Collections	56
Null and empty string values	58
XML shortcut with the p-namespace	59
XML shortcut with the c-namespace	60
Compound property names	61
Using depends-on	61
Lazy-initialized beans	61
Autowiring collaborators	62
Limitations and disadvantages of autowiring	63
Excluding a bean from autowiring	64
Method injection	64
Lookup method injection	65
Arbitrary method replacement	67
7.5. Bean scopes	68
The singleton scope	69
The prototype scope	70

Singleton beans with prototype-bean dependencies	71
Request, session, global session, application, and WebSocket scopes	71
Initial web configuration	72
Request scope	72
Session scope	73
Global session scope	73
Application scope	73
Scoped beans as dependencies	74
Custom scopes	76
Creating a custom scope	76
Using a custom scope	77
7.6. Customizing the nature of a bean	78
Lifecycle callbacks	78
Initialization callbacks	79
Destruction callbacks	79
Default initialization and destroy methods	80
Combining lifecycle mechanisms	81
Startup and shutdown callbacks	82
Shutting down the Spring IoC container gracefully in non-web applications	84
ApplicationContextAware and BeanNameAware	84
Other Aware interfaces	85
7.7. Bean definition inheritance	86
7.8. Container Extension Points	88
Customizing beans using a BeanPostProcessor	88
Example: Hello World, BeanPostProcessor-style	89
Example: The RequiredAnnotationBeanPostProcessor	91
Customizing configuration metadata with a BeanFactoryPostProcessor	91
Example: the Class name substitution PropertyPlaceholderConfigurer	92
Example: the PropertyOverrideConfigurer	93
Customizing instantiation logic with a FactoryBean	94
7.9. Annotation-based container configuration	94
@Required	96
@Autowired	96
Fine-tuning annotation-based autowiring with @Primary	99
Fine-tuning annotation-based autowiring with qualifiers	100
Using generics as autowiring qualifiers	105
CustomAutowireConfigurer	106
@Resource	106
@PostConstruct and @PreDestroy	108
7.10. Classpath scanning and managed components	108
@Component and further stereotype annotations	108
Meta-annotations	109
Automatically detecting classes and registering bean definitions	110
Using filters to customize scanning	111
Defining bean metadata within components	112
Naming autodetected components	115
Providing a scope for autodetected components	115
Providing qualifier metadata with annotations	116
7.11. Using JSR 330 Standard Annotations	117

[Click here to download full PDF material](#)