Spring Core

By Sunil Manaka

https://www.facebook.com/JavaTez

YouTube **JavaTez**

GITHub URL:

https://github.com/SunJavaKumar/SpringCore

First things first

1) Close all tabs on your system
2) Put your mobile on silent mode
3) Take a note book & pen
4) Please stay till end of the session

What is Spring?

Spring is lightweight framework to build java applications.

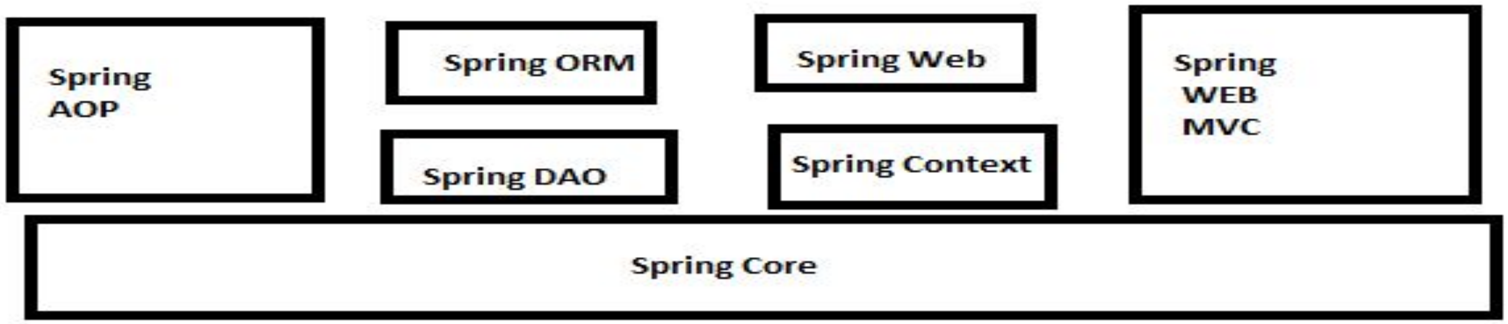
We can build any kind of java application using Spring (example: stand-alone, web or JEE application).

Pre-Requirements:

Good knowledge on JAVA is required.

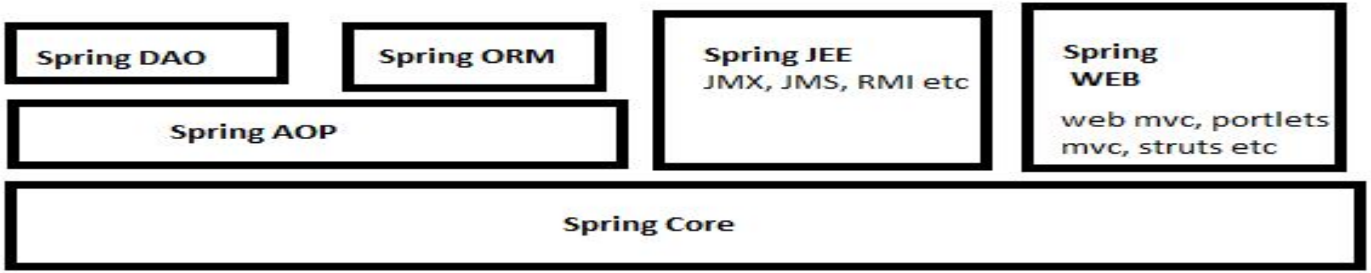
Spring 1.x Modules

Developed by Rod Johnson (Wrox, 2002)



- 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 Modules - released on October 2006



- ➢ 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.

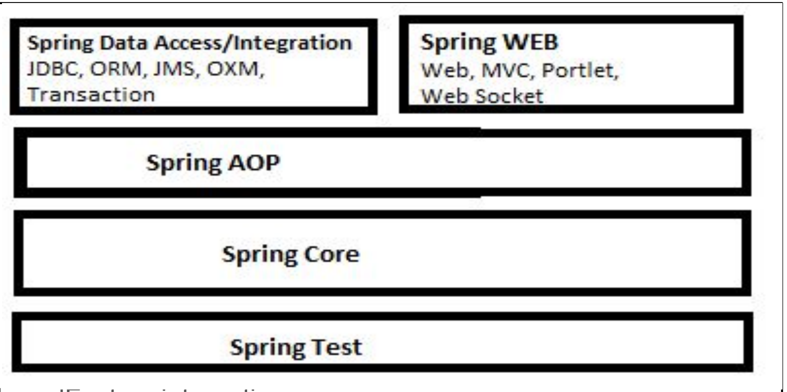
Spring 2.5 Features Released on November 2007

