

## Implementing Spring 4 and its integration features in Project Scenarios

### Workshop Details:

<b>Duration :</b>	2 days
<b>Objective and Take Away :</b>	Implementation of integration, performance, best practices and ready-ness for handling complex spring projects. <b>Take Away:</b> Project level acquaintance with best practices, design patterns, integration, security etc.
<b>Participants' Entry Profile :</b>	For developers experienced in java who wish to use Spring Framework to build their applications. Fresher's who have good knowledge of java and working knowledge of database. Knowledge of JNDI, REST, WEB Services, Hibernate and struts would be added advantage
<b>Training Methodology :</b>	The workshop will follow Synergetics methodology of <b>Concept Visualization</b> <b>Active Experimentation</b> <b>Application Development</b> The workshop will be 60% Hands-On with each participant having access to system during the session

### Setup Requirements

<b>Hardware and Software Requirements :</b>	Participant's as well as Trainer's Machine are required to have : <b>Hardware</b> Intel Core i3 2.2 GHz 80 GB HDD LCD Color Monitor 2 GB RAM Internet Connectivity for Maven dependencies <b>Software</b> Windows XP or 7 Internet Explorer 10 or above/Chrome/Firefox Eclipse Luna with STS plugins ( <b>Must match with bit architecture of machine</b> ) Apache Maven [included in eclipse luna] Oracle 10g/11g Express Edition Oracle JDBC Driver [ojdbc6.jar] Spring Framework 4.0 (All Jars)  <ul style="list-style-type: none"> <li>• The installable must match in bits architecture of the machine.</li> <li>• A shared disk-space among participants and trainer for trainer to share training stuff with all participants.</li> <li>• Internet connection to fetch maven dependencies .</li> </ul>
<b>Training Lab Requirements:</b>	Whiteboard 6 feet by 4 feet (minimum) Whiteboard markers – Red, Blue, Green, Black Video Projector (1024 X 768 resolutions)

## Pre-requisite for training

A skill-set equivalent to Synergetics-Spring IOC/DI, JDBC, Transactions and AOP(L1\_04\_SprngIOC4)

- Spring Framework and its architecture
- Dependency Injection
- Working with multiple Configuration files
- Lazy or eager bean creation
- Spring beans and their life cycle
- Spring JDBC
- Integration with Hibernate
- Transaction support in Spring
- Logger support in Spring
- Jar Distribution in Spring 4

## Course Content

<b>Day 1</b>	<ul style="list-style-type: none"> <li>➤ <b>Module 1: Bean life cycle and container callback hooks</b> <ul style="list-style-type: none"> <li>○ Bean Life Cycle Callbacks- initMethod and interfaces</li> <li>○ Life cycle annotations: @PostConstruct and @PreDestroy</li> <li>○ Container callback hooks on bean lifecycle</li> <li>○ Container life cycle in brief</li> </ul> </li>   <li>➤ <b>Module 2: Programmatic bean creation using Factory, Parameterized Factory</b> <ul style="list-style-type: none"> <li>○ Spring creating Singleton bean programmatically</li> <li>○ Spring using Factory for bean creation</li> <li>○ Spring using Parameterized factory for bean creation</li> </ul> </li>   <li>➤ <b>Module 3: More on Spring IOC</b> <ul style="list-style-type: none"> <li>○ Features of ApplicationContext <ul style="list-style-type: none"> <li>▪ Multiple application context</li> <li>▪ Event handling model</li> <li>▪ Internationalization support-MessageSourceAware</li> </ul> </li> </ul> </li>   <li>➤ <b>Module 4: Cache support in Spring</b> <ul style="list-style-type: none"> <li>○ Understanding cache abstraction</li> <li>○ The Cacheable and CacheEvict Annotations</li> <li>○ The custom key generation and conditional caching</li> <li>○ Configuring for ECCache</li> </ul> </li>   <li>➤ <b>Module 5: Unit Testing support in Spring</b> <ul style="list-style-type: none"> <li>○ The 'Profiles'</li> <li>○ JUnit Test Cases for Spring Beans</li> <li>○ @ContextConfiguration and @RunWith annotations</li> <li>○ Using Mock objects for testing</li> <li>○ Unit testing support in Spring 4</li> </ul> </li>   <li>➤ <b>Module 6: Spring 4 enhancements</b> <ul style="list-style-type: none"> <li>○ The @Lazy for lazy or eager loading of bean</li> <li>○ The @Primary annotation for 'byType' autowiring</li> <li>○ The support of Generics</li> <li>○ Conditional Bean Creation</li> <li>○ The DateTime support</li> </ul> </li> </ul>
--------------	--

<b>Day 2</b>	<ul style="list-style-type: none"><li>➤ <b>Module 7: Spring Integration</b><ul style="list-style-type: none"><li>○ Spring integration framework</li><li>○ Remoting: Using Spring remoting and the Spring HttpInvoker for remote access</li><li>○ Spring-JNDI integration</li><li>○ Spring WS- Publishing and consuming services</li><li>○ The @RestController and publishing REST services</li><li>○ Consuming REST services in Spring</li><li>○ JMS: Sending and receiving messages using the JmsTemplate</li><li>○ Spring-Hibernate integration: SessionFactory, HibernateTemplate, HibernateDaoSupport</li><li>○ Spring-Struts Integration</li></ul></li> <li>➤ <b>Module 8: Introduction to some of the Spring Modules</b><ul style="list-style-type: none"><li>○ Spring MVC and Spring Web Flow</li><li>○ Spring Security and Authorization</li><li>○ Spring OXM</li><li>○ Spring Batch</li><li>○ Spring Data</li><li>○ Spring JMX</li></ul></li></ul>