- 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.x Modules

Spring 3.0 in December 2009

Spring 3.1 in December 2011
Spring 3.2.5 in November 2013



Spring Data Access/Integration
JDBC, ORM, JMS, OXM, Transaction

Spring WEB
Web, MVC, Portlet, Web Socket

Spring AOP

Spring Core

Spring Test

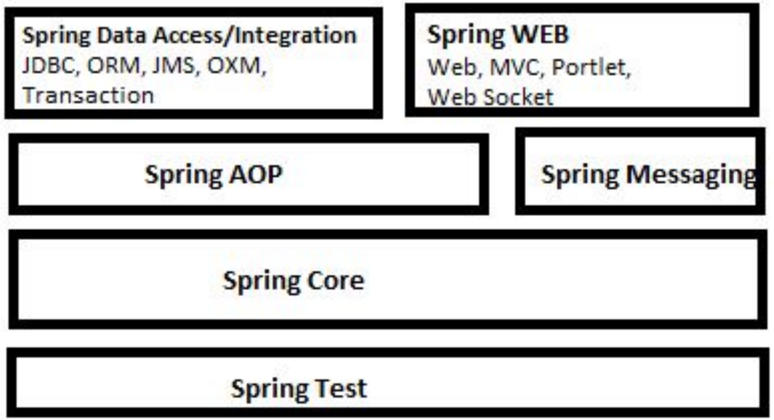
- Integration of core Java Config features and annotations
- Introduction of the Spring Expression Language (SpEL)
- General-purpose type conversion system and field-formatting system
- Validation enhancements and JSR-303 (Bean Validation) support
- Java 5 features such as generics, varargs, and other improvements
- New cache abstraction
- Support for Hibernate 4
- URI template variable enhancements
- New Spring MVC test framework.
- Jackson JSON 2 support and Support for Tiles 3.
- First--class support for Callables, Futures, ExecutorService adapters, and ThreadFactory integration
- Framework modules now managed separately with one source tree per module JAR
- Comprehensive REST support – New MVC XML namespace and additional annotations such as @CookieValue and @RequestHeaders for Spring MVC Figure 1-3. Overview of the Spring Framework, version 3.0.x
- 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
- Support for @DateTimeFormat without Joda Time and Global date and time formatting, New Gradle-based build system
- Migration to GitHub (https://github.com/SpringSource/spring-framework)
- @ControllerAdvice, @MatrixVariable, @RequestBody, @RequestBody, @RequestPart @ComponentScan, EnableTransactionManagement, @EnableCaching, @EnableWebMvc, @EnableScheduling, @EnableAsync, @EnableAspectJAutoProxy, @EnableLoadTimeWeaving,@Profile and @EnableSpringConfigured

Spring 4.x Modules

Spring Framework 4.0 was released in December 2013

Spring Framework 4.2.0 was released on 31 July 2015
Spring Framework 4.2.1, was released on 01 Sept 2015
Spring Framework 4.3 has been released on 10 June 2016

- Composed annotations were added
- JMS and web improvements
- Full support for Hibernate ORM 5.0
- The programming model has been refined
- Core container, testing, and general web improvements
- WebSocket, SockJS, and STOMP messaging
- Removal of deprecated packages and methods from the prior Spring 3 version
- Java 8 support, raising the minimum Java version to 6 update 18
- Java EE 6 and above is now considered the baseline for Spring Framework 4.0
- Groovy bean definition DSL, allowing bean definitions to be configured via
- Core improvements (eg., introduction of @AliasFor and modification of existing
- annotation to make use of it)
- Considerable improvements in the core container (inclusions of ASM 5.1, CGLIB
- 3.2.4, and Objenesis 2.4 in spring-core.jar) and MVC
- Spring TestContext Framework requires JUnit 4.12 or higher.
- Support for new libraries, including Hibernate ORM 5.2, Hibernate Validator 5.3,
- Tomcat 8.5 and 9.0, Jackson 2.8, and so on



Spring Data Access/Integration
JDBC, ORM, JMS, OXM, Transaction

Spring WEB
Web, MVC, Portlet, Web Socket

Spring AOP

Spring Messaging

Spring Core

Spring Test

Spring 5.0.x was released on September 28, 2017

- Support was dropped for Portlet, Velocity, JasperReports, XMLBeans, JDO, Guava, Tiles2, and Hibernate3.
- XML configuration namespaces are now streamed to unversioned schemas; versionspecific declarations are still supported but validated against the latest XSD schema.
- Overall improvements were introduced by harnessing the full power of Java 8 features.
- The Resource abstraction provides isFile indicator for defensive getFile access.
- Full Servlet 3.1 signature support in Spring-provided Filter implementations.
- Support for Protobuf 3.0.
- Support for JMS 2.0+, JPA 2.1+.
- Introduction of Spring Web Flow, a project that is an alternative to Spring MVC built on a reactive foundation, which means that it is fully asynchronous and non-blocking, intended for use in an event-loop execution model vs. traditional large thread pool with a thread-per-request execution model (built upon Project Reactor3).
- The web and core modules were adapted to the reactive programming model.4
- There are a lot of improvements in the Spring test module. JUnit 5 is now supported, and new annotations were introduced to support the Jupiter programming and extension model such as @SpringJUnitConfig, @SpringJUnitWebConfig, @EnabledIf, @DisabledIf.
- Support for parallel test execution in the Spring TestContext Framework

Spring Java Compatibility

Spring Version	Java Version
Spring 2.x	JDK 5
Spring 3.x	JDK 5+
Spring 4.x	JDK 6+
Spring 5.x	JDK 8+

- Java SE 10 = 54
- Java SE 9 = 53
- Java SE 8 = 52
- Java SE 7 = 51
- Java SE 6.0 = 50
- Java SE 5.0 = 49
- JDK 1.4 = 48
- JDK 1.3 = 47
- JDK 1.2 = 46
- JDK 1.1 = 45

NOTE: Unsupported major.minor version 5X.0

Spring History:

@author: Rod Johnson

Version	Dates
0.9	2002
1.0	2003
2.0	2006
3.0	2009
4.0	2013
5.0	2017

Spring Container

org.springframework.beans.factory.**BeanFactory**

org.springframework.context.**ApplicationContext**

What is Spring Bean ?

Configuring Metadata.

(XML/Annotations/JavaConfig)

How many ways we can create object in Spring?

Inversion of Control

Dependency Lookup

Dependency Injection

- Constructor Dependency Injection
- Setter Dependency Injection
- Field Dependency Injection

Circular Dependencies in Spring

Dependency Resolution Process

Dependency Configuration In-detail

- Primitive Types
- String Types
- Referencing Other beans
- Collection Types
- Null assignment
- Empty String assignment
- Compound Property resolution

Dependens-on

Bean Lazy Initialization

Autowiring in Spring

Autowiring through XML Config

- no
- byName
- byType
- constructor

@Autowired

@Primary

@Qualifier

@Required(On top of Setter Method)

Bean Scopes

- Singleton
- Prototype
- Request
- Session
- Application
- Custom Scope Example

@Scope(Annotation/JavaConfig)

Singleton vs Prototype

(Method Injection)

Bean Definition Inheritance

Spring Bean Initialization CallBacks

(init-method, InitializingBean, @PostConstruct)

Spring Bean Destruction CallBacks

(destroy-method, DisposableBean, @PreDestroy)

Combining Initialization Callbacks

Combining Destruction Callbacks

Method Replacement

Namespace ShortCut

Java Configuration in-detail

@Configuration

@Bean

(initMethod, destroyMethod, @Scope,name,@Descritpion)

@Import

Combining XML & Java Configuration

BeanPostProcessor

Bean Aware Interfaces

Spring Bean Life Cycle

Spring JSR-330 Support

@Named -> @Component

@Inject -> @Autowired

@Singleton -> @Scope(“singleton”)

BeanFactory

vs

ApplicationContext

Like, Comment & Subscribe.

Thank you.

Sunil Manaka